Informed gambling : conception and analysis of a multi-agent mechanism for discrete reallocation

DOI:
10.26481/dis.19991203jl

INFORMED GAMBLING

Conception and analysis of
a multi-agent mechanism
for discrete reallocation

INFORMED GAMBLING

Conception and analysis of
a multi-agent mechanism
for discrete reallocation

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit Maastricht,
op gezag van de Rector Magnificus,
Prof. dr. A.C. Nieuwenhuijzen Kruseman,
volgens het besluit van het College van Decanen,
in het openbaar te verdedigen op
vrijdag 3 december 1999 om 12.00 uur

door

Jacob Hilko Jan Lenting

Promotor: Prof. dr. H.J. van den Herik
Co-promotor: Dr. P.J. Braspenning

Leden van de beoordelingscommissie:
Prof. dr. H.L.M. Peters (voorzitter)
Prof. dr. H. Akkermans (Vrije Universiteit Amsterdam)
Dr. C.P.M. van Hoesel
Prof. dr. B. Huberman (Stanford University)
Dr. ir. J. Talmon

SIKS dissertation series No. 99-8

NWO The research reported in this thesis was funded by the Netherlands Organization for Scientific Research (NWO).

The support by KPN Research is gratefully acknowledged.

ISBN 90-9013282-1
NUGI 855
Subject headings: multi-agent systems / allocation / discrete optimization

Cover photos: Organice Entertainment

If we begin with certainties, we shall end in doubts, but if we begin with doubts, and are patient in them, we shall end in certainties.

Francis Bacon

Aan mijn ouders

Acknowledgements

I thank my supervisor, Jaap van den Herik, for his support and guidance. His teachings in scientific writing and rewriting are greatly appreciated. Without his uncanny ability to combine incessant thrust with angelic patience, this thesis would not have appeared in the present millennium, if at all.

Many thanks go to Peter Braspenning, my daily supervisor. He suggested me to take up research on multi-agent systems, and introduced me to the field. With fondness, I recall the many long discussions we had on my research, on the thesis, and on a variety of other topics. He has been an indepletable source of inspiration.

The guidance of Hans Peters has been invaluable for the micro-economic parts of the thesis. Likewise, I am indebted to Jean Derks for his willingness to act as a game theory oracle, and for his comments on the associated portions of the thesis.

Many other people have contributed to the thesis. Talks with David Connah, David Wastell, and Ian Pratt sharpened my understanding of emergent behavior in an early stage of the project. Frits Spieksma supplied the fast linear assignment algorithm that plays an important role in the IG testbed. Nico Roos brought me up to date on computational complexity theory. Discussions with Hans van der Stel and Jeroen Kuipers furthered my understanding of game theory and social choice theory. Talks with Amy Lansky, Barbara Hayes-Roth, Bernardo Huberman, Tad Hogg, Colin Williams, and Moshe Tennenholtz helped me to refine my initial conception of Informed Gambling. A talk with Kurt Konolige exposed important white spots in my ideas about the evaluation of mechanism performance. Questions posed by Koos Vrieze prompted me to investigate the gametheoretic properties of Informed Gambling.

Many people provided relevant literature, literature pointers, or valuable advice. I thank, next to those already mentioned, Inge Uljee, Eric Postma, Fred Wan, Dennis Breuker, Inge Lemmens, Floris Wiesman, Rui Ferreira, Peter Andras, Ruud van der Pol, Evgueni Smirnov, Jan Paredis, Ronald Westra, Tim Grant, Antoon Kolen, Innes Ferguson, Michael Masuch, Phil Agre, Jonathan Schaeffer, Voijn Zivojnovics, and Frank Mulder for their attentiveness.

I am grateful to my colleagues at the Computer Science Department for the pleasant and stimulating working atmosphere they helped to create. In particular, I thank Eric Postma and Leo Plugge for their moral support in difficult times. In the same spirit, I express my gratitude to Ben van Wersch.

Next to wetware, fully functional hardware and software is a *sine qua non* for computer science research. In this respect, I thank Jo Beerens, Patrick Schoo, Johan Thole, Harm Bakker, and — in particular — Maarten van der Meulen for their help with installation and maintenance, at work and at home.

The support and flexibility (yes, I do realize...) of Joke Hellemons, Marlies van der Mee, and Agnes Schreurs in administrative matters and wake-up calls is greatly appreciated. Special thanks go to Sabine Vanhouwe, whose helpfulness and loyalty went well beyond that of any other person I have ever worked with.

In the private sphere, I wish to express my gratitude to four close friends who had the stomach to endure a not always attentive Ph.D. researcher. Carin Bosch, Joerie Widdershoven, Hans van der Stel, and Daphne Scheiberlich: thank you for your tolerance, your help, and your friendship.

Last but not least, I thank my parents for their unwaning confidence and care.

Jacques Lenting — Maastricht, November 1999

Directions for Reading

The summary on page 375 describes the main topics of the thesis in a coherent manner. It should be understandable for anyone with a basic knowledge of computer programming. For readers who can understand Dutch: De Nederlandse samenvatting op pagina 381 is goeddeels te volgen zonder voorkennis op het gebied van de informatica.

A more detailed survey of the contents of the distinct chapters can be found in Sect. 1.4. Readers interested in a specific topic may find the schema below helpful to get a quick impression of the relevant parts of the thesis. The definition index on page 367 is likely to be useful in this case. It contains entries for terms referring to formal notions, and for frequently used symbols. With respect to the symbol entries, the index itself already provides some clue to their meaning, by listing the term entry associated with the symbol. Indexed terms can be found in the headings of a numbered definition, or in the running text. In the latter case, the first character of the term is underlined in the text. Terms and symbols defined within a numbered definition are represented by **boldface** page numbers in the index.

If you are specifically interested in:	peruse or read sections:
the most important conclusions	(1.2,) 8.2, 8.4
the notion of Informed Gambling	1.1.3, 2.2, 5.6, 5.8
the relation between IG and game theory	1.1.3, 5.3, 5.8, 8.4
the relation between IG and economics	1.1.3, 3.3.8
the inadequacy of Walrasian exchange in markets with indivisible goods	3.3., 4.1, 4.9
adequate notions of agent rationality	1.2, 5.4, 5.6.3, 5.8.2, 6.2.4, 7.5.5-8
the methodology for statistical evaluation of mechanism performance	4.5.1-2, 4.6.1, 6.5-6, 7.8.1-4
application of IG	5.4, (5.6,) 5.9, 8.3-4

Contents

Chapter 1

Introduction

1.1 The Problem Statement

1.1.1 The research questions

The problem statement of the research project[1] that resulted in this thesis was formulated as "to study the impact of architectural and protocol variations on the global performance of a distributed multi-agent system". The primary question associated with this problem statement is not

> How can we solve ... problems with multi-agent systems?

but the more fundamental question

> How do the coordination rules and agent characteristics *influence* the performance of a multi-agent system?

Although the latter question does not refer to any specific problem domain, any attempt to answer it necessarily involves the selection of a domain. Because we have taken care in picking a problem domain of some practical relevance, and conceived a *new* multi-agent approach to tackle this domain, the thesis addresses *both* of the above questions.

1.1.2 The problem domain

Domain requirements

Our choice of problem domain has been guided by two requirements.

[1] NWO/SION project 612-322-014

1. The domain should not be too narrow, so as to allow for generalization of our findings.
2. The domain should have practical relevance.

These requirements have led to the selection of a certain kind of *allocation problems* as our problem domain. The domain offers good prospects for generalization of findings, because allocation can be qualified as a *generic* problem. Many problems that are usually not thought of as allocation problems can be reformulated as such. Examples are constraint satisfaction problems such as the 8-queens problem and map coloring, and constrained optimization problems such as traffic routing and rostering. The practical relevance of our domain lies in the *kind* of allocation problems it involves. We study a subclass of allocation problems associated with discrete optimization problems. While substantial attention has been devoted to the conception of multi-agent frameworks for optimization problems featuring continuous variables (Wellman, 1992; Wellman, 1994b; Huberman & Clearwater, 1995; Ygge, 1998), the solution of discrete optimization problems by means of multi-agent systems is still a largely virgin area. Since discrete optimization problems also tend to be more difficult to solve with classical methods (i.e., by means of a monolithic algorithm) than problems with continuous variables, an effective multi-agent approach would be most welcome here.

Allocation and reallocation

In economic terms, an *allocation problem* concerns the allocation of *commodities* to *agents*.

Commodities can be virtually anything: houses, cars, oil, computing time, financial budgets, taxes, tasks, countries, colors, marriage partners, ... In the following, we use the terms *goods* and *resources* as synonyms for commodities.[2]

In the above context, an agent can be anybody or anything to whom or which goods can be allocated, as long as it is possible to define the *agent utility* as a measure of the satisfaction of an agent with its endowment (i.e., the goods allocated to it). In economics, this measure is often ordinal: Agents are presumed capable of expressing whether they prefer an endowment over some other endowment, but not necessarily capable of stating how strong this preference is. We choose to use a more precise, cardinal notion of agent utility. Quantification of the satisfaction of individual agents allows us to define an allocation problem as the problem of finding a good allocation of goods to agents, where the quality of an allocation is some *function* of the agent utilities. For the purpose of mechanism evaluation we use different functions to

[2]Distinctions between tangible and intangible commodities, and between consumptive and reusable ones are not relevant for our purposes.

evaluate different aspects of solution quality. In most of the thesis, however, we equate allocation quality with the average agent utility.

A *reallocation* problem is an allocation problem in which goods are already allocated initially. If — as we assume throughout the thesis — the cost of performing a reallocation is negligible in comparison with its merits, the presence of an initial allocation makes a difference only if the agents are self-centered and self-ruling. Such an agent will refuse to accept a reallocation, if it prefers its original endowment over the final endowment it would have once that reallocation would be performed. The constraint that a reallocation should be such that no agent loses utility in the process is known as *individual rationality*. In terms of preferences, individual rationality entails that no agent should prefer its initial endowment over its final one.

In economics literature, which devotes ample attention to allocation problems, the term reallocation is rare, but most of the problems labeled as "allocation problems" are, in fact, *reallocation* problems (Takayama, 1985; Hildenbrand & Kirman, 1988; Mukherji, 1990). The lack of distinction between these two problem classes in economic literature is due to the fact that mathematical economics is primarily devoted to the conception of models for markets. Market activity concerns either the exchange (trade) or the transformation (production) of goods — where money is also viewed as a good. In both cases, an economic agent is unable to undertake anything without an initial endowment. In other words, there are no allocation problems here; there is only reallocation.

This is not the case in our computer science context: Some constrained optimization problems translate to *allocation* problems, while others correspond with *reallocation* problems. Yet, the distinction between the two problem classes is not all that important for our purposes either, because any allocation problem can be turned into a reallocation problem by defining an initial allocation. In most cases, random allocation of the available goods, followed by reallocation renders an acceptable solution to the original allocation problem. In view of this, we use the term allocation as an umbrella term that covers allocation as well as reallocation.

A taxonomy of allocation problems

The entire class of allocation problems is too broad for one single solution method to work. Some taxonomy is required to study the performance of allocation mechanisms. Below, we classify allocation problems by distinguishing three different categories of goods. The distinction between divisible and indivisible goods is common in micro-economics (Hildenbrand & Kirman, 1988), while that between typed and untyped goods is our own.

We distinguish three categories of goods:

divisible goods: These goods either do not possess any minimal quantity or are usually traded in large quantities. An example is crude oil, where transaction volumes usually involve many tons, and *always* involve many minimal units (molecules).

typed indivisible goods: Like divisible goods, *typed* indivisible goods have the property that any two items (or, in the case of divisible goods, any two equal volumes) of the same good are completely equivalent (and should therefore have equal prices in a market-equilibrium situation). Typed indivisible goods *differ* from divisible goods in that they are typically traded in small quantities. Examples are the *bottles* of olive oil, *tins* of soup, and *pots* of marmalade you buy at your local supermarket.

untyped indivisible goods: Each instance of an untyped good is, in fact, a unique item. Examples of such goods are works of art, designer clothes, and antique furniture.

Note that (1) divisible goods are always typed, (2) untyped goods are always indivisible, and (3) grace to mass production, most indivisible goods are typed.

Solution procedures for allocation problems are sensitive to the category of goods involved. Walrasian exchange, for example, is well suited for allocation problems involving divisible goods, and it can also cope with untyped indivisible goods (Hildenbrand & Kirman, 1988; Quinzii, 1984). It is, however, inadequate in the face of typed indivisible goods, as we will show in Chapters 3 and 4. Because many constrained optimization problems translate to allocation problems with such typed indivisible goods, these allocation problems constitute a problem domain of practical importance. In the sequel, we refer to a good that is typed and indivisible as a *tool*, and to the associated reallocation problems as *tool reallocation* (TR) problems. TR denotes the set of all TR problems. We will treat the reallocation of untyped goods as a special case of TR. In other words, the problem class TR contains the reallocation problems where all goods are indivisible, and *typically* — but not necessarily — typed. The phrases 'typed market' and 'market with typed goods' refer to allocation problems that feature *at least one* typed good.

1.1.3 Informed Gambling

The Informed-Gambling framework

We use the Informed-Gambling (IG) framework primarily as a vehicle for fundamental analysis. However, because the targeted problem domain (i.e., TR) is a generic domain, with practical relevance, the framework can also serve a practical purpose. It can be used as a basis for the design of multi-agent mechanisms for certain types of discrete constrained optimization problems.

The general idea behind IG is that one can often prevent the onset of anti-social behavior (brought about by the agents' selfcenteredness) by withholding information from the agents. In IG, agents are required to commit themselves to certain courses of action in the face of uncertainty. This feature of IG obviates certain assumptions which are necessary in the context of existing mechanisms where agents do not face uncertainty, such as the assumption of perfect competition in the Walrasian exchange auction. Because the information made available to IG agents is of an aggregate nature, we can employ a notion of rationality that is nearly perfect,[3] and yet computationally feasible, grace to the fact that it is virtually impossible for IG agents to gain from hypothesizing on the behavior of particular other agents. In this respect, IG differs fundamentally from other multi-agent frameworks featuring uncertainty, such as the model of Bayesian games (Harsanyani, 1968; Mertens & Zamir, 1985).

The Informed-Gambling testbed

The specific IG mechanism we have implemented to perform our experiments is a tool *reassignment* mechanism. Reassignment is the special case of reallocation, where each agent possesses one item, which it would like to exchange against one other item. We have confined our experiments to reassignment, because such problems are sufficiently simple to allow for fast computation of an optimal solution,[4] while they are not too simple to allow for some generalization of our findings.

In a sense, tool reassignment is the purest form of tool reallocation. As indicated by the dotted line demarcating TR within RR in Fig. 1.1, it is subject to discussion how far the TR domain extends outward, because the distinction between divisible and indivisible goods is hazy. The criterion for divisibility is not whether the good *is* divisible,[5] but whether it is *acceptable* to treat it as if it were divisible. In the context of allocation, a commodity that is traded in portions of integral units can be regarded as divisible, if an allocation algorithm based on real-number arithmetic, followed by a rounding of the output to integral numbers, renders an outcome of acceptable quality.

Gasoline can be regarded as a divisible good, because no truck driver cares whether there are $3,000,\ldots,000,000$ or $3,000,\ldots,000,001$ molecules in the tank (nor whether there is 0 or 1). Spouses, on the other hand, cannot be treated as a divisible good, not because they cease to be a spouse when cut in half, but because most people — as well as authorities — attach considerable significance to the differences between having no spouse, one spouse, or two spouses.

[3]This entails that the agent's behavior can be described succinctly as 'utility maximization'.

[4]This is required to evaluate IG mechanism performance.

[5]Even the traditional example of a divisible good, oil, is not *truly* divisible, since it is traded in portions that comprise a *natural* number of molecules.

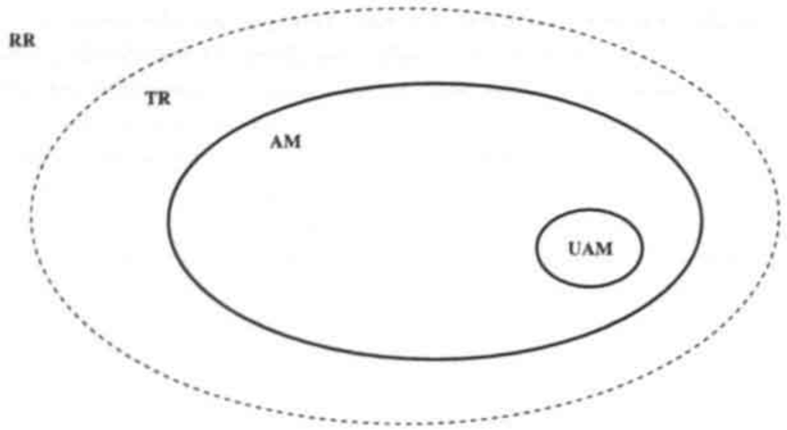


Figure 1.1: Venn diagram of reallocation problem classes:
RR = reallocation of (divisible or indivisible) goods;
TR = reallocation of tools (i.e., indivisible goods);
AM = reassignment of tools;
UAM = reassignment of untyped tools.

Evidently, in intermediate cases where exchange transactions involve the delivery of few, but more than one item, it is less easy to determine whether a good can be regarded as a divisible good. Moreover, it is not feasible to draw a line between TR and RR\TR which is solely based on the average, or minimal trade volume.[6] It can depend on various aspects of the problem domain where the borderline lies. Consider, for example, the problem of distributing 39 comic books over two children. If the children are seventeen year olds or two year olds, you are unlikely to run into trouble if you treat the books as a divisible goods, and round to one of the two nearest integers to attain a feasible allocation. This procedure may cause a small riot, however, if you are dealing with four-year olds who have just learned to count to twenty... (E. Postma, 1998, personal communication).

Such considerations do not play a role in reassignment problems. In a reassignment problem, downward rounding of the number of goods to be delivered amounts to no delivery at all. This is not acceptable in any realistic context. Consequently, there can be no discussion about the indivisibility of goods in the reassignment domain.

[6]The operator \ in RR\TR denotes set difference.

1.2 Basic Terms

Due to the fact that the research area of multi-agent systems is broad and young, there tends to be considerable terminological inconsistency between authors. Hence, to prevent misunderstanding, we explicitize the semantics that we assign to some basic terms.

computational entity A computational entity is a coherent piece of software that allows for a surveyable description of its inputs, outputs, and functionality within a software system.

agent An agent is a computational entity whose output can be interpreted as the outcome of a decision-making process determined by the agent's goals, its computational capabilities, and its knowledge of the situation.

multi-agent system A multi-agent system is a system that involves interacting agents.

self-centered A self-centered agent is an agent that is not sensitive to the needs of other agents, and not responsive to explicit requests for help, unless it is in the agent's own interest. Its behavior is determined purely by its sense of personal profitability.

self-sufficient A self-sufficient agent is an agent that can perform its tasks without supervision or aid from other (human or artificial) agents.

self-ruling A self-ruling agent is an agent that cannot be *commanded* to behave in some desired manner. It may act in conformance with a request, but only if this does not conflict with its own internal state (i.e., its goals and information).

autonomous An autonomous agent is a self-ruling agent.

tropistic An agent is tropistic if its behavior is based solely on observation of the *present* situation.

hysteretic An agent is hysteretic if its behavior is based on present observations, and recollection of past observations.

prospective An agent is prospective if its behavior is based on present observations, and contemplations with respect to the future.

contemplative A contemplative agent is an agent that is hysteretic as well as prospective, that is, its behavior is based on present observations, recollection of past observations and contemplations with respect to the future.[7]

[7]This is approximately concomitant with the notion of knowledge-level agent as defined in (Genesereth & Nilsson, 1987).

coordination Coordination entails the efforts needed to ensure that the global performance of a multi-agent system is satisfactory, despite the fact that most or all of the decisions are made by self-centered agents, on the basis of incomplete information.

protocol A coordination (or interaction or negotiation) protocol is a description of the interaction among agents or between agents and other computational entities in terms of message response.

coordination module A coordination module is a computational entity that is incorporated in a multi-agent system to coordinate the behavior of individual agents.

mechanism A (multi-agent) mechanism is a multi-agent system that features a coordination module, rather than a form of coordination that emerges from direct interactions between agents.

weak informational decentralization A *mechanism is weakly informationally* decentralized if the information which defines a problem instance is initially distributed over the agents, and at least some of this information remains private during the entire interaction process.

informational decentralization A mechanism is (strongly) informationally decentralized if

(i) it is weakly informationally decentralized, and

(ii) it does not require that agents have a common representation formalism for their private information.

operational decentralization A mechanism is operationally decentralized if the computational and design complexities are distributed evenly over the agents and the coordination module, to prevent the coordination module from becoming a bottleneck in computation or design.

An example of a mechanism that is weakly, but not strongly informationally decentralized is a monetary auction. The functionality of such a mechanism hinges on the assumption that every agent is capable of expressing its private information (in this case: utilities for different items) in terms of a *common* vocabulary (in this case: currency). The Informed-Gambling framework involves strongly decentralized mechanisms.

1.3 Our Position within Computer Science

1.3.1 Distributed AI and multi-agent technology

This thesis is an account of research performed in the field of Distributed Artificial Intelligence (DAI) (Huhns, 1987; Bond & Gasser, 1988b; Gasser & Huhns, 1989), nowadays often referred to as Multi-Agent Systems (MAS). While Artificial Intelligence is mainly concerned with modeling or implementing individual intelligence, DAI focuses on *group* intelligence. DAI researchers are interested in the *emergence* of group intelligence (i.e., collective behavior which is adequate from a global perspective) from relatively simple interactions between computational agents.

DAI is a broad research area. It is concerned with disparate subjects such as distributed constrained optimization and the design of agent communication languages. One used to partition the DAI field into the subfields Distributed Problem Solving (DPS) and Multi-Agent Systems (MAS) (Bond & Gasser, 1988a), but the distinction between DPS and MAS is far from clear,[8] and nowadays MAS tends to be used as a synonym for DAI. In view of the broadness of the DAI field, some partitioning is desirable, however. The partitioning I propose is normative, and involves two dimensions. The first of these concerns the distinction between Multi-Agent Modeling (MAM) and Multi-Agent Technology (MAT). MAM comprises MAS research that is motivated by the desire (or driven by the habit) to develop satisfactory models for collective or social *human* intelligence. Usually, the researchers in this area pay little attention to computational efficiency or technological applicability, and statements such as "AI is a science, not a technology" (Castelfranchi, 1997) indicate that this is — at least for some researchers — a matter of principle rather than the state of the art.

The ultimate aim in MAT is to develop useful technology. Developing concepts which are satisfactory from a scientific point of view can play a role here, but only inasmuch as this is relevant for technological development. Our own position is within MAT.

1.3.2 Open versus closed multi-agent technology

In turn, MAT can be subdivided into research on closed systems and research on open systems. In closed multi-agent systems, all agents are fully known system com-

[8] As an example, (Kraus *et al.*, 1995) define DPS as the research area where system designers have full control of the behavior of all agents, and MAS as the complementary area, where at least some agents are not under the designer's control. Subsequently, they classify the work of Wellman (1992) as MAS research, where — according to their own definition — it does not belong. The cause of this inconsistency lies with their overeager conclusion that, since DPS agents *can* be designed to strife for a global instead of a personal goal, they *will* be designed as such.

ponents. In open systems, there may be interaction with unknown, external agents. Closed Multi-Agent Technology (CMAT) entails the design of *entire* multi-agent systems capable of solving a concrete computational problem via agent interaction. This includes the research area that used to be referred to as DPS. The primary motivation for CMAT is the fact that the design complexity of a multi-agent system tends to be lower than that of a functionally equivalent, monolithic algorithm. Hence, CMAT may enable us to provide acceptable solutions for problems which defy solution by means of classical algorithms, or to lower the cost of software development in domains that are principally susceptible to both programming paradigms. The attitude of CMAT researchers tends to be pragmatic rather than principled. Examples of CMAT are the design of distributed blackboard systems for sensor data interpretation (Durfee, 1988), and the design of artificial markets for transportation planning (Wellman, 1994b).

In contrast, Open Multi-Agent Technology (OMAT) involves the design or analysis of *part* of a multi-agent system which is not specified (or even specifiable) as a whole. This part can be a coordination protocol for a specific environment like the Internet (Decker *et al.*, 1997), or a more general protocol suited for a variety of existing, autonomous agents (Rosenschein, 1993), an agent communication language (Finin *et al.*, 1996), or it may concern an abstract conceptual framework for open multi-agent technology in general (Hewitt, 1991). Because much of the research in OMAT is rather abstract, and not bound to any specific application context, design principles of a general, often philosophical nature play a dominant role here.

There exists a (large) twilight zone between OMAT and MAM of research literature which many researchers regard as relevant for MAT, but which is based on concepts that, in my view, are better suited for MAM, or — in some cases — not suitable for MAS at all. These authors do not provide an explicit motivation for the ontology they employ or propose, nor do they refer to other authors who provided such a motivation. Their ontologies seem to stem from inarticulate intuitions on what the essential concepts are in (human) information processing and social interaction. Examples are (Cohen & Levesque, 1987; Werner, 1989; Shoham, 1993).

While our methods and views primarily stem from CMAT, our work does have implications for OMAT. This is due to the fact that, in designing IG, we took care to ensure that IG agents act in a way that permits viewing them as agents which are self-ruling and self-centered. In retrospect, the dual goal of conceiving an effective CMAT mechanism that is also useful for OMAT has led us into a lot of dead ends. Ultimately, however, it did render a framework that is principally suited for both research areas. Within CMAT, the IG framework can serve as a basis for the design of a specific IG mechanism for a specific type of constrained optimization problem, and within OMAT, the IG protocol can be used to reallocate tools between existing, autonomous agents. More importantly, the dead ends we ran into render valuable insight in the limitations of popular OMAT concepts such as Pareto optimality, and various notions of agent rationality.

Though relevant for OMAT, the Informed-Gambling framework is more akin to constrained optimization techniques outside DAI — such as simulated annealing (Aarts & Korst, 1989), and Boltzmann neural networks (Lenting, 1995) — than to many areas in OMAT. In particular, our work has little kinship to or relevance for DAI research on agent communication languages such as KQML (Finin *et al.*, 1996), or DAI frameworks in the twilight zone between OMAT and MAM.

There is, of course, a relationship between our work and earlier DAI research on allocation (Miao *et al.*, 1988; Kurose & Simha, 1989; Lumer & Huberman, 1990; Sycara *et al.*, 1991; Kuwabara & Ishida, 1992; Kraus *et al.*, 1995). The DAI approach most akin to Informed Gambling in this respect is the CMAT framework of Market-Oriented Programming (MOP) (Wellman, 1992; Wellman, 1994a; Wellman, 1994b; Huberman & Clearwater, 1995; Ygge, 1998). Like a MOP agent, an IG agent acts like a *homo economicus* who attempts to increase its satisfaction by exchanging goods with other agents, and both frameworks involve centralized markets: Agents do not interact directly, but via one or more auctioneers. However, due to the fact that MOP is primarily targeted at reallocation of *divisible* goods, while IG was designed for tool reallocation, the two frameworks differ substantially. The most important differences are listed below.

- In MOP, the agents perform simple, deterministic calculations based on price information, while in IG, agents perform more complex, probabilistic computations based on information on supply and demand.
- IG agents act as entrepeneurs. They must take (calculated) risks to improve their situation, whereas MOP agents do not face any uncertainties.
- In MOP, solutions are approximations of a market equilibrium (where supply equals demand for all goods), and the search for a solution is based on the economic method of tâtonnement, which resembles gradient descent. In contrast, IG solutions are approximations of a correlated equilibrium (where every agent sticks to its last bid), and the search process is reminiscent to that of simulated annealing.
- MOP involves an artificial economy with traders, producers, and money. IG involves neither money nor producers. It features only barter trade.

The absence of money in IG is not just a coincidental feature, but a deliberate constraint that we impose throughout the thesis. The motivation for this constraint stems mainly from the desire to arrive at a framework that is not only relevant to CMAT, but also for OMAT.

In a CMAT framework, artificial money can be introduced without undesirable side effects, because the system designer can define the behavior of the agents completely. In an OMAT context, however, the designer of a protocol or an agent does not have any authority over existing agents. To constrain the spectrum of possible agent responses in an open system, one usually employs the working hypothesis that all agents will behave as self-ruling, self-centered utility maximizers. Since such agents will refuse to attach any value to artificial money, one would have to use real money in an open system. This has several disadvantages.

First, the *effectiveness* of real money as an instrument of agent coordination can be problematic, because it hinges on the assumptions that

(i) any absence of goods can be compensated with money;

(ii) every agent possesses enough money to compensate any other agent, if so desired.

The combination of these two assumptions is known as the assumption of transferable utility. It is often criticized by economists as being too strong. A concrete example of a case in which the transferable-utility assumption is inadmissible concerns the air-traffic control domain: No amount of money can compensate an airplane (crew) for being repeatedly denied an approach corridor while the plane is running short on fuel. More generally, any problem with hard constraints is troublesome in this respect.

A second disadvantage of reliance on money in the context of open systems concerns the fact that a self-centered agent may behave in an anti-social manner. A good example is hoarding. While this can also occur in markets without money, it tends to be much more difficult to gain from hoarding in a barter-trade economy.[9]

1.4 Thesis Overview

1.4.1 Tool reallocation problems

Chapter 2 comprises the specification and representation of tool reallocation (TR). After an informal sketch of our prototype TR problem, we present some examples of constraint satisfaction and constrained optimization problems that can be reformulated as TR problems. This illustrates the genericity of the TR problem domain.

Our *representation* of reallocation problems differs from the representation commonly used in game theory and economics: instead of vectors, we employ bags

[9] This is especially so for frameworks such as IG, where *uncertainty* is used — instead of money — to ensure that cardinal utility differences are reflected in the final outcome, even though utilities are not communicated.

(a.k.a. multisets) as basic constructs. The bag is a better representation primitive for TR than the vector, because TR problems are essentially discrete, and can be represented more concisely in terms of bags.

Our formal *specification* of TR problems is distributed in the sense that the *solutions* of TR problems are defined in terms of exchange proposals rather than the underlying agent utilities. However, the solution quality is defined as an aggregate of the utilities obtained by the agents. As such, TR problems are specified as distributed problems, seen through the eyes of an omniscient external observer. The key part of this problem specification is our relative-utilitarian viewpoint. This entails that the utility obtained by an agent expresses its degree of *satisfaction* rather than its profit. This viewpoint is the basis of our evaluation of IG mechanism performance.

1.4.2 Distributed approaches to TR

Chapter 3 comprises an analysis of the prospects for distributed approaches to TR. This analysis serves as a first step toward the conception of Informed-Gambling mechanisms. Our goal has been to conceive a multi-agent approach to TR that is more distributed than constraint-directed negotiation (CDN) (Sathi & Fox, 1989), where agents interact via a mediator agent who does most of the computation.

First, we explain why we refrain from considering the possibility to solve TR problems in a *fully* decentralized fashion, that is, without employing any coordinator center(s): While coordination through local interaction is feasible for some problem domains, it is too inefficient for domains which require multilateral coordination, such as the TR domain.

Subsequently, we search for a form of centralized coordination in which the central coordination module does not constitute a bottleneck, as it does in CDN. Inspired by Sathi and Fox (1989), we consider two orthogonal strategies for TR: proposal composition and proposal relaxation. Composition can be viewed as a bottom-up approach. It entails the incremental construction of a solution which satisfies a subset of the submitted proposals. Relaxation is a top-down approach. It comprises the adaptation of proposals to arrive at a set of proposals that are jointly satisfiable. In the CDN approach described in (Sathi & Fox, 1989), these two approaches are intertwined. We consider them separately.

The main result of our analysis of composition-based reallocation is the following. We prove that the general composition problem is NP-hard, and show that a restriction of the exchange proposals to elementary ones (i.e., one-for-one exchanges) renders a composition problem that can be solved in polynomial time. This implies that an approach which allows only elementary proposals constitutes a more distributed form of mediated negotiation than that embodied in CDN: It turns the composition task of the mediator into a tractable problem.

In CDN-style *relaxation*, the mediator tries to persuade agents to accept specific proposal adaptations. Because the mediator is bound to constitute a bottleneck in such an approach, we investigate another style of relaxation, in which the mediator only provides incentives for proposal adaptation. Such a style of relaxation is used in the Walrasian exchange auction. This mechanism lies at the root of the extensive economic General Equilibrium Theory, which — in turn — has been used as the basis for WALRAS, a MOP system that aims to solve constrained optimization problems by searching for a competitive equilibrium[10] in the associated artificial markets (Wellman, 1992; Wellman, 1994b).

We identify two drawbacks of Walrasian exchange: (1) meagre solution quality and (2) frequent absence of Walrasian equilibria.

The solution-quality drawback of walrasian exchange is a consequence of the absence of money in combination with the indivisibility of the goods. In the context of divisible goods, competitive equilibria have the agreeable property that the associated allocations are not only Pareto optimal, but also optimal in a utilitarian sense (Ygge, 1998, Chapter 3). This is true even in markets with only one divisible good, provided that this good is possessed and appreciated by all agents, so that it can be used as money. Unfortunately, it is *not* true for Walrasian exchange in TR markets. We provide an example which shows that the utilitarian quality of Pareto-optimal allocations can, in fact, be *arbitrarily low* in this case. We also demonstrate that the addition of money leads to a different equilibrium allocation, which is optimal in the utilitarian sense.

In a pragmatic sense, the second drawback extends beyond Walrasian exchange, to General Equilibrium Theory. Whether competitive equilibria exist or not, it appears to be hard to approximate them in reallocation problems featuring indivisible goods. As a case in point, we mention the attempt to apply MOP to configuration design (Wellman, 1994a). Wellman attributed the failure of this attempt to the fact that certain conditions for equilibrium existence in micro-economic theory, such as convexity of the agent-utility functions, can not be satisfied (or even defined) in discrete problem domains.

Indeed, theorems on the existence of Walrasian equilibria in markets with indivisible goods are rare, and pertain almost exclusively to markets with money. The only theorem that applies to barter-trade markets concerns the *reassignment* of *untyped* goods. In other words, micro-economic theory does not tell us anything about equilibrium existence in *TR markets* (i.e., barter trade markets involving reallocation of *typed* indivisible goods).

[10] The notion of competitive equilibrium is a generalisation of that of Walrasian equilibrium to markets with money and/or production.

1.4.3 Walrasian equilibria in TR markets

In Chapter 4, we combine theoretical analysis with experimental investigation to assess how grave the aforementioned problem of equilibrium absence is for TR markets. Our approach comprises three steps.

1. First, we conceive a fast algorithm to verify the existence of a Walrasian equilibrium for certain typed reassignment problems, and prove its validity.
2. In our experimental investigation, we employ this algorithm to estimate the percentage of markets that do have Walrasian equilibria for various kinds of tool reassignment markets. The outcomes reveal that a Walrasian equilibrium *seldom* exists in such markets, except for markets with very few (say, at most 4) agents, or markets that are almost untyped (e.g., if all but one of the goods are untyped, and there are only two instances of the typed good).
3. Finally, we explain why one can expect that the problem of equilibrium absence is of comparable gravity in markets that constitute full-fledged reallocation problems (rather than reassignment problems).

1.4.4 The framework of Informed Gambling

Chapter 5 comprises the definition and theoretical analysis of Informed-Gambling mechanisms. IG mechanisms are introduced in three steps.

1. First, we develop the enveloping notion of iterative mechanism, which has its roots in mathematical economics (Maskin, 1985; Postlewaite, 1985; Myerson, 1985). In an iterative mechanism, agents repeatedly communicate with a central coordination module, until some termination condition is satisfied. This condition, as well as the final outcome, is a function of the messages last sent by the agent.
2. Subsequently, we highlight the basic ideas behind IG by discussing its precursor, delegated negotiation (DN). In view of the conclusions drawn in Chapter 3, DN agents can only submit *elementary* reallocation proposals, that is, one-for-one exchanges and unconditional offers or requests that involve only one tool type. The key concept in DN — as well as in IG — is that of *commitment under uncertainty* as a means to prevent anti-social group behavior within an agent community. Commitment under uncertainty entails that agent runs a calculable risk of losing a tool it offered in exchange for a tool which it did *not* ask for. This auction rule provides an incentive for rational agents to relax their demands.

   In DN, the agents do not get the opportunity to *adjust* their proposals to information updates. After they have submitted their elementary exchange proposals, a partial reallocation immediately takes place. If some of an agent's proposals are rejected, the agent may lose the tools offered in these proposals in exchange for tools for which it has little use. Its only respite is a second chance to exchange such tools for other tools of agents that suffered a similar fate.

3. The third part of Chapter 5 comprises the formal definition of Informed Gambling as a class of iterative mechanisms, where agents repeatedly adjust their proposals until a quiescent state emerges.

In the remainder of Chapter 5, we provide a detailed (i.e., implementable) specification of an IG mechanism for reassignment. We analyze its game-theoretic properties, characterize the rationality of its agents, and demonstrate its effectiveness in preventing the onset of certain anti-social agent behavior. We also identify a problem with the termination of its relaxation process, and propose to use *negotiation weariness* as a remedy.

Finally, we explain how we envision the application of IG to real-life problems as closed and open coordination mechanisms respectively.

1.4.5 Testbed experiments

Chapter 6 consists of two parts. In the first part, we describe the testbed that has been used to experiment with different variants of Informed Gambling. The second part comprises a survey of the experiments that were performed using this testbed, and a discussion of the employed methodology.

1.4.6 Experimental results

In Chapter 7, we present and analyze the experimental findings. These concern the performance of the IG reassignment mechanism defined in Chapter 5 on various types of reassignment problems, the variations in performance that can be obtained by changing the behavior of the agents, or the coordination rules, and a performance comparison with other, Walrasian-like mechanisms.

1.4.7 Conclusions, reflection, and future research

Next to a summary of the most important conclusions, the final chapter comprises a reflection on the implications of our research for fundamental research on MAS in general, and some suggestions for future research.

The difficulties encountered in trying to come up with an efficient mechanism for TR that can be used in open as well as closed systems has provided some insight which we believe to be useful for fundamental MAS research, which is often directed toward open systems, and not targeted toward any specific domain. We discuss the implications of our findings and failures for reigning views on agent rationality, agent autonomy, and the value of gametheoretic concepts such as Pareto optimality, and strategy proofness.

The suggestions for future research concern the application of IG to real-life optimization problems, as well as some fundamental issues. Future fundamental research concerns the suitability of IG's agent models for open systems, while our discussion of IG's application prospects entails the formulation of properties a real-life domain should have in order for IG to be a promising approach.

Chapter 2

Tool Reallocation Problems

2.1 Informal Description of the TR Domain

2.1.1 Our prototype TR problem

The problem for which Sathi and Fox (1989) conceived their Constraint-Directed Negotiation (CDN) approach has played a crucial role in our conception of the TR problem domain. CDN was developed to solve a reallocation problem that arises in a software-engineering company where different teams work on different projects. Each team has one or more resources at its disposal for the duration of a project. Resources, in this context, are computer workstations, and each agent represents a project team. The company possesses different types of workstations. Because different projects tend to require a different mix of workstation types, a reallocation problem arises whenever a set of new projects is initiated.

Our perception of the tool reallocation domain is a generalization of the above problem. It is described informally by the following prototype problem.

> A number of agents, each owning one or more tools which they use in dealing with a continuing stream of tasks confronting them, decide to cooperate by combining their respective tools into one community pool. From then on, they reallocate tools between them whenever it seems appropriate to do so. To determine the appropriateness of reallocation, the community maintains a statistic that measures the effectiveness of the current tool allocation as well as the improvement that can be expected from reallocation. Reallocation is attempted whenever this statistic drops below a certain treshold.

We describe the prototype TR problem as a recurring problem, because this permits us to weaken the agents' self-centeredness without tampering with their identity as *rational* agents. This is important in open systems in particular. Here, one usually employs the working hypothesis that all existing agents behave rationally, that is, as self-centered utility maximizers. While this is a plausible and concise characterization of the behavior of existing agents, the implied taboo on agent benevolence complicates the task of conceiving an effective form of coordination considerably. This setback can be mitigated by extending the temporal horizon of the agents. Income taxes used to finance support for the elderly would meet much more resistance if twens were unconscious of the fact that they too will grow old. In this sense, behavior that is benevolent — and hence irrational — in the short run, can be rational in the long run.

We make some assumptions with regard to tool characteristics and reallocation cost that are not mentioned explicitly in the above description of our prototype problem.

(i) Tools are indivisible.

(ii) Tools are reusable.

(iii) The cost of the physical reallocation of tools is negligible in comparison with differential tool utilities.

The criterion for tool reusability is whether the resource can be used a second time without undoing the task that it helped to accomplish. Hence, nails and screws are not tools, while execution time and memory space are. Note that reusability is, in fact, already implied in the concept of recurring reallocation: It is difficult to imagine such problems in the context of consumptive resources.

The last assumption is purely a matter of simplification. It ensures that physical reallocation costs need not be taken into account when deciding what tools are most appropriate in view of current tasks.

2.1.2 The problem context: closed vs. open systems

In Chapter 1, we distinguished between multi-agent technology for open systems (OMAT) and for closed systems (CMAT) as the two areas within MAT. The above description of our prototype TR problem aims to cover both areas.

If one thinks of the agents in the description as computational front ends for human agents, there are principally two possibilities. Either the computational agents act as brokers for their human clients, collecting the necessary information on the client's desires before engaging themselves in a negotiation with other computational agents, or there is ongoing interaction between computational agents and human clients while the clients' problems are being solved. In both cases — and especially so in the interactive case — it is appropriate to regard the computational agents as autonomous and self-centered. In other words, we are dealing with an OMAT context.

However, there is nothing that forces us to regard the agents as front ends for human computer users. The agents can represent something entirely different, if the prototype problem is a CMAT problem. In a CMAT context, the notion of agent is much broader. Typically, the only property that a computational entity must have to qualify as a CMAT agent is that it either knows the utilities of all conceivable endowments, or has the capability to compute these utilities. Hence, in CMAT, an agent can be *any* construct to which a utility for tools can be attributed. As such, it can represent the interests of a team of programmers working on a project, but also those of an airplane approaching an airport, a queen seeking for a suitable position on a chessboard, or even a color that 'desires to be assigned' to objects. Examples of these last two cases are presented in the next section.

2.1.3 The genericity of TR

Allocation and reallocation are generic problems. Many problems that are usually not regarded as (re)allocation problems can be reformulated as such. Below, we present three examples of CMAT problems that can be reformulated as TR problems. The first two of these are constraint satisfaction problems, while the third is a constrained optimization problem. Note that we only provide a *formulation* of the respective problems as a (re)allocation problem. Effective *solution* of the problems by means of a distributed reallocation algorithm is another matter. We address this issue later, in Chapter 8, when we characterize the kind of problem domains for which an investigation of the applicability of our Informed-Gambling framework would make sense. However, the actual application of IG or other reallocation mechanisms to real-life problems is outside the scope of this thesis.

Example 2.1 (Map coloring)
A map coloring problem is depicted in Fig. 2.1. The problem constitutes the coloring of the $N = 8$ countries on the map, using at most 4 colors, such that no two neighboring countries have the same color. This can be reformulated as an allocation problem by associating each of the 4 colors with an agent, and viewing each country as a resource that must be allocated to an agent. Define the satisfaction of an agent with its endowment as the number of countries in the endowment which are not adjacent to another country in the endowment, minus N times the number of pairs of neighboring countries in the endowment. This penalty for possessing neighboring countries ensures that any endowment with neighboring countries in it is less satisfactory than the empty endowment. Define the solution quality as the average agent satisfaction. This amounts to the total number of allocated countries divided by the number of agents. Hence, the quality of an allocation is bounded from above by $8/4 = 2$ in this case, where the maximum of 2 can be attained only if all countries are allocated. Since every map can be colored with 4 colors, such a total allocation

always exists. Consequently, any allocation of maximal solution quality corresponds with a solution of the map coloring problem. △

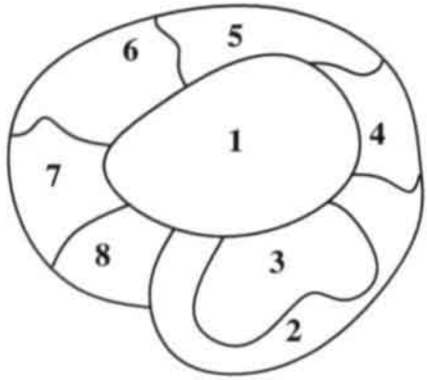


Figure 2.1: A map coloring problem.

The above example reformulates the map coloring problem as an allocation problem featuring untyped indivisible goods, and cardinal agent utilities.

Example 2.2 (8 queens)
The 8-queens problem can be expressed as "Put 8 queens on a chessboard such that no two queens attack each other, that is, share a row, column, or diagonal". This problem can be translated into an allocation problem by associating an agent with each queen, and considering the rows, columns and diagonals as resources that can be allocated to an agent. If the rows and columns are numbered from 1 to 8, each of the 15 downward diagonals can be described as $\{(c, r) \mid r + c = k\}$*, where* r *and* c *denote row- and column-numbers respectively, and the values* k *are used to number the diagonals (*$k = 2, \ldots, 16$*). Likewise, each of the 15 upward diagonals is defined by* $\{(c, j) \mid r - c = l\}$*,* l *ranging from* -7 *to* 7*. Associating each of the 8 columns with an agent, agents should try to get hold of resource quadruplets* (r, c, k, l) *that* match, *that is,* $r + c = k$ *and* $r - c = l$*. In essence, this comes down to stating that an agent must* possess *the row, column, upward, and downward diagonal associated with a square before it can place a queen there (see Fig. 2.2). Define the satisfaction of an agent as 1 if the agent possesses four resources, which form a matching quadruplet. Otherwise, the satisfaction is 0. Define the global solution quality as the average of the agents' satisfactions. Finding a solution to the 8-queens problem is now tantamount to finding an allocation of quality one.* △

The above example reformulates the 8-queens problem as an allocation problem featuring untyped indivisible goods. Although the agent utilities are defined as cardinal

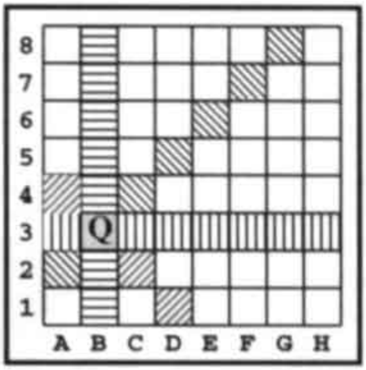


Figure 2.2: To place its queen on square B3 an agent must possess column B, row 3, downward diagonal A4-D1, and upward diagonal A2-G8.

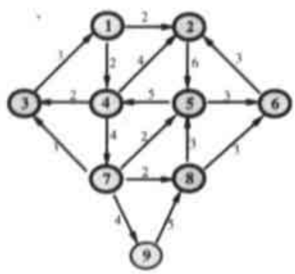
functions in the example, it should be clear that they can be defined ordinally also. In essence, agent satisfaction is a boolean variable here.

Example 2.3 (Escherville traffic routing)
The village of Escherville is a popular mountain holiday resort for the elderly. The nine châlets that make up the village are relatively far apart, and though mountain paths between them exist, these are too steep for most of the visitors. Fortunately, the châlets are also connected by a network of mountain streams, which are utilized by the local canoe taxi company to provide transportation services. Canoes of varying size (2-5 passengers) are stationed near the châlets and each taxi peddler operates his or her canoe on its dedicated connection, ferrying downstream, and carrying the canoe back (along the path) afterwards.

The network is drawn in Fig. 2.3, together with an example travel list. The numbers next to a connection denote the capacity of the canoe that is used to provide service over the connection. The numbers in the circles are the housenumbers of the châlets.

The taxi company has an opening for a traffic router, whose task will be to determine a passenger routing schedule for the next day on the basis of the travel list for that day. The schedule should minimize the average number of canoe changes per passenger. △

The above example describes an allocation problem with typed, indivisible resources. Here, the passengers are the agents, the connections between châlets are the resource types, and the number of instances of each type equals the capacity of the associated connection. An agent's utility for a resource bag equals zero if the resources do not constitute a path from its source to its destination, and M minus the length of the



Travel list:

From:	To:	Passengers:
1	5	2
3	4	2
5	3	2
8	3	1
8	2	3
7	6	2

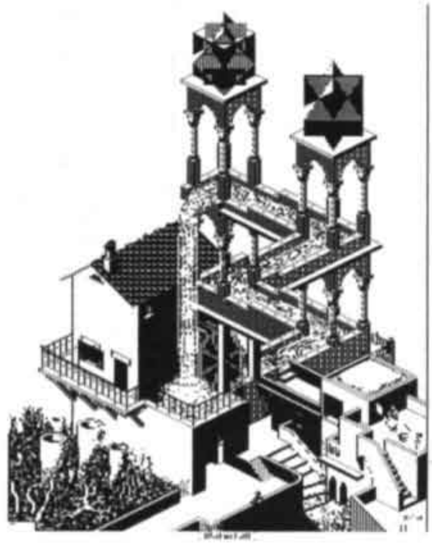


Figure 2.3: The Escherville transportation network.

path, otherwise. Here, M can be any sufficiently large number (i.e., the length of the longest path plus one). The global utility equals the average agent utility.

2.2 Formal Representation of TR Problems

2.2.1 Representation as the first step to solution

Although one usually views problem specification, representation, and solution as distinct activities, the choices one makes when conceiving a representation and specification for a problem inevitably influence the solution method.

In the present case, this pertains to the fact that our representation framework involves reallocation *proposals*. This reflects our choice in favor of a solution method that comprises a negotiation (i.e., message exchange) process between agents and a coordination module, where the agents communicate proposals rather than utilities. While this choice stems primarily from our aim to conceive an approach to TR that features a high degree of operational decentralization, there are, in fact, multiple reasons to opt for the communication of proposals instead of utilities.

communication constraints Communication of *all* utilities or preferences may consume too much bandwidth or time.

privacy constraints Agents may not be willing to reveal their utilities.

bounded rationality In many practical situations, human or artificial agents do not possess full, explicit knowledge of their utilities or even preferences. Often, an agent is only aware of the alternative it likes best, and a few others that come close.

simplification Imposing constraints on the information that is communicated is liable of being detrimental to solution quality in the sense that the *communicated* information may be insufficient to guarantee the *optimality* of a solution. However, the task of finding a solution that seems acceptable, *in view of* the information that is available, will generally constitute a problem that is considerably *simpler* than the overall problem facing an omniscient problem solver. As such, a distributed approach featuring information hiding may in fact lead to *better* solutions than a centralized one, in cases where the requirement of solution *optimality* (rather than acceptability) would incur too much computation cost anyway.

2.2.2 Bags as representation primitives

In economics, the endowments of agents involved in reallocation problems are referred to as "commodity bundles", and represented as *vectors* in $\mathbb{R}^k$. Such a representation is less apt for indivisible goods. The natural representation for a bundle of indivisible goods is a tuple of nonnegative integer numbers, that is, an element of $\mathbb{N}_0^k$ (where $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$). Representation and reasoning about indivisible goods is easier if one thinks of portions of such goods as *bags*. The notion of bag (or multi-set) is not used in mathematical economics, but fairly common in computer science. A bag is similar to a set, except that the identity of a bag is not determined by the elements it contains, but the *number* of times an element occurs in a bag. As such, it is a useful notion in any context that features objects that are equivalent in some sense. Tool reallocation problems constitute an example of such a context, because a tool is a *typed* good. It does not matter whether an agent has hammer 1 or hammer 2 at its disposal, as long as their task-essential properties (shape, material, weight) are the same. For this reason, we represent the tool endowments of agents as *bags* of tool *types*, rather than sets of tools.

In its formal definition, we identify a bag with its indicator function. While the indicator function of a set is binary-valued, indicator functions for bags can take on any nonnegative integer value. To support the reader in distinguishing bags from sets, we denote bags with Greek and sets with Latin characters. $\mathbb{N}$ denotes the set $\{1, 2, 3, \ldots\}$ of natural numbers, and $\mathbb{N}_0$ denotes the set $\mathbb{N} \cup \{0\}$. The abbreviation iff stands for "if and only if".

Definition 2.4 (bag, domain, multiplicity, carrier, empty bag)

Let S be a nonempty set. A bag β over S is a function

$$\beta : S \to \mathbb{N}_0$$

S is referred to as the domain of β. For $x \in S$, $\beta(x)$ is called the multiplicity of x in β. The subset $\breve{\beta} = \{x \in S : \beta(x) > 0\}$ of S is called the carrier of β. If the domain of β is apparent from the context, β can be specified by listing the elements of $\breve{\beta}$, with their respective multiplicities. In such cases, we use D_β to denote the domain of β. $\emptyset$ denotes the empty bag (or set). △

Definition 2.5 (bag size, finite bags)
Let β be a bag over the nonempty set S. Then the size of β is

$$|\beta| \triangleq \sum_{x \in S} \beta(x)$$

A bag is called finite if it has finite size. The set of all finite bags over a domain S is denoted by $\mathfrak{B}(S)$. △

In this thesis, we only consider finite bags. Note that the domain of a finite bag β can be infinite; only its carrier $\breve{\beta}$ must be finite.

If S is finite, $\mathfrak{B}(S)$ can be identified with $(\mathbb{N}_0)^{|S|}$, the set of $|S|$-tuples of nonnegative integer numbers. Such a tuple representation is common in game theory. However, small bags can be represented more concisely by enumerating their contents. This is especially so if the carrier of a bag is much smaller than its domain. Suppose, for example, that we are dealing with a bag of two type-a tools and one type-w tool in the context of a reallocation problem with tool types ranging from a to z. We can denote this bag by enumeration, either as $\{a, a, w\}$, or as $\{a : 2,\ w\}$. This is much more concise than the tuple notation used in economics, which would come down to

$$(2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0)^T \tag{2.1}$$

The element-of relation is defined for bags as $x \in \beta \Leftrightarrow x \in \breve{\beta}$. With respect to *indexing* over a bag, we remark that an expression like $\sum_{x \in \beta} f(x)$, with $\beta = \{a, a, w\}$, and f some function on D_β, should be interpreted as $f(a) + f(a) + f(w)$, rather than $f(a) + f(w)$.

Definition 2.6 (indexing over a bag)
Let f be a function on S, for example, $f : S \to \mathbb{R}$, and $\beta \in \mathfrak{B}(S)$. Then

$$\sum_{x \in \beta} f(x) \triangleq \sum_{x \in S} \beta(x) \cdot f(x)$$

△

Analogous definitions apply to indexing involving other repetitive operators than $\sum$, such as $\prod$ and $\bigcup$, and to expressions of the form $\{x \in \beta \mid \ldots x \ldots\}$. In other words, the latter expression denotes a bag, not a set. To represent the *set* of elements of β such that ..., we use expressions of the form $\{x \in \breve{\beta} \mid \ldots\}$.

The definitions of subset, superset, power set, union, intersection, set difference and Cartesian product for sets are easily generalized into natural definitions for bags. In each of the following definitions S denotes an arbitrary nonempty set.

Definition 2.7 (subbag, superbag, power set)
Let α and β be bags over S. Then

$$\alpha \subseteq \beta \quad \Leftrightarrow \quad (\forall x \in S)\ \alpha(x) \leq \beta(x)$$

In this case, α is called a subbag of β, and β a superbag of α. The power set $\mathcal{P}(\beta)$ of β is the set of all subbags of β.

△

As an example, the power set of $\{a, a, b\}$ is the set $\{\emptyset, \{a\}, \{b\}, \{a, b\}, \{a, a\}, \{a, a, b\}\}$

The union of two or more bags corresponds with the result of physically throwing the contents of each of the bags together into one bag.

Definition 2.8 (bag union ($\alpha \uplus \beta$))
Let $\beta_i, i = 1, \ldots, n$ be bags over S. The union $\biguplus_{1..n} \beta_i = \beta_1 \uplus \ldots \uplus \beta_n$ is the bag α, defined by

$$(\forall x \in S) \quad \alpha(x) = \sum_{i=1}^{n} \beta_i(x)$$

△

We use $\uplus$ rather than $\cup$ to indicate that the union of two bags, thus defined, is *not* the straightforward generalization of set union: The union of setlike bags (i.e., bags β_i such that $(\forall x \in \breve{\beta}_i)\ \beta_i(x) = 1$) is generally not a set. The actual generalization of set union to bags is what is usually (Banâtre & Le Métayer, 1993) referred to as the *maximum* $\bigcup_{1..n} \beta_i = \beta$ of the bags, defined by

$$(\forall x \in S) \ \beta(x) = \max_{i=1..n} \beta_i(x) \tag{2.2}$$

The intersection of a number of bags is the largest[1] bag with the property that it is a subbag of each.

[1]Here, "largest" means that the bag is a superbag of any other bag with the same property.

Definition 2.9 (bag intersection ($\alpha \cap \beta$))
Let β_i , $i = 1, \ldots, n$ be bags over S. The intersection $\bigcap_{1..n} \beta_i = \beta_1 \cap \ldots \cap \beta_n$ is the bag α, defined by

$$(\forall x \in S) \quad \alpha(x) = \min_{i=1..n} \beta_i(x)$$

△

The difference of α and β is the bag you get if you physically remove the intersection of the two from α.

Definition 2.10 (bag difference ($\alpha \backslash \beta$))
Let α and β be bags over S. Then the difference $\alpha \backslash \beta$ of α and β is the bag γ, defined by

$$(\forall x \in S) \quad \gamma(x) = \max\{0\,,\ \alpha(x) - \beta(x)\}$$

△

The definition of bag product is identical to that of set product, except that the multiplicity of $\langle x, y \rangle \in \beta_1 \times \beta_2$ is the product of the respective multiplicities of x and y in β_1 and β_2.

Definition 2.11 (bag product ($\alpha \times \beta$))
Let $\alpha \in \mathfrak{B}(S)$ and $\beta \in \mathfrak{B}(T)$. Then $\alpha \times \beta$ is the bag $\gamma \in \mathfrak{B}(S \times T)$, defined by

$$(\forall x \in S)\ (\forall y \in T) \quad \gamma(\langle x, y \rangle) = \alpha(x) \cdot \beta(y)$$

△

Example 2.12 (bag product)
If $\alpha = \{1, 1, 2\}$ and $\beta = \{1, 3, 3\}$ then $\alpha \times \beta = \{\langle 1, 1 \rangle : 2\,,\ \langle 2, 1 \rangle\,,\ \langle 1, 3 \rangle : 4\,,\ \langle 2, 3 \rangle : 2\}$
△

The definitions up to this point are based on (Banâtre & Le Métayer, 1993). The ones that follow are our own.

A construct that appears to be useful is the bag $\xi(\beta)$ of the multiplicities occurring in a bag β. We refer to $\xi(\beta)$ as the multiplicity type of β.

Definition 2.13 (multiplicity type)
Let $\beta \in \mathfrak{B}(S)$ be denoted by $\{x_1 : m_1, \ldots, x_n : m_n\}$. Then the multiplicity type of β is the bag

$$\xi(\beta) \triangleq \{m_1, \ldots, m_n\}$$

$\mathfrak{B}_\alpha(S)$ denotes the set of bags over S with multiplicity type α. △

Example 2.14 (multiplicity type)
Let $\alpha = \{a, a, a, b, b, c, d, d, f\}$, *and* $\beta = \{a, a, b, b, b, c, c, d, e\}$. *Then* $\xi(\alpha) = \xi(\beta) = \{3, 2, 2, 1, 1\}$, *and* $\xi(\xi(\alpha)) = \{2, 2, 1\}$.

△

If we rename the tool types occurring in α (in Ex. 2.14) according to the schema

$$a \leftarrow b;\ b \leftarrow a;\ c \leftarrow e;\ d \leftarrow c;\ f \leftarrow d$$

α transforms into β. In other words, α and β are alphabetic variants. It is clear from Def. 2.13 that two bags are alphabetic variants iff their multiplicity types are identical. Hence, the notion of multiplicity type can be used to get rid of representation symmetries.

2.2.3 Representation of allocations

In most scientific literature, an allocation is represented by a function $f : X \to Y$ that specifies, for each agent x in an agent space X, an endowment y in a commodity space Y. In mathematical economics, the agent space X is usually a finite set,[2] and the commodity space Y is typically a finite-dimensional continuum. In other words, the most frequent semantics of "allocation" is that of a function $f : \{1, \ldots, n\} \to \mathbb{R}_+^k$, where $\mathbb{R}_+$ denotes the set of nonnegative real numbers. Economic literature featuring allocations of the form $f : \{1, \ldots, n\} \to \mathbb{N}_0^k$ exists also, though it is relatively rare. However, in economics, a finite agent population X is *always* viewed as a *set*, never as a bag.[3]

For most of this thesis, this view toward allocations is quite acceptable. In our experimental evaluation of reallocation mechanisms, however, agents with the same tool utilities are essentially indistinguishable. In such a case, representing allocations as functions has the disadvantage that allocations represented by different functions may actually be identical. One can avoid this by defining allocations as mappings from a *bag* of agents to a bag of tools. To this avail, we introduce the concept of *multifunction*. A multifunction is identical to a function, except that its domain and value space are bags instead of sets.

Definition 2.15 below is a straightforward generalization of the definition of a function as a particular kind of relation. It expresses that a multifunction from α to β associates each element x in the carrier of α with a subbag of $\alpha(x)$ (not necessarily all different) elements from β.

[2] In some cases (e.g., (Allen & Hellwig, 1989) and (Artstein & Wets, 1989)), not only the commodity space, but also the agent space is uncountably infinite.

[3] Sometimes one considers uncountably infinite agent communities X, partitioned into agent *types*. Although these can be regarded as infinite-sized bags, there are not defined as such.

Definition 2.15 (multifunction)
Let α and β be bags. Then a multifunction ϕ from α to β is a subbag of $\alpha \times \beta$ with the property

$$(\forall x \in D_\alpha) \quad \sum_{y \in D_\beta} \phi(\langle x, y \rangle) = \alpha(x)$$

△

Just as it is customary with functions to write $f(x) = y$ instead of $\langle x, y \rangle \in f$, we will use a notation of the form $\phi(x) = \{y_1 : m_1, \ldots, y_n : m_n\}$ to express that

$$\begin{aligned} &\phi(\langle x, y_1 \rangle) = m_1 \;\wedge\; \ldots \;\wedge\; \phi(\langle x, y_n \rangle) = m_n \\ &\quad\wedge\;\; (\forall z \notin \{y_1, \ldots, y_n\})\; \phi(\langle x, z \rangle) = 0 \end{aligned}$$

Example 2.16 (alternative multifunction notations)
If $\phi = \{\langle 1, 2 \rangle, \langle 1, 3 \rangle : 2, \langle 2, 2 \rangle : 3\}$, then $\phi(1) = \{2, 3 : 2\}$, and $\phi(2) = \{2 : 3\}$. △

Multifunctions are more expressive than functions. One can describe the recipe "Put a grain of salt on two of the eggs and a grain of pepper on the third" in terms of a bag of eggs $\alpha = \{egg : 3\}$, a salt-and-pepper shaker $\beta = \{salt : 1000, pepper : 1000\}$, and a multifunction $\phi : \alpha \to \beta$ defined as $\phi(egg) = \{salt : 2, pepper\}$. It is not possible to express this recipe in terms of a function $f : \breve{\alpha} \to \breve{\beta}$. Since $\breve{\alpha} = \{egg\}$ and $\breve{\beta} = \{salt, pepper\}$, there exist only two possibilities for f, namely $f(egg) = salt$, and $f(egg) = pepper$. Neither of these expresses the recipe. To formalize the recipe in terms of functions, we must first transform the bag of eggs $\{egg : 3\}$ into the set $\{egg_1, egg_2, egg_3\}$. In other words, the first instruction to the cook would be to number the eggs....

In Definition 2.6, we specified how to interpret a bag-indexed operation like $\sum_{x \in \beta} f(x)$ if f is a normal function on the domain of β. If the summand of a summation contains a multifunctional expression of the form $\phi(x)$ instead of a functional expression like $f(x)$ in the above summation, we interpret ϕ as a nondeterministic function. Such an interpretation is similar to the interpretation of correspondences in economics (Hurwicz, 1986). Economists use correspondences to represent the possible outcomes of nondeterministic procedures. Formally, a correspondence f is a set-valued function. Yet, the semantics of $f(1) = \{1, 2, 5\}$ is that the outcome of the procedure f with input 1 is *one* of the numbers 1, 2, or 5 rather than the *set* containing these numbers. Similarly, one can interpret $\phi(1) = \{2, 2, 2, 4\}$ as "The outcome of ϕ with input 1 is either 2 (with a probability of 75%), or 4 (with a probability of 25%)". The associated interpretation of

$$\frac{1}{|\alpha|} \sum_{x \in \alpha} \phi(x)$$

is that of the *expected* outcome of ϕ with an input from α. This interpretation of multifunctional expressions underpins the following definition of bag-indexed summation of multifunctional expressions.

Definition 2.17 (multifunction indexing)
Let S and T be nonempty sets, $\langle T; +\rangle$ an abelian semi-group,[4] $\alpha \in \mathfrak{B}(S)$, $\beta \in \mathfrak{B}(T)$, *and let $\phi : \alpha \to \beta$ be a multifunction. Then*

$$\sum_{x\in\alpha} \phi(x) \triangleq \sum_{(x,y)\in\phi} y \tag{2.3}$$

△

Again, analogous definitions apply to indexing with operators like $\prod$ and $\biguplus$.

If $f : A \to B$ is a function with domain $D \subset A$, the *range* of f is defined as $R_f = \bigcup_{x\in D}\{f(x)\}$. The following definition of multifunction image is a generalization of this definition of function range. Note that Def. 2.18 involves multifunction indexing.

Definition 2.18 (multifunction image)
Let S,T,α, and β be as in Def. 2.17, and let $\phi : \alpha \to \beta$ be a multifunction. Then the image of ϕ is

$$Im\,\phi \triangleq \biguplus_{x\in\alpha} \{\phi(x)\}$$

△

Unlike functions, multifunctions $\phi : \alpha \to \beta$ generally do *not* have the property that $Im\,\phi \subseteq \beta$. As an example, the image of the multifunction $\phi : \{1,2,2\} \to \{1,2,2\}$, defined by $\phi = \{\{1,1\},\{2,1\},\{2,1\}\}$ is $Im\,\phi = \{1\} \uplus \{1\} \uplus \{1\} = \{1,1,1\}$, which is not a subbag of $\{1,2,2\}$. This is the reason that we choose to speak of the *image* of a multifunction, rather than the range. Note, however, that Def. 2.18 does coincide with the usual definition of function range, if the bag α happens to be a set.

To provide concise definitions of certain bag constructs, it is convenient to define a 'flattening' operator that removes the outermost structure of composite objects, thus turning a bag of bags of elements into a bag of elements.

Definition 2.19 (flattening a bag of bags)
Let γ be a bag of bags. Then $\underline{\downarrow}\gamma$ (pronounced as "flat gamma") is the bag, defined by

$$\underline{\downarrow}\gamma = \biguplus_{\beta\in\gamma} \beta$$

△

[4] This ensures that indexed summation is well-defined on T.

As an example, $\underline{\downarrow}\{\{a,a,b\},\{b,c\}\} = \{a,a,b,b,c\}$.

Definition 2.20 (bag allocation, partial allocation)
Let α and β be bags over S and T, respectively. An allocation of α over β (or, to the elements of β) is a multifunction $\delta : \beta \to \mathcal{P}(\alpha)$ with the property that

$$\underline{\downarrow}\, Im\,\delta = \alpha \tag{2.4}$$

The set of all allocations of α over β will be denoted as $\mathcal{A}(\alpha,\beta)$. We speak of a partial allocation of β over α, if not necessarily all of the elements of α are allocated, that is, if

$$\underline{\downarrow}\, Im\,\delta \subseteq \alpha \tag{2.5}$$

△

Equation (2.5) simply states that one cannot allocate more than one has. In mathematical economics (Takayama, 1985), this is referred to as the *feasibility* of an allocation. In our terminology, an allocation is feasible by definition. However, we will sometimes speak of an allocation δ of α over β in situations where only the carrier or the domain of α is specified. In such cases, we assume that the allocation is both feasible and total (i.e., not partial). In other words, α is then defined implicitly by $\alpha \triangleq \underline{\downarrow}\, Im\,\delta$.

If the bag β in Def. 2.20 happens to be a set (i.e., if $(\forall x \in \breve{\beta})\,\beta(x) = 1$), then our definition of allocation coincides with the usual, function-based definition of a feasible allocation. If β does contain elements with multiplicity greater than 1, the two differ, in the sense that some allocations which are distinguishable from the functional perspective are considered to be one and the same from the multifunctional perspective. As an example, the function-based allocations $f_1 = \{\langle egg_1, salt\rangle, \langle egg_2, pepper\rangle\}$, and $f_2 = \{\langle egg_1, pepper\rangle, \langle egg_2, salt\rangle\}$ are essentially identical if egg_1 and egg_2 are indistinguishable (i.e., instances of the same type).

2.2.4 Representation of reallocation proposals

A reallocation proposal can be represented by a pair of tool bags, one specifying the tools which the agent wishes to acquire (the demand bag), the other specifying the tools which it wishes to release (the supply bag). However, it is more economical to combine the demand bag and the supply bag into one construct, using positive multiplicities for the demand, and negative ones for the supply. We refer to such a construct as a *generalized* bag or gbag. The gbag $\{a:3, b:-2\}$ thus represents a proposal to release two b's in exchange for three a's.

Definition 2.21 (gbag, carrier, domain, size, finite gbags)
Let S be a nonempty set. A gbag γ over S is a function

$$\gamma : S \to \mathbb{Z}$$

S is referred to as the domain of γ. The subset $\check{\gamma} = \{x \in S \mid \gamma(x) \neq 0\}$ of S is called the carrier of γ. The size of γ is defined as $\sum_{s \in S} |\gamma(s)|$. A finite gbag is a gbag of finite size. The set of all finite gbags over S is denoted by $\mathcal{G}(S)$. △

We will use a more concise notation for small gbags. The gbag $\{a,\ b : -2\}$ is represented in this notation as $\{bb \rightleftharpoons a\}$, the gbag $\{a : -2, b : -1, c : 2, d\}$ as $\{aab \rightleftharpoons ccd\}$, and the gbag $\{a : 2\}$ as $\{ \rightleftharpoons aa\}$.

Defs. 2.8 and 2.19 of $\uplus$ and $\underline{\downarrow}$ for bags can be applied to gbags and bags of gbags without modification. As for the semantics in terms of reallocation proposals, the union of two proposals describes their aggregation, and the flattening of a bag of proposals describes the aggregation of the proposals in the bag. As an example, $\underline{\downarrow}\{\{aa \rightleftharpoons b\}, \{b \rightleftharpoons c\}\} = \{aa \rightleftharpoons b\} \uplus \{b \rightleftharpoons c\} = \{aa \rightleftharpoons c\}$.

We categorize reallocation proposals along two orthogonal dimensions. A proposal is either elementary or composite, and it is either a one-way or an exchange proposal. An exchange proposal expresses the desire to exchange tools against other tools. Hence, it is represented by a gbag with positive as well as negative multiplicities. A one-way proposal pertains to a transaction that involves either tool acquisition or tool relinquishment, but not both. Hence, one-way proposals are represented by a gbag that has only negative or only positive multiplicities. A proposal is an *elementary proposal* if neither its demand bag nor its supply bag contains more than one element. Hence, elementary exchange proposals are of the form $\{a \rightleftharpoons b\}$, whereas elementary one-way proposals are either of the form $\{a \rightleftharpoons \}$ or $\{ \rightleftharpoons a\}$. We denote the set of elementary proposals involving tool types from the set R by $\mathcal{G}_1(R)$. A composite proposal is a proposal that is not elementary.

In general, an agent in our Informed-Gambling framework submits a *bag* of proposals rather than a single proposal. To motivate this design decision, we make a small excursion from the current issue of representation to that of proposal semantics.

2.2.5 Semantics for reallocation proposals

Suppose that some agent communicates the proposal $\{xx \rightleftharpoons yy\}$. This gbag expresses a desire to acquire two y's in exchange for two x's. But what does that mean in a situation where this transaction is impossible, whereas it would be possible to exchange *one* x for *one* y? Does the agent prefer the latter exchange over no exchange at all?

If x stands for a bicycle and y for a motorbike, then this is probably so if the agent acts on behalf of a pizza delivery service. But it is probably not so if the motorbikes are meant to be used by the agent's sons during their vacation in France. It would be convenient if our representation covers both cases. One way to accomplish this is to let the pizza delivery agent submit a *bag* of proposals (viz., $\{\{x \rightleftharpoons y\}, \{x \rightleftharpoons y\}\}$ instead of $\{xx \rightleftharpoons yy\}$), with the semantics "Please satisfy as many of these proposals as possible".

If each agent is expected to submit a *bag* of proposals, both the pizza deliverer and the vacation planner can express their desires accurately: the pizza deliverer can submit $\{\{x \rightleftharpoons y\}, \{x \rightleftharpoons y\}\}$, and the vacation planner $\{\{xx \rightleftharpoons yy\}\}$.

Note that, while we commit ourselves to a representation of proposal submissions as bags of gbags, we do not commit ourselves to any associated semantics. The semantics "Please satisfy as many proposals as possible" proposed in the above scenario merely serves as an example to illustrate that a representation of submissions as bags of gbags is more versatile than a representation of submissions as gbags.

2.2.6 Representation of proposal-related constructs

To make sense of the definition of formal constructs below, the reader should keep in mind that the context in which these constructs play a role is that of a message exchange process between agents and a central coordinator. For now, one can think of this exchange as a synchronous process, where all agents submit their proposals simultaneously, and each agent receives a reply from the coordinator before any new proposals are submitted.

We define formal representations for the following proposal-related notions.

demand and supply bags: These represent the demand and supply associated with a single proposal (i.e., a gbag).

proposal profile: This represents the proposal bags of all agents (submitted in a single round of the message-exchange process).

scarcity profile: This represents the net demand and supply associated with a proposal profile.

market profile: This represents the gross demand and supply associated with a proposal profile.

The demand and supply bags associated with a single proposal γ are denoted by γ^+ and γ^- respectively. The formal definitions of these constructs are straightforward. Recall that $\mathfrak{B}(S)$ denotes the set of all finite bags over S, and $\mathcal{G}(S)$ the set of all finite gbags over S.

Definition 2.22 (proposal demand bag)
Let $\gamma \in \mathcal{G}(S)$. Then the demand bag of the gbag γ is the bag $\gamma^+ \in \mathfrak{B}(S)$ defined by

$$(\forall x \in S) \quad \gamma^+(x) = max(0, \gamma(x))$$

△

Definition 2.23 (proposal supply bag)
Let S be an ordered, nonempty set, and $\gamma \in \mathcal{G}(S)$. Then the supply bag of the gbag γ is the bag $\gamma^- \in \mathfrak{B}(S)$ defined by

$$(\forall x \in S) \quad \gamma^-(x) = max(0, -\gamma(x))$$

△

The demand and supply bags associated with a proposal *bag* (i.e., a bag of gbags) are defined as the union of the demand/supply bags of the proposals in the proposal bag. As such, they represent the gross demand and supply associated with the proposal bag. Since the definitions for the supply bag β^- and the demand bag β^+ of a proposal bag β are analogous, we provide a formal definition for the supply bag only.

Definition 2.24 (supply bag of a proposal bag)
Let S be a nonempty set, and $\beta \in \mathfrak{B}(\mathcal{G}(S))$ be a proposal bag. Then the supply bag of β is the bag $\beta^- \in \mathfrak{B}(S)$ defined by

$$\beta^- = \biguplus_{\gamma \in \beta} \gamma^-$$

△

As an example, the supply bag of the proposal bag $\beta \triangleq \{\{a \rightleftharpoons b\}, \{b \rightleftharpoons c\}\}$ is the bag $\beta^- = \{a, b\}$.

Definition 2.25 (proposal profile)
Let I denote a community of agents, and R a set of resource types. Then a proposal profile is a function $\psi : I \to \mathfrak{B}(\mathcal{G}(R))$. △

The scarcity of a tool type is its demand minus its supply. The scarcity profile associated with a proposal profile describes the scarcities of the tools mentioned in the proposal profile.

Definition 2.26 (scarcity profile)
Let $\psi : I \to \mathfrak{B}(\mathcal{G}(R))$ be a proposal profile. Then the scarcity profile associated with ψ is the gbag σ, defined by

$$\sigma = \underline{\sqcup\!\sqcup}\, Im\, \psi$$

△

The flattening operator is used twice in Def. 2.26 for the following reason. According to Def. 2.18 of *Im*, $Im\,\psi$ is the *bag* of proposal *bags* submitted by the agents. Hence, $\underline{\downarrow}\, Im\,\psi$ is the bag of all submitted proposals, and $\underline{\downarrow\downarrow}\, Im\,\psi$ is the gbag that describes the aggregate effect of these proposals. Consequently, $(\underline{\downarrow\downarrow}\, Im\,\psi)(x)$ denotes the scarcity (i.e., the aggregate demand minus the aggregate supply) of tool type x.

The market profile μ associated with a proposal profile ψ describes the gross supply and demand for tool types mentioned in the proposal profile. It is defined in terms of the bag $\underline{\downarrow}\, Im\,\psi$ of all submitted proposals.

Definition 2.27 (market profile)
Let $\psi : I \to \mathfrak{B}(\mathcal{G}(R))$ be a proposal profile. Then the market profile associated with ψ is the ordered pair of bags $\mu = \langle \mu^-, \mu^+ \rangle$, where

$$\begin{aligned} \mu^- &= (\underline{\downarrow}\, Im\,\psi)^- \\ \mu^+ &= (\underline{\downarrow}\, Im\,\psi)^+ \end{aligned}$$

△

If a reallocation proposal is accepted, it can be executed on the endowment of the agent that communicated it. Formally, we define the execution of a single proposal in terms of gbag union.

Definition 2.28 (proposal execution)
Let $\gamma \in \mathcal{G}(R)$ denote a reallocation proposal submitted by an agent with endowment β. Then the execution of γ on β is defined in terms of the function $exec : \mathcal{G}(R) \times \mathfrak{B}(R) \to \mathcal{G}(R)$ as

$$exec(\gamma, \beta) = \gamma \uplus \beta$$

△

Obviously, the execution of a proposal should turn the original endowment β into a new endowment $exec(\gamma, \beta)$. This implies that the outcome should be a bag, not a gbag. This is the case iff the agent is actually *capable* of delivering the promised tools, that is, if the supply bag γ^- of the proposal is a subbag of the agent's initial endowment β. We refer to a proposal with this property as a (locally) feasible proposal.

For the execution of a proposal *profile* (as a whole) to be feasible, the profile must meet two conditions. First, each agent must be capable of delivering the tools which it proposes to supply. We refer to this property as the local feasibility of the proposal profile. Second, we must have conservation of tools. In other words, gross supply must equal gross demand for all tool types. This is referred to as the global feasibility of the proposal profile. Formal definitions are presented below.

Def. 2.29 states that a proposal profile is locally feasible iff the supply bag of each agent's proposal bag is a subbag of its endowment.

Definition 2.29 (local feasibility)
Let I denote a community of agents, R a set of tool types, and let $\delta(i) \in \mathfrak{B}(R)$ denotes the initial tool endowment of agent i. Then the proposal profile $\psi : I \to \mathfrak{B}(\mathcal{G}(R))$ is locally feasible with respect to δ iff

$$(\forall i \in I) \quad (\psi(i))^{-} \subseteq \delta(i) \tag{2.6}$$

△

Equality of gross demand and gross supply is equivalent to zero scarcity. Hence, the global feasibility of a proposal profile can be expressed either in terms of the associated market profile, or in terms of the scarcity profile. We use the latter in the definition below.

Definition 2.30 (global feasibility)
Let $\psi : I \to \mathfrak{B}(\mathcal{G}(R))$ be a proposal profile, and $\sigma = \underline{\downarrow\downarrow}\, Im\, \psi$ the associated scarcity profile. Then ψ is globally feasible iff

$$\sigma = \emptyset \tag{2.7}$$

△

If a proposal profile ψ is globally feasible, and locally feasible with respect to an initial allocation δ, then $Exec(\psi, \delta)$ denotes the allocation that results from executing $\underline{\downarrow}\psi(i)$ on $\delta(i)$ for all agents i.

Definition 2.31 (proposal profile execution)
Let I denote a community of agents, R a set of tool types, $\delta_1 : I \to \mathfrak{B}(R)$ an initial allocation, and $\psi : I \to \mathfrak{B}(\mathcal{G}(R))$ a proposal profile that is globally feasible, and locally feasible with respect to δ_1. Then the execution of the profile ψ on the allocation δ_1 renders a new allocation $Exec(\psi, \delta_1) = \delta_2$, defined by

$$(\forall i \in I) \quad \delta_2(i) = exec(\underline{\downarrow}(\psi(i)), \delta_1(i))$$

△

2.2.7 Representation of agent satisfaction

When contemplating the representation of an agent's satisfaction with its endowment, an important distinction is that between *utilities* and *preferences*. We will speak of agent utility if an agent's satisfaction is expressed cardinally, by attaching a nonnegative number to any conceivable agent endowment, where a high number denotes a high degree of satisfaction.

Preferences are ordinal measures. As such, they are less expressive than utilities. An agent preference merely denotes, for any pair (x, y) of conceivable endowments, whether the agents prefers x over y, y over x, or is indifferent with respect to x and y. It does not express *to what extent* the agent prefers x over y.

If agent satisfaction is expressed cardinally, the quality of an allocation can be defined numerically, in terms of the agents' utilities, in various ways. Depending on the problem domain, one can equate allocation quality with the average agent utility (utilitarian), the average normalized agent utility (relative-utilitarian), the average quadratic deviation from the average agent utility (egalitarian), or yet another formula.

In the context of a preferential measure for agent satisfaction, it is unreasonable to expect a fine-grained measure for allocation quality. In view of the relative coarseness of preferences in comparison with cardinal agent satisfaction measures, any derivative of such an ordinal measure will be relatively coarse also. In fact, common preference-based measures for allocation quality, such as Pareto optimality and core membership are binary-valued. An allocation is either Pareto optimal or it is not. It is either a core element or it is not.

An advantage of ordinal measures over cardinal ones is that they are less demanding, and hence, more widely applicable. Contrary to a cardinal measure, a preference does not require that agent satisfaction is cardinally expressible, nor that the satisfaction of different agents is comparable. In fact, the notion of preference has been developed in response to the conviction (which grew in the 1930's among social choice theorists) that the — then common — cardinal measures were unrealistically demanding in these two respects (Sen, 1986).

For the portion of our research that pertains to CMAT, neither cardinal expressibility, nor inter-agent comparability constitute much of a problem. Open systems, however, are a different matter. Inter-agent comparability amounts to the existence of a universal standard measure for agent satisfaction. Such a fixed standard is at odds with the very idea of openness. Since this thesis aims to be relevant for OMAT as well as CMAT, we opt for a measure that does not require us to assume inter-agent comparability. Our measure does presume cardinal expressibility, because we believe that the *boolean* solution quality measures associated with an ordinal representation of agent satisfaction are too coarse for a proper evaluation of mechanism performance.

2.2.8 Relative-utilitarian agent-level utility

Below, we define the agent-level utility as a relative-utilitarian measure for agent satisfaction. The definition entails that the utility attached by an agent to a bag of tools is a real number between zero and one (inclusive), and that the utility of a bag of tools is never smaller than that of any subbag thereof. The philosophy behind this definition is "It can be profitable to acquire more tools, but enough is enough", while the common adage in game theory and micro-economics is "It is always profitable to acquire more".

Definition 2.32 (agent-level utility)
Let R be a set of resource types, and let $\Gamma \in \mathfrak{B}(R)$ denote the bag of tools present in an agent community I. Then an agent-level utility function is a function $u : \mathcal{P}(\Gamma) \to [0,1]$ with the following properties

1. $u(\emptyset) = 0$
2. $u(\Gamma) = 1$
3. $\gamma_1 \subset \gamma_2 \Rightarrow u(\gamma_1) \leq u(\gamma_2)$

△

While we consider the assumption "Enough is enough" more realistic for tool reallocation than "More is always better", Def. 2.32 does not *hinge* on the former assumption. There may well exist some superbag Ω of Γ to which the agent attributes a *value* $v(\Omega)$ that exceeds $v(\Gamma)$, but this is irrelevant for the agent's *satisfaction* u, if we presume that the agents are aware that they cannot acquire any tools except those in Γ. In fact, if we think of the agent's satisfaction as based on its valuation v of endowments, it appears that only property 3 in Def. 2.32 is vital, in the sense that it is the only property that requires us to make a constraining assumption on v. This is illustrated by Prop. 2.33 below, which also clarifies why the agent-level utility defined in Def. 2.32 was announced as a relative-utilitarian measure.

Proposition 2.33
Let $\Gamma \in \mathfrak{B}(R)$ denote the bag of tools present in some agent community, and let $v(x)$ denote the value attributed to the endowment $x \in \mathfrak{B}(R)$ by some agent in the community. Assume that v has the properties

$$(\forall \gamma_1, \gamma_2 \in \mathfrak{B}(R)) \qquad \gamma_1 \subset \gamma_2 \Rightarrow v(\gamma_1) \leq v(\gamma_2) \tag{2.8}$$

$$v(\Gamma) > v(\emptyset) \tag{2.9}$$

Define u in terms of v as

$$u(\beta) = \frac{v(\beta) - v(\emptyset)}{v(\Gamma) - v(\emptyset)} \tag{2.10}$$

Then the restriction of u to the domain $\mathcal{P}(\Gamma)$ has the properties 1 and 2 in Def. 2.32.

The proof is left to the reader. ■

Note that the condition $v(\Gamma) > v(\emptyset)$ in Prop. 2.33 is not much of a restriction: In view of (2.8), an agent for which this condition is not met cannot profit from *any* reallocation within the agent community.

As such, the only constraining assumption underlying our notion of agent-level utility is condition 3 in Def. 2.32: that an agent cannot gain utility by getting rid of tools. While this is not very restrictive, one can imagine cases in which retaining useless tools is costly, because the agent has to rent storage space for them. We can accommodate such cases by changing (2.10) into

$$u(\beta) = \frac{v(\beta) - v(\emptyset)}{\max_{\alpha \subseteq \Gamma} v(\alpha) - v(\emptyset)} \tag{2.11}$$

As a consequence, property 3 in Def. 2.32 disappears, and property 2 changes into

$$(\exists \beta \subseteq \Gamma) \;\; u(\beta) = 1$$

An important advantage of relative-utilitarian measures (such as the agent-level utility of Def. 2.32) is that they employ only such weak assumptions. As a consequence, a relative-utilitarian measure has a wider range of applicability than, for example, a utilitarian measure. Utilitarian measures are more demanding in that they require the satisfaction of different agents to be comparable. Agents are required to express their satisfaction in the same language, so to speak. Hence, by basing our performance evaluation on relative-utilitarian measures, we increase the generalizability of our experimental findings.

A relative-utilitarian measure for solution quality expresses a point of view that lies between utilitarianism and egalitarianism. From a utilitarian point of view, the wealth of a country does not change, if all citizens donate their entire income to the country's head of state. From a relative-utilitarian point of view, however, this constitutes a dramatic decrease of the country's wealth.

Even though an allocation that is optimal in a relative-utilitarian sense need not be optimal in a utilitarian sense, the conclusions we shall draw from our experiments do have *some* bearing on domains in which one employs a utilitarian measure, because relative-utilitarian optimality does imply utilitarian near-optimality, if there are no large differences between the utilities which different agents can attain if they acquire their respective first preferences.

2.3 Formal Problem Specification

2.3.1 Specification of allocation quality

The representations described in the previous section are almost sufficient to provide a formal specification of our prototype TR problem. The only ingredient that is still missing is a definition of allocation quality in terms of agent-level utilities.

Because TR is a generic problem, which occurs – in different guises – in a variety of contexts, there is no single optimality criterion that is always adequate. Indeed, when evaluating reallocation mechanisms in Chapter 7, we will employ several performance metrics to evaluate solution speed, and different aspects of solution quality. However, there are two solution quality measures that are likely to be the most important ones, certainly in most CMAT settings, and probably in many OMAT settings as well. Both measures can be used to provide a formal specification of TR problems, and the respective specifications are equivalent. The only difference between the two measures is that the allocation effectiveness is a *normalization* of the community-level utility: An allocation of maximal community-level utility has an effectiveness of 1. In our experimental evaluation, we use the effectiveness as our primary measure of mechanism performance, and the maximal value of the community-level utility as a measure for the difficulty of TR problem instances.

Def. 2.34 defines the community utility of a tool allocation as the average agent-level utility obtained by the agents in the community. The community tool bag is the bag of tools available for reallocation (i.e., the union of the initial tool endowments of the agents in the community).

Definition 2.34 (community utility)
Let R be a set of tool types, an let I denote a finite set of agents with agent-level utilities $u_i(\cdot)$ as in Def. 2.32, and tool endowments $\delta(i) \in \mathfrak{B}(R)$. Let $\Gamma \triangleq \underline{\downarrow} \mathit{Im}\, \delta$ denote the community tool bag, and $\mathcal{A}(\Gamma, I)$ the set of allocations of Γ over I.
Then the community utility is a function $C : \mathcal{A}(\Gamma, I) \to [0, 1]$, and the community utility of δ is defined as

$$C(\delta) \triangleq \frac{1}{|I|} \sum_{i \in I} u_i(\delta(i))$$

△

The community utility is a suitable measure to *compare* the quality of different allocations. However, it tells us little about the quality of an allocation in an absolute sense, because its maximal value depends on the difficulty of the problem at hand. For very difficult problems (where few of the agents' desires can be fulfilled) the maximum value of the community utility can be close to 0, whereas for very easy problems it can equal 1. As such, the community utility is not an adequate measure for the performance of a reallocation mechanism.

To obtain an adequate measure for mechanism performance, we divide the community utility of the allocation by the community utility of an optimal allocation. We refer to the resulting measure as the *effectiveness* of the allocation.

The allocation effectiveness is an *absolute* measure of mechanism performance in terms of allocation quality. An effectiveness of 1 implies that no mechanism can perform better, and an effectiveness of 0 means that no mechanism can do worse.

Definition 2.35 (allocation effectiveness, opt. community utility)
Let Γ, I, $\mathcal{A}(\Gamma, I)$, and δ be as in Definition 2.34.
Then the effectiveness is a function $E : \mathcal{A}(\Gamma, I) \to [0,1]$ that is defined as follows. Let C^* denote the optimal community utility, that is, the highest community utility that can be obtained with an allocation from $\mathcal{A}(\Gamma, I)$. Formally,

$$C^* = \max_{\phi \in \mathcal{A}(\Gamma, I)} C(\phi)$$

Then the effectiveness of the allocation δ equals

$$E(\delta) = \frac{C(\delta)}{C^*}$$

△

Note that, while the (optimal) community utility and the effectiveness are relative-utilitarian concepts throughout this thesis, they are not *inherently* relative-utilitarian. It depends on the underlying notion of agent-level utility whether the above definition of effectiveness constitutes a utilitarian or a relative-utilitarian measure of allocation quality.

2.3.2 Formal specification of TR

Using the effectiveness metric as the measure of choice for the quality of reallocation mechanism, we arrive at the following formal specification of TR as an inherently distributed problem.

Definition 2.36 (tool reallocation (TR))
Let I denote a set of agents, and R a set of tool types. A tool reallocation problem is a pair $\langle \delta, \widetilde{U} \rangle$ where

- *$\delta : I \to \mathfrak{B}(R)$ is an initial allocation;*
- *$\widetilde{U} : I \times \mathcal{P}(\Gamma) \to [0,1]$ is a profile of agent-level utilities. Here, $\Gamma \triangleq \underline{\downarrow} \mathit{Im}\, \delta$ denotes the community tool bag.*

A proposal profile $\psi : I \to \mathfrak{B}(\mathcal{G}(R))$ constitutes an optimal solution of $\langle \delta, \widetilde{U} \rangle$ iff

1. *ψ is locally feasible with respect to δ, and*

2. *ψ is globally feasible, and*

3. *execution of ψ on δ renders an allocation of maximal effectiveness, that is,* $E(\ Exec(\psi, \delta)\ ) = 1$.

We will denote the space of all TR problems by TR. △

Def. 2.36 defines tool reallocation as the problem of arriving at a fully executable (i.e., locally and globally feasible) profile of reallocation proposals that, when executed, renders an optimal allocation.

If TR problems are to be solved in a distributed fashion, by communicating reallocation proposals, while the agents' *utilities* remain private information, no agent (or coordination module) can acquire full knowledge of the problem instance. Hence, demanding optimality may be too high-strung. In view of this, we will use Def. 2.36 only as a *guideline* in our search for an adequate distributed solution method. In other words, a multi-agent mechanism need only render allocations of *near*-maximal effectiveness to qualify as an adequate distributed solution method,

Obviously, the profile of the agents' first submitted proposals reflects their first preferences. Hence, if these proposals happen to be jointly satisfiable, executing them will render an optimal allocation. Of course, this is seldom the case. While it is reasonable to assume that any submitted profile is locally feasible, an *initial* profile is unlikely to be *globally* feasible. Usually, there are discrepancies between the desires of different agents. As a consequence, some tool types will be scarce, while there may be an abundant supply of others. Hence, solving a distributed TR problem amounts to finding a way to turn the initial profile into one that is globally feasible, without sacrificing too much allocation quality. In the next chapter, we explore various ways to accomplish this in a distributed fashion.

Chapter 3

Distributed Approaches to TR

3.1 Coordination Strategies

3.1.1 Central vs. distributed coordination

For a globally coherent solution to emerge from agent interactions, these interactions must be coordinated. In many natural multi-agent systems (e.g., ant colonies), such coordination is entirely distributed. Ants who contribute to the coordinated construction of an ant hill interact directly with other such ants, and there are no coordination centers in the form of foremen.

While a fully distributed approach to TR is principally possible, we do not explore this possibility. For TR, such an approach tends to be less efficient than approaches which feature indirect communication via a central coordination module, because adequate reallocation requires global knowledge of the problem. There are two main reasons for this. First, reallocation problems typically call for *multilateral* cooperation. Two agents with exchange proposals $\{a \rightleftharpoons b\}$ and $\{b \rightleftharpoons c\}$ can cooperate (i.e., exchange tools) only with help of a third party (e.g., an agent with a proposal $\{c \rightleftharpoons a\}$). Second, reallocation problems typically feature *conflicting* opportunities for cooperation. As such, a decision to seize an opportunity to cooperate can obstruct a more profitable cooperation.

An example is provided in Fig. 3.1. The figure depicts an initial proposal profile of a TR problem comprising 5 agents. Their respective proposals are listed inside the circles. For the sake of simplicity, we assume that adaptation of the proposals is not an option. If a submitted proposal is not accepted, the submitter does not take part in the reallocation at all. With this assumption, the quality of a solution is proportional to the number of tool exchanges that take place. In other words, cooperation should involve as many agents as possible.

In the current example, there are are only two groups of agents that can cooperate

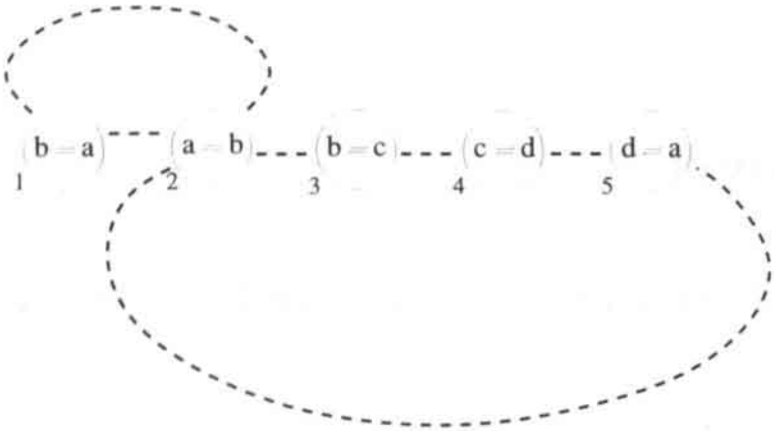


Figure 3.1: A drawback of direct agent interaction: The likely cooperation between agents 1 and 2 obstructs the cooperation between agents 2, 3, 4, and 5.

(i.e., exchange tools): the group comprising agents 2, 3, 4, and 5, and the pair of agents 1 and 2. However, once agents 1 and 2 decide to exchange their tools, the cooperation within the group of four agents is no longer possible.

To decide which opportunities for cooperation should be grasped at, a *global* perspective on the problem is required. To acquire such a global perspective, some of the information that is distributed over the agents must be aggregated. Having each agent perform this task on its own would lead to a lot of duplicate work. It is generally more efficient, with respect to computation as well as communication, to let a dedicated module take care of the aggregation of local information. This is especially so if the module can communicate with all agents directly.

Once a coordination module has aggregated all local information into a global view of the situation, it is principally capable of solving the problem as least as proficiently as the community of agents could do after information aggregation. However, a coordination module which, after collecting the necessary information, solves the entire problem on its own, is a computation bottleneck by definition. We aim to prevent this, and will strive for an approach where the coordination module does not perform any complex computations.

3.1.2 Proposal composition and relaxation

With our formal definition of tool reallocation (Def. 2.36 on page 42), solving a TR problem amounts to transforming an initial profile of reallocation proposals into one that is globally feasible, without sacrificing too much allocation quality. As we explained in Sect. 1.3.2, we impose two constraints on a solution procedure for TR.

1. Agents communicate only reallocation proposals. All other information on their utilities for tools is to remain private.

2. The procedure should not involve the use of real or artificial money.

In the current chapter, we investigate two strategies to solve the TR problem described in Def. 2.36 in this manner: proposal composition and proposal relaxation.

Proposal composition entails the construction of a globally feasible *subprofile* of proposals within the original profile of agent submissions. As such, it can be qualified as a bottom-up approach. The term 'composition' stems from (Sathi & Fox, 1989), who mention constraint composition as one of the techniques employed in their Constraint-Directed Negotiation (CDN) approach. Although our notion of composition is basically the same as that of Sathi and Fox, we will embed composition in a context where it plays a more prominent role than in CDN, and provide formal definitions for several variants of composition.

Proposal relaxation entails persuading or motivating the agents to adapt their original proposals so as to turn the initial proposal profile into one that is globally feasible. As such, relaxation constitutes a top-down approach.

In CDN, relaxation is a matter of persuasion. The coordination module in CDN acts as a *mediator* who queries the agents to attain a global perspective on the problem, and attempts to persuade the agents to relax their proposals in a specific direction. We will refer to this method of relaxation as *mediated negotiation* in the sequel. Because mediated negotiation incurs a concentration of computation and design complexity in the coordination module, this mediator is liable of becoming a (design and computation) bottleneck. We prefer a method of proposal relaxation with a higher degree of operational decentralization. Consequently, we focus on approaches where the agents take the initiative, and the coordination module merely provides *incentives* for relaxation. More particularly, we investigate the potential of the Walrasian auction, in which an auctioneer provides such incentives by changing the prices (i.e., exchange ratios) of the tool types.

3.2 Composition-Based Solution of TR

Before we specify the role we envision for composition as a strategy to solve TR problems, we sketch how it was employed in constraint-directed negotiation. This sketch is phrased in terms of our own terminology and notation, rather than those of (Sathi & Fox, 1989).

3.2.1 Composition in CDN

Sathi and Fox (1989) developed their constraint-directed negotiation (CDN) approach to tackle the (real-life) reallocation of computer workstations in a software-engineering company. As we explained in Sect. 2.1.1, a workstation reallocation problem can be regarded as an instance of TR, where an agent represents a project team, and a tool is a workstation.

The CDN approach was conceived by studying how human problem solvers deal with the problem. This led to the identification of three problem solving activities.

Composition concerns the construction of proposal clusters by matching a proposal which involves a request for a tool of a certain type with another proposal in which that tool type is offered. The aim is to form a cluster of proposals such that any tool occurring in a proposal of the cluster is matched with a tool in another proposal in the cluster. The entire cluster of proposals can then be executed, since it amounts to a globally feasible bag of proposals.

Relaxation concerns a negotiation process in which the mediator suggests specific adaptations of submitted proposals to the agents, which it deems to be useful in the sense that these adaptations would allow for more extensive composition to take place.

Reconfiguration concerns the exchange of workstation components (such as monitors and other peripherals) to arrive at a community bag of workstation types that comes closer to the desired community bag (i.e., the bag $(\Gamma \uplus \mu^+) \backslash \mu^-$, where Γ denotes the current community tool bag, and μ^+ and μ^- denote the demand and supply components of the current market profile.)

In each of the three activities, the initiative lies with the mediator. The activities are intertwined. Composition can be used to perform a partial reallocation, but it can also serve to identify useful relaxation and reconfiguration attempts. The overall control scheme that determines how the mediator combines these activities is not specified in detail by (Sathi & Fox, 1989). This suggests that it is complex.

3.2.2 Composition combined with revision

Because we are primarily interested in the TR domain as a test domain for fundamental research on multi-agent systems, we prefer a *simple* control structure that is *principally* suited for any TR problem over a *complex* control structure that is *particularly* suited to solve a specific real-life reallocation problem. To arrive at such a simple control structure, we refrain from incorporating reconfiguration and relaxation. Instead, we combine composition with *proposal revision*.

Proposal revision aims to ensure that any submitted proposal has at least some chance of being accepted. To guarantee this, any tool type requested in a proposal profile should be offered in at least one other proposal of the profile. Revision is easy to implement as a message exchange process between the agents and the mediator. The idea is for the mediator to communicate the set of tool types offered in the last submitted proposal profile. The agents then respond by constraining their demand to the tools types mentioned as deliverable by the mediator. Because it is conceivable that an agent which has submitted $\{x \rightleftharpoons y\}$ is no longer willing to relinquish tool type x if it appears impossible to obtain y, the set of offered tool types can shrink in the process of revision. Hence, the revision process will generally take more than one round. It is, however, a trivial process, provided that we do not allow the agents to offer a tool type in their proposals that they did not offer in previous rounds.

The combination of composition with revision constitutes a *greedy* approach to solve TR problems. After revision of the initial proposal profile, the mediator searches for a globally feasible subbag of the bag of submitted proposals, and executes this subbag. Subsequently, the process of revision and composition is repeated with those agents who submitted proposals that were rejected. This continues until all proposals of the still active agents are accepted, or revision points out that all remaining agents prefer their current allocation over anything their still active companions can offer them.

The composition/revision scheme, thus described, is not guaranteed to work. The process may go on endlessly, because one can conceive proposal bags that are stable under revision, but do not contain any nonempty, globally feasible subbags.[1] To make the above scheme work, we would have to add either some form of proposal relaxation, or the simple, but crude method of excluding some agent(s) from further participation, whenever composition stagnates. We do not bother to elaborate on such a fail-safe provision, because it is not required in the composition/revision scheme which we will ultimately arrive at.

Evidently, in whatever way composition and revision are combined, revision is not liable of turning the mediator into a computation or design bottleneck. Consequently, we focus on composition.

3.2.3 Complexity of general proposal composition

The composition of reallocation proposals can be viewed as a heuristic reformulation of Def. 2.36 of tool reallocation, which takes into account that the mediator does not have access to the agents' utility information. It must go by the communicated proposal profile. Because a proposal profile does not provide any clues as to how important a specific proposal is to its submitter, the mediator assumes that all

[1] A simple example is the proposal bag $\{\{aa \rightleftharpoons b\}, \{b \rightleftharpoons a\}\}$.

proposals are equally important. We refer to this conjecture as the *uniform-utility* assumption, and to the associated interpretation of composition as the uniform-utility composition problem (UUCP).

Although the mediator needs to know the coupling of proposals to their respective submitters to *execute* the accepted proposals, the *selection* of these proposals (under the uniform-utility assumption) does not require such knowledge. Consequently, we can define UUCP in terms of the bag of all submitted proposals, instead of the submitted proposal profile.

Even with the uniform-utility assumption, one can define UUCP in various, slightly different ways. To clarify the differences between these variants, we provide a pictorial representation of a UUCP instance in the form of a three-dimensional puzzle.

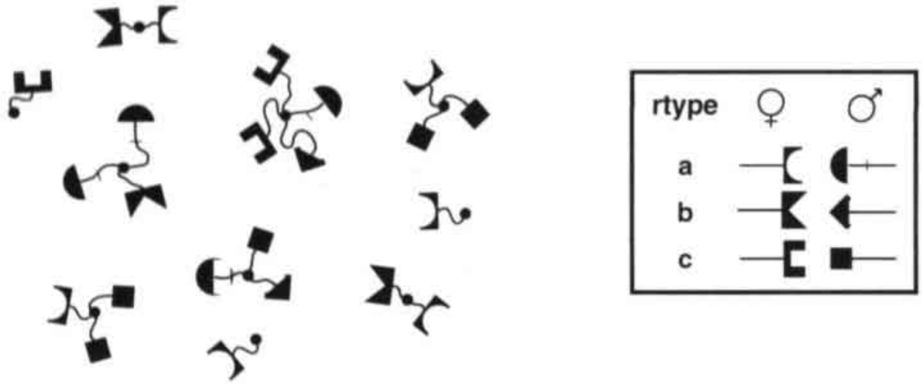


Figure 3.2: Representation of the UUCP instance $\{\{ab\rightleftharpoons cc\},\{aa\rightleftharpoons b\},\{cc\rightleftharpoons a\}{:}2,\{abc\rightleftharpoons\},\{\rightleftharpoons ab\}{:}2,\{\rightleftharpoons a\}{:}2,\{\rightleftharpoons c\}\}$ as a 3-dimensional puzzle.

This is done in Fig. 3.2. Each piece of the puzzle consists of a number of flexible strings connected to a central core, with connector terminals of various shapes at the ends of the strings. A tool type corresponds with a *pair* of connector types: A male connector is used if a tool is offered in a proposal, and a female connector if the tool is requested. Solving the composition problem thus corresponds to knitting pieces together, in such a manner that there are no open ends (i.e., uncoupled connectors).

Below, we define four variants of UUCP in increasing order of demandingness.

random composition (UUCP$^{\bullet}$)
: This is the simplest form of composition. It entails the selection and execution of an *arbitrary* globally feasible subbag of the proposal bag.

random cluster composition (UUCP$^{\odot}$)
: This is the operational definition *you* would probably employ when asked to solve the composition problem instance of Fig. 3.2. It involves the construction of a cluster (i.e., a single, connected knit of puzzle pieces without open ends) instead of a globally feasible subbag (which can contain multiple clusters).

LBF composition (UUCP*)
: Largest-Bag-First (LBF) composition entails the selection and execution of a *maximal*, globally feasible subbag of proposals.

LCF composition (UUCP$^{\circledast}$)
: Largest-Cluster-First (LCF) composition entails the selection and execution of a maximal *cluster* of jointly executable proposals.

We recall from Sect. 3.2.1 that composition in CDN was inspired by the problem-solving strategies used by human reallocators. Human problem solvers are inclined to use a bottom-up approach, and tend to come up with a solution comprising a single *cluster* of proposals.[2] In terms of our bag-based representation framework, a cluster is a globally feasible proposal bag which cannot be split into *smaller* globally feasible subbags. Hence, any solution found with random cluster composition is also a solution for random composition. Obviously, this also applies to LCF- and LBF-composition. So variants of composition which involve cluster construction are at least as demanding as the corresponding variants with proposal bags.

Sathi and Fox (1989) mention that human problem solvers exhibit a preference for *large* clusters. We refer to this preference as the Largest-Cluster-First (LCF) heuristic. It is not discussed in (Sathi & Fox, 1989) whether the use of this heuristic is based on the experience that it tends to lead to better solutions.[3] Pondering on the adequacy of the heuristic in terms of the effectiveness of the resulting allocations, one can imagine that satisfying as many first preferences as possible is a good heuristic in cases where the agent-level utility of second-best alternatives tends to be low (say, 10% or less). On the other hand, one can also imagine problems where an agent whose proposals were rejected can still obtain a considerable utility gain from a renewed attempt involving 10 other agents, but much less so if there are only 2 other agents left.

[2]Sathi and Fox (1989) speak of a cycle, or cascade, where we use the term cluster.

[3]Its popularity can also be explained psychologically, as stemming from the urge to get the job over with as soon as possible.

In any case, there is little ground to expect that the LCF heuristic will produce significantly better allocations than the Largest-Bag-First (LBF) heuristic, since both heuristics seem to be based on the belief that it is profitable to satisfy many of the agents first preferences, and *if* this belief is correct, LBF will do at least as well as LCF.

We provide formal definitions for UUCP$^\bullet$ and UUCP*. UUCP$^\odot$ and UUCP$^\oplus$ differ from UUCP$^\bullet$ and UUCP* only in the additional condition that the target subbag (α and α^* in the formal definitions below) should be a cluster. Formally, the condition that the bag α be a cluster can be expressed as

$$(\forall \gamma \subsetneq \alpha)\ \ \gamma \neq \emptyset \Rightarrow \underline{\downarrow}\gamma \neq \emptyset \tag{3.1}$$

Because random composition does not involve any optimization, it is formalized as a *decision* problem. Note that random cluster composition is associated with the same decision problem.

Definition 3.1 (random composition (UUCP$^\bullet$))
Let R denote a set of tool types. Then UUCP$^\bullet$ *comprises the following problem.*

Given a bag $\beta \in \mathfrak{B}(\mathcal{G}(R))$ of reallocation proposals, determine whether

$$(\exists \alpha \subseteq \beta)\ \ \alpha \neq \emptyset \wedge \underline{\downarrow}\alpha = \emptyset \tag{3.2}$$

△

Definition 3.2 (LBF composition (UUCP*))
Let R denote a set of tool types. Then UUCP* *comprises the following problem.*

Given a bag $\beta \in \mathfrak{B}(\mathcal{G}(R))$ of reallocation proposals, determine a subbag $\alpha^* \subseteq \beta$ with the property

$$\underline{\downarrow}\alpha^* = \emptyset\ \wedge\ (\forall \alpha \subseteq \beta)\ \ \underline{\downarrow}\alpha = \emptyset \Rightarrow |\alpha| \leq |\alpha^*| \tag{3.3}$$

△

Looking back at Fig. 3.2, neither variant of composition seems to be easy. The following proposition tells us that this is indeed so.

Proposition 3.3
UUCP$^\bullet$ *is NP-complete.*

We prove the NP-completeness of UUCP$^\bullet$ by restriction to the *subset-sum problem*, a well-known, NP-complete decision problem (Moret, 1998, p.251). More specifically, we show that subset-sum is isomorphic (under one-reduction) to the special case UUCP$_1$ of UUCP$^\bullet$ where there is only one tool type, one of the proposals is a one-way proposal with an empty demand bag, and all other proposals have empty supply bags.

Definition 3.4 (UUCP$_1$)
UUCP$_1$ *is the special case of* UUCP$^\bullet$ *with* $\beta \in \mathfrak{B}(\mathcal{G}(R))$ *such that*

(i) $|R| = 1$

(ii) $(\exists! \gamma \in \beta)\ (\exists x \in R)\ \gamma(x) < 0$

△

The following definition describes the subset-sum problem in terms of bags.

Definition 3.5 (subset-sum problem)
Given $\beta \in \mathfrak{B}(\mathbb{N})$, *and* $M \in \mathbb{N}$, *determine whether*

$$(\exists \alpha \in \mathfrak{B}(\mathbb{N}))\ \ \alpha \subseteq \beta \wedge \sum_{x \in \alpha} x = M \tag{3.4}$$

△

Proof of Prop. 3.3.
We show that the subset-sum problem is isomorphic with the special case UUCP$_1$ of UUCP$^\bullet$, where $\beta \in \mathfrak{B}(\mathcal{G}(R))$ with $|R| = 1$. We use $R = \{a\}$ in the proof, but the identity of R does not matter as long as it comprises only one tool type.

Let $S_{\beta,M}$ be an arbitrary instance of the subset-sum problem. We will transform this into an instance C_γ of UUCP$_1$, and show that

1. The answer to $S_{\beta,M}$ is always the same as the answer to C_γ.
2. The transformation involves a number of computation steps that is polynomial in $|\beta| + 1$, the size of $S_{\beta,M}$.
3. The transformation is one-to-one, and surjective.

Let f denote our transformation. We define $f(S_{\beta,M})$ to be the UUCP$_1$ instance C_γ defined by the following proposal bag $\gamma \in \mathfrak{B}(\mathcal{G}(\{a\}))$:

$$\gamma \triangleq \Big(\biguplus_{x \in \beta} \{\{a : x\}\}\Big) \uplus \{\{a : -M\}\} \tag{3.5}$$

Here $\{\{a : k\}\}$ represents a bag with one element, a gbag demanding $k \in \mathbb{Z}$ occurrences of a.

The answer to C_γ is also the answer to $S_{\beta,M}$. We prove this by showing that a positive answer to C_γ implies a positive answer to $S_{\beta,M}$, and vice versa.

A positive answer to C_γ implies that there exists a subbag $\sigma \subseteq \gamma$, such that

$$\sigma \neq \emptyset \wedge \underline{\downarrow}\sigma = \emptyset \tag{3.6}$$

Since $\{a : -M\}$ is the only element of γ that supplies a, while all other elements of γ demand it, it follows from $\underline{\downarrow}\sigma = \emptyset$ and $\sigma \subseteq \gamma$ that $\{a : -M\} \in \sigma$, and that, hence, $|\underline{\downarrow}(\sigma \setminus \{\{a : -M\}\}| = M$. Together with (3.5), this implies that the bag $\alpha \in \mathfrak{B}(\mathbb{N})$ defined by

$$\alpha \triangleq \biguplus_{\theta \in \sigma \setminus \{a:-M\}} \{\theta(a)\} \tag{3.7}$$

is a subbag of β that satisfies the condition $\sum_{x \in \alpha} x = M$ in (3.4), in Def. 3.5 of $S_{\beta,M}$. So the answer to $S_{\beta,M}$ is also "yes".

The converse is also true. If the answer to $S_{\beta,M}$ is "yes" then (3.4) is satisfied by some α. Using this α, define

$$\sigma \triangleq (\biguplus_{x \in \alpha} \{\{a : x\}\}) \uplus \{\{a : -M\}\} \tag{3.8}$$

Then σ is a nonempty subbag of γ such that $\underline{\downarrow}\sigma = \emptyset$. So the answer to C_γ is also "yes".

The transformation of $S_{\beta,M}$ into C_γ is polynomial in $|S_{\beta,M}| = |\beta| + 1$:
If a (g)bag α is represented in computer memory as a list of pairs of the form $[\ldots, \langle x, \alpha(x) \rangle, \ldots]$, where x ranges over $\check{\alpha}$, the bag of gbags γ (which defines C_γ) is represented as a list of the form $[\ldots, \langle [\langle a, k \rangle], l \rangle \ldots]$, where the element shown expresses that γ contains l proposals offering k type-a tools. Using such a representation, the construction of γ according to (3.5) takes $O(|\beta| + 1)^2$ steps.

Finally, it is obvious that the transformation f defined by (3.5) is a one-to-one mapping from the set of all subset-sum instances to UUCP_1. If we replace σ in (3.7) by the bag γ defined in (3.5), the obtained α equals the bag β of $S_{\beta,M}$. In other words, by replacing σ with γ, we turn (3.7) into a specification of the inverse transformation f^{-1} of f. This inverse mapping is well-defined for any instance of UUCP_1. Hence f is also surjective. ■

Corollary 3.6 *Random cluster composition* ($\text{UUCP}^{\odot}$) *is NP-complete, while LCF- and LBF-composition* ($\text{UUCP}^{\oplus}$ *and* UUCP^{*}) *are NP-hard.*

Apparently, neither of the four types of composition we have considered is tractable. Hence, as an algorithmic component of a composition/revision strategy for TR, proposal composition falls short with respect to our strife to prevent the mediator from becoming a computation or design bottleneck. Since we think that the mediator's task should *at least* be tractable, and cannot conceive a variant of proposal composition that is less demanding than random composition, we will seek salvation in a more rigorous simplification.

3.2.4 Elementary composition

Elementary composition differs from general composition in that it only involves *elementary* reallocation proposals, that is, proposals of the forms $\{x \Rightarrow y\}$, $\{x \Rightarrow\}$ and $\{\Rightarrow y\}$. The four variants of UUCP give rise to four kinds of elementary composition. For reasons that will become clear later, we focus on the LBF variant ECP$^\bullet$, and the comparison of this variant with the LCF variant ECP$^\circledast$. As with UUCP, we will use the acronym ECP in statements that apply to either variant.

Initially, our analysis is confined to the composition of elementary *exchange* proposals (i.e., proposals of the form $\{x \Rightarrow y\}$). One-way proposals ($\{x \Rightarrow\}$ and $\{\Rightarrow x\}$) will be incorporated later.

The formal definitions of the four variants of elementary composition differ from the associated variants of UUCP only in that the proposals are elementary. Consequently, we only provide a formal definition for ECP$^\bullet$.

Definition 3.7 (elementary LBF composition (ECP$^\bullet$))
Elementary LBF composition problem comprises the following problem.
Given a set of tool types R, and a bag $\beta \in \mathfrak{B}(\mathcal{G}_1(R))$ of elementary exchange proposals, determine a maximal subbag $\alpha^\bullet \subseteq \beta$ of jointly executable proposals, that is, a subbag $\alpha^\bullet$, satisfying

$$\underline{\downarrow}\alpha^\bullet = \emptyset \ \wedge \ (\forall \alpha \subseteq \beta) \ \ \underline{\downarrow}\alpha = \emptyset \Rightarrow |\alpha| \le |\alpha^\bullet| \tag{3.9}$$

△

3.2.5 Graphic representation of ECP instances

To gain insight in the characteristics of a problem domain, a graphic representation of problems is often helpful. The natural graph representation of an ECP instance associates proposals with nodes. We used such a *proposal-centered* representation in Fig. 3.1, and a similar representation was used by Sathi and Fox (1989).

The *proposal graph* of a composition problem is a directed graph, in which each node corresponds with a single proposal. An example is provided in Fig. 3.3. As apparent from the figure, there is an arc from node i to node j iff the tool type offered in proposal i equals the tool type requested in proposal j. The arc is labeled with this tool type.

The structural characteristics of a *proposal graph* can be interpreted in terms of the associated composition problem as follows:

1. The indegree (i.e., the number of incoming arcs) of node i equals the number of proposals that match with the right member (the tool type requested) of proposal i, that is, the number of candidate suppliers to i.

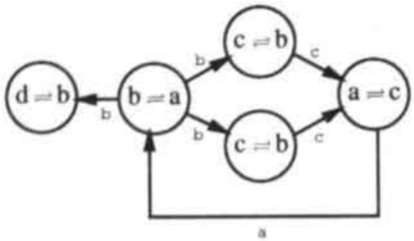


Figure 3.3: Proposal graph of $\{\{d \rightleftharpoons b\}, \{b \rightleftharpoons a\}, \{c \rightleftharpoons b\}{:}2, \{a \rightleftharpoons c\}\}$.

2. The outdegree of node i (i.e., the number of outgoing arcs) equals the number of proposals that match with the left member (tool type offered) of proposal i, that is, the number of candidate customers for i.

3. Any *simple* cycle[4] in the graph represents a solution to the ECP$^{\oplus}$ instance described by the graph.

These cycles are simple, because an elementary exchange proposal involves the deliberation to exchange *one* tool for *one* other.

Hence, a *solution* to ECP* can be visualized in the proposal graph G as a subgraph G' of G, which is maximal in terms of the nodes of G it comprises. Since each proposal in ECP concerns only one exchange (viz. one tool to be received and one to deliver), the indegree and outdegree of all nodes in the solution graph G' must be 1. In other words, the simple cycles which constitute the solution must be disjoint. Hence, ECP* can be formulated in terms of the associated proposal graph as:

> "Determine a set of disjoint simple cycles that comprises a maximum number of nodes."

From the above structural properties, it is apparent that all of the incoming arcs of a node representing the proposal $\{x \rightleftharpoons y\}$ have the label y. In general, all of the incoming arcs of a node have the same label, and so do all arcs emanating from a node. As a consequence of these label consistency constraints, certain graph structures cannot occur within a proposal graph. An example is the structure pictured in Fig. 3.4(a). A proposal graph that contains a subgraph like the one in Fig. 3.4(a) does not represent any sensible composition problem. This is a consequence of the requirements that each of the incoming arcs of a node must have the same label, as well as all arcs emanating from a node. Subsequently applying these rules to the arcs pointing from node 1, and those to node 2 in Fig. 3.4(a), reveals that the only consistent labelings are those of the form shown in Fig. 3.4(b), where all arcs have the same label. This implies that node 3 represents a proposal of the form $x \rightleftharpoons x$,

[4] A cycle is simple iff it traverses no node more than once

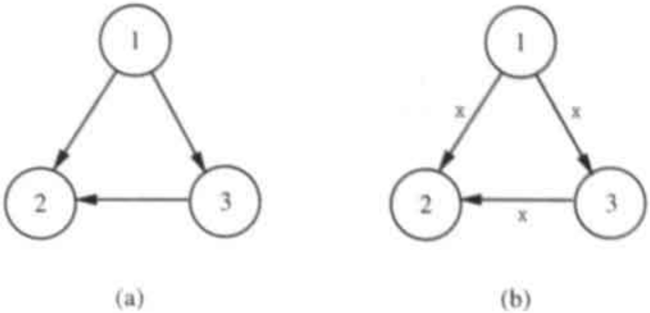


Figure 3.4: A directed graph of the form (a) cannot occur within a proposal graph, because it cannot be labeled sensibly (b).

which is obviously senseless, since the utility of a tool is completely determined by its type.

We have shown that not every graph can occur as the proposal graph of a sensible ECP instance. There is an alternative graphic representation which does not suffer from this disadvantage, and is also more concise than the proposal-graph representation. As such, this representation is better suited as an instrument for the analysis of ECP.

3.2.6 Tool graphs

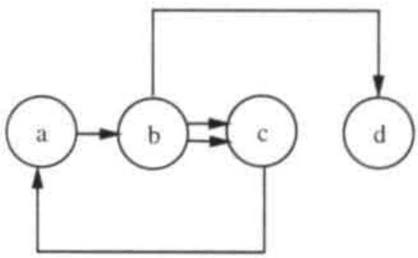


Figure 3.5: The tool graph of $\{\{d \rightleftharpoons b\}, \{b \rightleftharpoons a\}, \{c \rightleftharpoons b\}:2, \{a \rightleftharpoons c\}\}$.

In the *tool graph* of a composition problem, each node corresponds with a tool type, and an arc from node i to node j represents an exchange proposal $\{j \rightleftharpoons i\}$. Obviously, the tool graph of a composition problem is the *dual* of its proposal graph. The structural characteristics of a tool graph can be interpreted as follows.

1. The indegree of node i equals the number of proposals that specify a tool of type i as one they are willing to relinquish, in other words, the *supply* of tool

type i.

2. The outdegree of node i equals the number of proposals that specify a tool of type i as one they would like to acquire, in other words, the *demand* for tool type i.

3. Any cycle in the graph represents a solution to the $\text{ECP}^{\oplus}$ instance associated with the tool graph.

The tool-graph representation has several advantages over the proposal-graph representation.

conciseness Since there can be multiple occurrences of a proposal in a proposal bag, a tool graph is generally a multi-graph. Because a *proposal graph* represents different occurrences of the same proposal as *distinct* nodes, the tool graph of a composition problem will generally be more concise than the proposal graph.

satisfiability From a tool graph, it can easily be determined whether the composition problem is completely satisfiable. This is so iff the indegree equals the outdegree for all nodes.

viability With the tool-graph representation, it is easy to determine whether a given graph is the tool graph of a viable composition problem. Any directed multi-graph without *loops*[5] can occur as the tool graph of a viable composition problem.

Due to the above advantages, the tool-graph representation offers better opportunities to analyze the composition problem. To demonstrate this, we now return to the question whether ECP^* should be qualified as *superior* to $\text{ECP}^{\oplus}$. In Sect. 3.2.3, we argued that both the LBF and the LCF heuristic express the belief that it is advantageous to compose a large portion of the *initial* proposals. If this is true, we can compare the adequacy of the two heuristics in terms of the number of initial proposals accepted by LBF- and LCF composition/revision. In this respect, LBF is at least not *inferior* to LCF, since it selects at least as many such proposals as LCF does. This raises the question to what extent LBF is superior to LCF. While it is difficult to answer this question for UUCP^* and $\text{UUCP}^{\oplus}$, it is easy to conceive a tool graph which shows that ECP^* can be superior to $\text{ECP}^{\oplus}$ to an arbitrary extent.

The tool graph in Fig. 3.6 represents an initial proposal bag of eight exchange proposals involving six different tool types. Because we only look at the number of initial proposals in a solution, we can assume that revision is all-or-nothing. Either the original proposal is resubmitted, or it is withdrawn. In this case, the initial proposals accepted by LBF composition/revision are those selected by LBF composition in the first iteration of the composition/revision process. In contrast,

[5]Loops are arcs from a node to itself

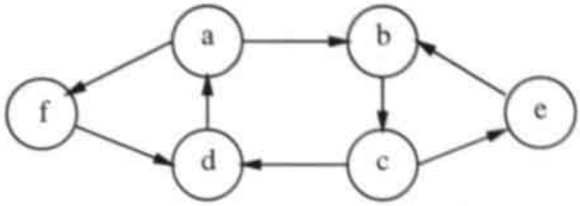


Figure 3.6: Superiority of the LBF over the LCF heuristic.

some of the initial proposals accepted by LCF composition/revision can stem from later iterations, if (and only if) there are cycles in the original tool graph which are arc-disjoint with the maximal cycle selected in the first iteration. However, this is not the case in the tool graph of Fig. 3.6. Here, the largest cycle is *abcd*, and there is no other cycle in the tool graph that is arc-disjoint with *abcd*. Hence, the outcome of LCF composition/revision is the cluster $\{\{a \rightleftharpoons b\}, \{b \rightleftharpoons c\}, \{c \rightleftharpoons d\}, \{d \rightleftharpoons a\}\}$ of the four proposals represented by the arcs of the cycle *abcd*. In contrast, the LBF heuristic selects the set of cycles $\{bce, afd\}$. Again, there are no arc-disjoint cycles left once this selection is made, but the associated solution comprises six proposals, instead of four.

The example in Fig. 3.6, which features a tool graph with a central cycle of four arcs, adjoined by two peripheral cycles of three arcs, can be generalized to a graph featuring a central cycle of $2n$ arcs, adjoined by n peripheral cycles of $2n-1$ arcs. If we measure the relative adequacy of the LCF heuristic in comparison with the LBF heuristic in terms of the fraction of accepted proposals, then the relative adequacy of the LCF heuristic in the generalized example equals $\frac{2n}{n(2n-1)} = \frac{2}{2n-1}$. In other words, in terms of the percentage of accepted initial proposals, LCF composition can be inferior to LBF composition to an arbitrary degree.

However, the phrase "inferior to an arbitrary degree" should not be taken too heavily. After all, it pertains only to the worst case. On average, the LCF heuristic may well be nearly as proficient as the LBF heuristic. Moreover, while the LBF heuristic is never inferior to LCF in terms of the percentage of *fully* satisfied agents, it *can* be inferior in terms of the community utility of (i.e., the average degree of satisfaction with) the final allocation. In the example of Fig. 3.6, we assumed that revision is all-or-nothing, that is, no agent has any satisfactory alternatives for its first preference. If we assume instead that the victims (i.e., the agents whose proposals are rejected) of LCF do have satisfactory alternatives, while the victims of LBF do not, LCF may well lead to an allocation of a higher community utility than LBF.

Suppose, for example, that the second-best alternative for all agents is to stick to their current endowment. If this alternative renders an agent-level utility of 99% for the 4 LCF victims, and only 1% for the 2 LBF victims, the community utility obtained with LCF equals $U_{LCF} = \frac{4 \cdot 100 + 4 \cdot 99}{8 \cdot 100} > 99\%$, while $U_{LBF} = \frac{6 \cdot 100 + 2 \cdot 1}{8 \cdot 100} < 76\%$.

3.2.7 Tool-graph balancing

In TR problems that involve many instances of each tool type, a composition attempt will often lead to the execution of the large majority of submitted proposals. On average, it is easier to solve such composition problems in a top-down manner, by determining which proposals should be *rejected*. Obviously, the complementary bag is a solution to ECP* if the size of this 'dump bag' is minimal. Because the aim of removing proposals is to achieve balance between tool supply and demand, we refer to the problem of finding a dump bag of minimal size as *the balancing problem*.

To turn an imbalanced composition problem into a balanced one without relaxation,[6] two goals must be pursued:

- reduction of the demand for undersupplied tools;
- reduction of the supply for oversupplied tools.

Undersupplied tools are represented in a tool graph as sources (i.e., nodes where the outdegree exceeds the indegree), while oversupplied tools are represented as sinks. Hence, demand reduction for undersupplied tools can be performed in the tool graph by removing one or more outgoing arcs at a source. Likewise, the supply of oversupplied tools can be reduced by removing incoming arcs at a sink.

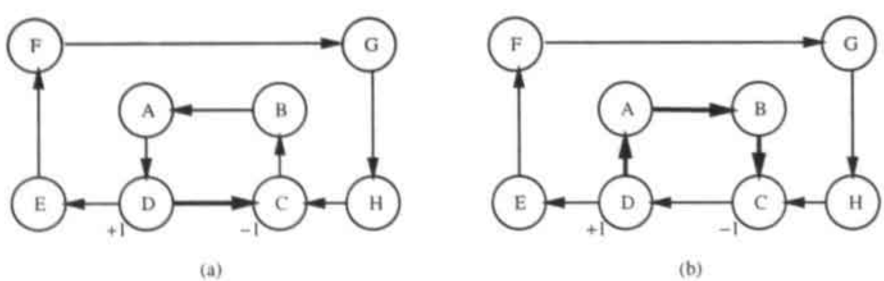


Figure 3.7: Two relatively easy instances of the balancing problem.

If we are in the fortunate situation of Fig. 3.7(a), faced with only one (+1) source, which is adjacent to a (-1) sink, we can achieve both goals at once by removing the

[6]Here, "without relaxation" means: without adapting existing proposals, or introducing new ones.

arc from source to sink. In the somewhat less fortunate situation of Fig. 3.7(b), where the arc between source and sink points in the wrong direction, we must delete other arcs. This, however, merely *shifts* the source and the sink to originally balanced nodes. Obviously, to minimize the number of deleted arcs, we should proceed along the *shortest path* from source to sink.

The situation is more complicated in the presence of multiple sources and sinks. We have to determine shortest paths from sources to sinks, and find out *which assignment* of sources to sinks involves a minimum total number of deleted arcs. Linear assignment problems can be solved in polynomial time (Papadimitriou & Steiglitz, 1982), but the balancing problem is generally not a *linear* assignment problem: the cost (i.e., the number of arc deletions) associated with assigning some source to some sink generally depends on the assignment of other sources to other sinks. This is due to the fact, that some of the shortest paths may have shared arcs. The presence of shared arcs implies that the shortest path between a source and a sink may *cease to exist* as a consequence of deleting a shortest path between another pair of nodes. Fig. 3.8 presents an example.

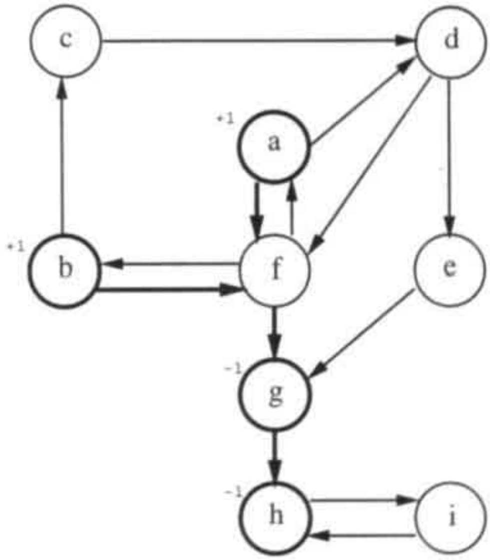


Figure 3.8: A balancing problem with competing shortest paths.

The balancing problem features two sources, associated with the undersupplied tool types a and b, and two sinks, associated with the oversupplied types g and h. The shortest paths from a and b to g and h are afg, bfg, $afgh$, and $bfgh$, respectively. Thus, the path sets $\{afg, bfgh\}$ and $\{afgh, bfg\}$, would seem to constitute alternative solutions to the balancing problem, each comprising the deletion of 5 arcs.

However, in both cases, the shortest paths from source to sink share the *single* arc from f to g. Since a single arc cannot be deleted twice, neither of the two path sets constitutes a solution to the balancing problem. If we start with deleting the shortest path from a to one of the sinks, the shortest path from b to h in the reduced tool graph becomes *bcdegh*. In this case, the solution, $\{afg, bcdegh\}$, comprises 7 deletions. If we choose to start with source b, we arrive at the solution $\{bfg, adegh\}$, comprising only 6 deletions.

The lesson in this scenario is that, in the presence of *bottlenecks* (i.e., shared arcs) in the set of shortest paths from sources to sinks, we have to try out *different orders* of shortest-path deletion. The worst-case complexity, in this case, is $O(n \cdot e \cdot t!)$, where n and e denote the number of nodes and arcs in the tool graph, and t denotes the *tension* between supply and demand.[7] Summarizing, balancing tends to be computationally expensive for composition problems with bottlenecks and a high tension.

3.2.8 Incorporation of one-way proposals

Up to this point, we have confined our attention to composition problems comprising elementary *exchange* proposals. To express composite exchanges in terms of elementary proposals, we also need one-way proposals. If an agent currently endowed with a type-x workstation desires to exchange this workstation for two type-y PCs, it can only express this in terms of elementary proposals if it is allowed to submit the one-way proposal $\{\rightleftharpoons y\}$, next to the exchange proposal $\{x \rightleftharpoons y\}$.

To incorporate one-way proposals, we introduce a virtual tool type $\perp$, different from any real tool type. The one-way proposals $\{a \rightleftharpoons\}$ and $\{\rightleftharpoons a\}$ are represented as $\{a \rightleftharpoons \perp\}$ and $\{\perp \rightleftharpoons a\}$. The node representing the special tool $\perp$ in the tool graph is treated like any normal tool node. This reflects that a mediator engaged in balancing cannot overcome a discrepancy between supply and demand by merely convincing agents to supply additional specimens of the undersupplied tools. To arrive at an acceptable reallocation, the oversupply must be taken care of also. Of course, it is usually easier to persuade agents to keep their tools than it is to persuade them to release tools, but this is a matter of relaxation, rather than balancing.

This is illustrated in Fig. 3.9, which shows a tool graph equal to that of Fig. 3.8, except for an additional, one-way supply of tool types a and b from the $\perp$-node. The sources at the a- and b-nodes have disappeared, but a new source (of multiplicity 2) is formed at the $\perp$-node. Consequently, to solve this balancing problem, the same arcs as in Fig. 3.8 have to be deleted, in addition with those emanating from the $\perp$-node. Thus, convincing agents to supply additional tools does not solve this

[7] The tension is defined as the sum of the multiplicities of the sources (or the sinks), where the multiplicity of a source (or sink) is equal to the absolute value of its indegree minus its outdegree.

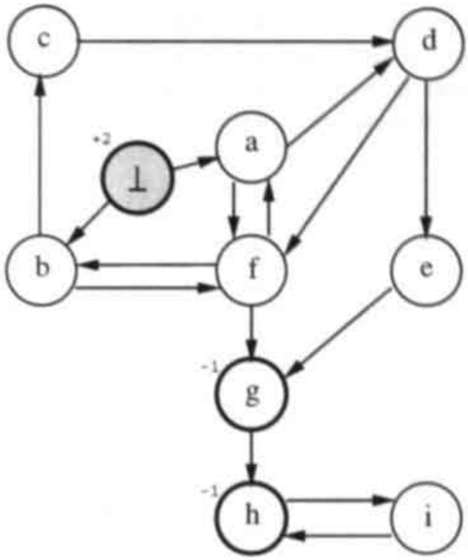


Figure 3.9: The representation of one-way proposals $\{a \rightleftharpoons\}$ and $\{b \rightleftharpoons\}$.

balancing problem, unless the mediator also convinces some agents to accept the surplus g- and h-tools. This would be reflected in the tool graph of Fig. 3.9 by two additional arcs from the g- and h-nodes to the $\perp$-node. With the addition of these arcs, the source at the $\perp$-node disappears, as well as the sinks at the g- and h-nodes. Since there are no sinks left, the relaxed composition problem associated with this tool graph is completely satisfiable.

3.2.9 Elementary composition via linear assignment

In Sect. 3.2.7, we investigated the solution of elementary composition problems via balancing of the associated tool graphs. It appeared that the balancing method is adequate for (elementary) composition problems with relatively low tension, but not very efficient in the general case.

In this section, we describe a procedure to reformulate elementary composition problems as linear assignment problems that is applicable to problems with low as well as high tension. Linear assignment amounts to what was referred to earlier as *untyped* assignment. The assignment, in this case, is not one of sources to sinks in the tool graph, but one of items to agents. The standard representation for linear assignment problems is a square utility matrix, where the element in row i, column j of this matrix denotes the utility of item j to agent i. A solution to the linear assignment problem is defined as an association of each row of the matrix with a column, such that (1) each column is used exactly once, and (2) the sum of the thus selected matrix entries is maximal. In the following we specify how these matrix entries must be defined to let a solution of the linear assignment problem correspond to a solution of a given elementary composition problem.

Definition 3.8 (linear assignment problem)
A linear assignment problem can be formulated in terms of a square (n-by-n) utility matrix Q as

Determine a permutation π of $1, 2, \ldots, n$ such that $\sum_{i=1}^{n} Q_{i,\pi(i)}$ is maximal.

△

To translate an ECP* instance into an assignment problem, we must specify the matrix Q of definition 3.8 in terms of the bag $\{\{s_i \rightleftharpoons d_i\}\}_{i \in I}$ of exchange proposals that defines the ECP* instance (i.e., the bag β in Def. 3.7).

As a first step, we reformulate ECP*. Remember that "adherence to the exchange proposal $\{s \rightleftharpoons d\}$" entails that the agent will either receive tool type d in exchange for tool type s, or it will keep s.

Lemma 3.9
Let $\beta = \{\{s_i \rightleftharpoons d_i\}\}_{i \in I}$ be a bag of elementary exchange proposals with $I = \{1, \ldots, n\}$, $s_i, d_i \in R = \{1, \ldots, m\}$, and $m \leq n$. Then the ECP instance defined by β can be reformulated as: "Find an assignment $a : I \rightarrow R$ of tools to agents which maximizes the number of proposals in β that are executed, under the constraints of conservation of tools and adherence to exchange proposals." Formally, this can be expressed as:*
Maximize

$$|\{i \mid a(i) = d_i\}| \tag{3.10}$$

under the constraints

$$(\forall r \in R) \;\; |\{i \mid a(i) = r\}| = |\{i \mid s_i = r\}| \tag{3.11}$$

and

$$(\forall i \in I) \;\; a(i) = s_i \; \vee \; a(i) = d_i \tag{3.12}$$

Proof.
The expression (3.10) denotes the number of accepted proposals, constraint (3.11) expresses conservation of tools (i.e., global feasibility), and constraint (3.12) expresses adherence to the proposals $\{s_i \rightleftharpoons d_i\}$. ■

Now that we have acquired a formulation of ECP* that is closer to our definition of a linear assignment problem, we turn to the utility matrix Q in this definition. This matrix Q is constructed in two steps. In the first step, we translate the proposals of the agents into an n-by-m matrix $\widehat{Q}$. In the second step, this matrix is transformed into the n-by-n matrix of Def. 3.8. The variable S_k in the second step denotes the supply of tool type k, that is, $S_k = |\{i \in I \mid s_i = k\}|$.

1. Construct an n-by-m matrix $\widehat{Q}$ defined by

$$\widehat{Q}_{i,j} = \begin{cases} 1 & , \quad j = d_i \\ 0 & , \quad j = s_i \\ -n & , \quad otherwise \end{cases}$$

2. Construct the n-by-n matrix Q from the n-by-m matrix $\widehat{Q}$ by copying columns from $\widehat{Q}$ to Q while replicating column k of $\widehat{Q}$ into S_k identical columns of Q. Let f be the function that maps column numbers of Q to corresponding column numbers of $\widehat{Q}$, that is, $Q_{i,j}$ originated from $\widehat{Q}_{i,f(j)}$.

Proposition 3.10
If $\hat{a} = [q_1, q_2, \ldots, q_n]$ is a solution of the assignment problem Q, with Q constructed in the above manner from an ECP* *instance $C = \{\{s_i, d_i\}\}_{i=1,\ldots,n}$, then $a(i) = f(q_i), i = 1, \ldots, n$ defines a solution to C.*

Proof.
We prove this by ensuring that each of the three conditions expressed in Lemma 3.9 (by the equations (3.10), (3.11), and (3.12)) is satisfied. First, we prove the adherence to proposals expressed in (3.12). In view of the definition of Q, it suffices to prove that none of the selected elements Q_{i,q_i} equal $-n$.

$$(\forall i \in I) \;\; Q_{i,q_i} \neq -n \tag{3.13}$$

Suppose that Eq. 3.13 would not hold, that is, $Q_{k,q_k} = -n$ for some k. Then, the total utility $\sum_{i=1,\ldots,n} Q_{i,q_i}$ would be negative, since

$$\sum_{i=1}^{n} Q_{i,q_i} = Q_{k,q_k} + \sum_{i \neq k} Q_{i,q_i} \leq Q_{k,q_k} + (n-1) \cdot 1 = -n + (n-1) = -1$$

This would imply, however, that $\hat{a}$ is suboptimal, since any permutation π_0 that satisfies $(\forall i \in I)\; f(\pi_0(i)) = s(i)$ (i.e., all agents keep their original tool types) renders a *higher* total utility (viz., 0). Hence, Eq. 3.13 holds for any solution q_i of the assignment problem.

From the validity of Eq. 3.13, and step 1 in the construction of Q, it follows immediately that all entries Q_{i,q_i} selected from Q are either 0 or 1. Hence, constraint 3.12 (adherence to proposals) of the reformulation of ECP* in Lemma 3.9 is met. Constraint 3.11 (conservation of tools) is met also, since $\hat{a}(\cdot)$ only *permutes* the column indices of Q, S_j of which are mapped to tool type j by $f(\cdot)$ (due to step 2 in Q's construction). All that remains to be checked is that maximization of the quantity $\sum_{i=1}^{n} Q_{i,q_i}$ also leads to maximization of $|\{i : a(i) = d_i\}|$ (3.10). This is obviously so, since all of the q_i are either 0 or 1, and $q_i = 1 \Leftrightarrow a(i) = d_i$. ∎

Since the time complexity of the above translation of elementary composition into linear assignment is quadratic in the number of elementary proposals, and linear assignment problems can be solved with combinatorial optimization techniques in (third-order) polynomial time (Papadimitriou & Steiglitz, 1982), the translation procedure defines a composition algorithm which is third-order polynomial in the number of proposals.

Corollary 3.11 ECP* *is tractable.*

3.3 Relaxation-Based Solution of TR

As a solution strategy within CDN, relaxation is not described as lucidly as composition. In (Sathi & Fox, 1989), relaxation is described as an ensemble of techniques used in inter-human negotiations, without specifying how these techniques are combined. However, we do not need a precise description of CDN-style relaxation to conclude that it is not adequate for our purposes. Relaxation in CDN entails that the mediator tries to persuade agents to accept specific relaxations. Since the agents do not take any initiatives in this scheme, the full weight of the relaxation task rests on the shoulders of the mediator. Hence, the mediator is bound to be a bottleneck, in computation as well as design.

We prefer a form of relaxation where the mediator merely provides incentives for relaxation, leaving the search for acceptable alternatives to the agents. A simple mechanism featuring such incentive relaxation is the exchange of goods by means of a *Walrasian auction* (Hildenbrand & Kirman, 1988).

3.3.1 The Walrasian auction

The idea behind the Walrasian auction (conceived as a market model by the 19th-century, French economist Léon Walras (Walras, 1874)) is that changes of commodity prices can act as incentives for agents to change their reallocation proposals in a manner that affects the demand for and supply of commodities. To reduce the scarcity of a good, one can raise its price. This will make it more attractive to sell the good, and less attractive to buy it. Likewise, one can attempt to reduce an oversupply of some good by lowering its price.

Like any auction, the Walrasian auction is a *centralized* market: all agents communicate their bids to the auctioneer, who aggregates these into a scarcity profile, and sets new prices depending on the scarcities of the various goods. Instead of a single auctioneer, one can employ multiple ones, each dealing with one specific tool type. If these auctioneers reside on different processors, this can enhance the robustness of the auction in terms of hardware failures. On the design level, however, this scheme is equivalent to an auction with a single auctioneer.

It is also possible to get rid of the auctioneer altogether, by distributing its functionality over the traders, but this increases the conceptual complexity of the Walrasian auction, and incurs a communication overhead (Hurwicz, 1986). An example of such distributization, and the communication overhead it incurs, will be presented later, in Sect. 5.5.5. For now, however, we assume that there is one auctioneer.

The prices in Walras' original framework do not involve any money; they are exchange ratios. As such, the Walrasian auction is essentially a matter of barter trade. Traders pay for a commodity bundle that they want to obtain with goods that they currently possess. An agent may for instance pay for three pounds of rice and 2.21 gallons of beer with 10.5 pounds of potatoes and 5.4 pounds of cherries. Obviously, this presumes that the goods involved are *divisible*.

A bid in a Walrasian auction specifies the commodity bundle an agent would like to possess in lieu of its current endowment. Except for the context of divisible goods, this amounts — in our terminology — to a single, composite exchange proposal with a supply bag that equals the agent's endowment.

In economic literature, one generally assumes that the auctioneer already knows all initial endowments. Thus, the agents can specify their desired final endowments instead of the desired exchanges. In this context, the *budget* of an agent is the set of commodity bundles that it can afford to bid on, in view of its current endowment and the going prices of commodities. An agent always bids on a commodity bundle of maximal utility within its budget. In general, this behavior initially leads to bids that are not jointly satisfiable. The auctioneer tries to reduce the tension between supply and demand by adjusting prices, after which the agents reassess their budget and submit new bids. The bidding stops when supply equals demand for all goods. Such a state — in which the exchanges implied by the last-submitted bids are jointly executable — is called a Walrasian equilibrium, and the allocation that results from executing the bids is called a Walrasian allocation. The attainment of Walrasian equilibrium is referred to as *market clearance*.

3.3.2 General Equilibrium Theory

The notion of Walrasian auction lies at the basis of the extensive General Equilibrium Theory (GET) of micro-economics (Hildenbrand & Kirman, 1988; Takayama, 1985). GET generalizes Walras' market model of traders by incorporating producers. Whereas trading can be described as an activity aimed at obtaining a commodity bundle of higher *utility* than ones current endowment, producers are more aptly characterized as *profit* maximizers. Hence, contrary to the original Walrasian framework, GET involves money as an integral component of its market models. Here, money is formally defined as a commodity that can be used to compensate any shortage of other commodities in the endowment of any agent.

GET has been used by Wellman (1992) as the basis for a computational reallocation framework to tackle transportation in congestive networks. The framework is known as the market-oriented programming (MOP) approach (Wellman, 1992; Wellman, 1994a; Wellman, 1994b).

Although Wellman's (1992; 1994b) MOP implementation is called WALRAS, the employed equilibrium concept is closer to a Marshallian than to a Walrasian equilibrium (Takayama, 1985), and the auction in WALRAS differs from the Walrasian auction (described in Section 3.3.1) in several respects. The auctioneer, for example, needs to be somewhat more sophisticated than the prototypical Walrasian auctioneer, because agents do not simply specify their preference under the going prices. They also convey how they *would* bid if prices were slightly different.

Also, the original MOP domain of transportation in congestive networks does not involve allocation in the usual sense (of, for example, Def. 2.20), because the amount of goods (in this case, transportation capacity) available in the agent community is principally *infinite*: Links in congestive networks are modeled as containers with a capacity that is not bounded by any fixed constant, but by the fact that transportation costs become excessive as the network becomes congested. Despite — or rather, because — of this anomaly, this problem domain is susceptible to solution by means of a market-based algorithm. Later on, MOP has been applied successfully to other optimization problems that do involve a finite supply of (divisible) commodities, such as temperature regulation (Huberman & Clearwater, 1995), and power load management (Ygge, 1998).

An attempt to apply MOP to a configuration design problem was less successful. It appeared to be difficult to attain a (near-)equilibrium state. This problem was attributed by Wellman (1994a) to the discrete nature of configuration design. This can be interpreted as an indication that the application of market-based approaches such as MOP and Walrasian exchange is problematic in domains involving indivisible goods. On the other hand, the problem may also be specific to configuration design. To gain more insight in this issue, we have investigated to what extent the economic literature provides theoretical support for Walrasian exchange in markets with indivisible goods. The discussion of our findings requires a more precise characterization of the concepts involved than was given up to this point. Hence, we describe the key notions of Walrasian equilibrium and Walrasian auction in formal terms.

In contemporary economic literature, and DAI literature on research rooted in economics, such as MOP, the term 'Walrasian equilibrium' is often used as a synonym for the GET term 'competitive equilibrium' (Ygge, 1998, p. 200). We do make a distinction between the two concepts. In the following, the terms Walrasian equilibrium and Walrasian auction have their *original* semantics, implying that the context is one of pure exchange. Neither money nor producers play a role. The only deviation from Walras' original framework is that we specifically look at the potential of the Walrasian auction in the face of *indivisible* goods.

3.3.3 Walrasian equilibria

We start with the formal definition of a Walrasian equilibrium in its original context: that of an exchange economy with a finite number of agents and a finite number of *divisible* commodities. Boldface lowercase symbols denote vectors, and boldface capitals denote matrices.

Let $I = \{1, \ldots, n\}$ denote a community of agents, and $R = \{r_1, \ldots, r_l\}$ the set of commodities available in this community. A commodity bundle allocated to — or desired by — agent i is represented as an l-tuple $\mathbf{x}_i$ of nonnegative real numbers. The j-th component x_{ij} of such a tuple denotes the amount of good r_j allocated to agent i. Thus, an entire allocation of goods is represented by a matrix $\mathbf{X}$ where row i specifies the endowment $\mathbf{x}_i$ of agent i. In the context of a Walrasian auction, the matrix $\mathbf{X}$ specifies the *desired* allocation associated with the bids submitted by the agents. The *initial* allocation is denoted by the matrix $\mathbf{E}$, and the initial endowment of agent i by $\mathbf{e}_i$. The expression $\mathbf{x} \succ_i \mathbf{y}$ signifies that agent i prefers the commodity bundle $\mathbf{y}$ over the commodity bundle $\mathbf{x}$. Hence, a reallocation problem instance can be specified as a pair $\langle \mathbf{E}, \{\succ_i\}_{i \in I} \rangle$ of an initial allocation and a tuple of agent preferences.

Let $\mathbf{p} \in \mathbb{R}^l_+$ denote an l-tuple of strictly positive prices, which define the exchange ratios of the commodities. The *budget* $B_i(\mathbf{p})$ of agent i is the set of commodity bundles that agent i can afford under the current exchange ratios:

$$B_i(\mathbf{p}) = \{\mathbf{x}_i \in \mathbb{R}^l : \sum_{j=1..l} p_j \cdot x_{ij} \leq \sum_{j=1..l} p_j \cdot e_{ij}\} \tag{3.14}$$

In the following, we employ the usual shorthand notation $\mathbf{p} \cdot \mathbf{e}_i$ for the inner product of p and e_i. Thus, Eq.(3.14) transforms into

$$B_i(\mathbf{p}) = \{\mathbf{x}_i \in \mathbb{R}^l \mid \mathbf{p} \cdot \mathbf{x}_i \leq \mathbf{p} \cdot \mathbf{e}_i\} \tag{3.15}$$

Definition 3.12 (Walrasian equilibrium)
Let $\mathbf{X}$, $\mathbf{p}$, B_i, $\mathbf{E}$, and $\{\succ_i\}_i$ be defined as above. Then a bidding state $\langle \mathbf{X}, \mathbf{p} \rangle$ constitutes a Walrasian equilibrium of the market $\langle \mathbf{E}, \{\succ_i\}_i \rangle$ iff the following three conditions are satisfied.

(i) $(\forall i \in \{1,..,n\})\ \mathbf{x}_i \in B_i(\mathbf{p})$	*(All allocated commodity bundles are affordable)*
(ii) $(\forall \mathbf{y} \in \mathbb{R}^l)\ \mathbf{y} \succ_i \mathbf{x}_i \Rightarrow \mathbf{p} \cdot \mathbf{y} > \mathbf{p} \cdot \mathbf{e}_i$	*(Any commodity bundle preferred over an allocated one is not affordable.)*
(iii) $\sum_{i=1..n} \mathbf{x}_i = \sum_{i=1..n} \mathbf{e}_i$	*(There is conservation of goods.)*

△

Condition (iii) in Def. 3.12 is the target condition of the Walrasian auction. It is typically not fulfilled initially, while the conditions (i) and (ii) are met in every round. They describe the bidding behavior of the agents in a Walrasian auction. Condition (i) specifies that agents never bid on an endowment that they cannot afford. Condition (ii) entails that they always bid on an (affordable) bundle which maximizes their utility. Such bidding behavior is rational, provided that no agent is capable of exerting a significant influence on the evolution of prices by communicating false bids. The absence of agents that are sufficiently influential to manipulate prices is referred to as *perfect competition* in economics.

To ensure perfect competition, one usually assumes that markets comprise very many traders, each contributing only little to the total trade volume. This is a questionable assumption in CMAT, but then, we do not need perfect competition in a CMAT context, since the system designer has full control over the behavior of the agents. While rational agent behavior is desirable in CMAT, it is not an absolute necessity. In the context of OMAT, however, perfect competition is vital for the applicability of a Walrasian auction.

Because the agents in a Walrasian auction respond in a self-centered manner, the attainment of condition (iii) — also known as *market clearance* — is the task of the auctioneer. The common strategy to accomplish this is referred to as tâtonnement. This is French for "groping". In its prototypical form, tâtonnement resembles gradient descent. It entails that the prices of goods which are undersupplied are increased (to reduce the demand and stimulate the supply of these goods), and the prices of oversupplied goods are lowered (to stimulate their demand and reduce their supply). Whereas the term tâtonnement is often used in economic literature as a reference to *any* price-setting algorithm that attempts to attain zero excess demand for all goods, we reserve it for *simple* schemes that are *solely* based on scarcity information, such as the above, prototypical form.

The simplicity of tâtonnement implies that the Walrasian auction constitutes a relaxation-based approach to reallocation in which the central coordination module (i.e., the auctioneer) is less of a computation bottleneck than the mediator in CDN-style relaxation. Hence, the Walrasian auction can be qualified as operationally decentralized.

Of course, this presumes that tâtonnement will indeed render a Walrasian equilibrium within an acceptable number of rounds. This is not always the case, but economic literature provides a wide array of useful sufficient conditions for the existence of Walrasian equilibria and the effectiveness of tâtonnement in markets with divisible commodities (Hildenbrand & Kirman, 1976; Aliprantis *et al.*, 1989). The question is, of course, whether a similar statement can be made with respect to indivisible commodities, and, more particularly, TR problems.

3.3.4 Walrasian auctions for TR

As far as a theoretical basis for the application of a Walrasian auction to TR problems is concerned, we identify three minimal requirements:

1. an *existence theorem* which guarantees the existence of a Walrasian equilibrium in TR, or a relevant subdomain of TR;
2. an efficient *auction protocol* (such as tâtonnement) that yields a Walrasian equilibrium for TR problems within an acceptable number of bidding rounds;
3. a guarantee that the Walrasian allocations rendered by the auction are of acceptable quality in a relative-utilitarian sense.

We discuss each of these requirements in a separate section.

3.3.5 Walrasian equilibria in TR problems

As mentioned earlier, the economic GET offers a broad spectrum of sufficient conditions for the existence of Walrasian equilibria in markets with divisible goods. Whereas some of these conditions (e.g., gross substitutability[8]) are criticized for being unrealistic (Hildenbrand & Kirman, 1988), the economic theory on equilibrium existence can be qualified as an adequate theoretical foundation for computational mechanisms to reallocate divisible goods (such as MOP).

Unfortunately, economic theory has much less to offer with respect to the existence of Walrasian equilibria in TR markets. There are a few papers on the existence and properties of Walrasian equilibria in markets with indivisible goods, but all of these concern *assignment markets*, that is, tool reallocation problems in which each agent possesses and desires only one tool. Moreover, the majority of papers on this subject pertains to markets with money (Kaneko, 1982; Quinzii, 1984; Gale, 1984; Tadenuma & Thomson, 1991). Although Quinzii (1984) claims that his existence result for general assignment markets with money also extends to assignment markets *without* money, it has been shown by Wako (1987) that, in the absence of money, Quinzii's

[8]Gross substitutability entails that raising the price of one good never incurs a decrease of the demand for *any* other good.

result only covers the degenerate case where there is guaranteed to be no trade at all.

We have found only a handful of papers in economic literature on Walrasian *exchange* in markets with indivisible goods. Each of these pertained to the special case of *untyped* assignment. In an untyped assignment market, each good comprises a single unique item. Hence, each item can be assigned a unique price (Shapley & Scarf, 1974; Roth & Postlewaite, 1977; Wako, 1984; Ma, 1994).

The economics literature does not distinguish between untyped and typed goods. However, from the proof of the theorem on Walrasian equilibrium existence in assignment markets (Shapley & Scarf, 1974), it is clear that the goods are presumed to be untyped: The proof hinges on the assumption that it is allowed to assign a different price to each item, rather than to each type of item.

In principle, we can use an algorithm that renders a Walrasian equilibrium in untyped assignment markets to solve a tool reassignment problem, by treating the (typed) tools as if they were (untyped) items. However, this implies that we must assign prices to all tools instead of only the tool *types*. This is not an attractive prospect, especially if there are many more agents — and hence, many more tools — than tool types. Moreover, though we can speak of a Walrasian equilibrium in an untyped assignment market, the term Walrasian auction is not really appropriate: the auctioneer in (Shapley & Scarf, 1974) uses a protocol in which price changes depend on the submitted bids, instead of the induced scarcities. Moreover, this protocol is such that agents are forced, rather than motivated, to adapt their bids.

The absence of satisfactory equilibrium existence theorems is not the only hindrance for the application of Walrasian exchange to TR problems. Ending up in disequilibrium is also more *troublesome* in the face of indivisibility. In computational implementations of General Equilibrium Theory, such as MOP, one generally aims for *approximate* equilibrium. As soon as the scarcities of all goods are *close* to zero, a reallocation takes place, in which (some of) the agents do not get exactly what they asked for (Wellman, 1992). This is acceptable in the context of *divisible* goods: Getting 7.496 tons of fuel instead of 7.5 is not really painful. However, the difference between getting one oil tanker instead of two - or worse, none instead of one - is another matter.

3.3.6 An auction protocol for TR

In view of the observed lack of economic theory on the existence of Walrasian equilibria in TR markets, it seems premature to ponder over auction protocols to attain such equilibria. Nevertheless, we will discuss the Top-Trading-Cycles (TTC) algorithm that can be used as an auction protocol to attain a Walrasian equilibrium in an untyped assignment market (Shapley & Scarf, 1974).

1. Assign the same positive price to all items, and label all agents as unselected.
2. Sollicit bids from the agents by communicating the current prices.
3. Compute the supply of and demand for the items from the submitted bids.
4. If supply equals demand for all items, a Walrasian equilibrium is reached. Perform the exchanges proposed in the bids, and finish.
5. Otherwise, portray the bids submitted by the *unselected* agents in a tool graph, representing agents who refrained from bidding by an arc from the agent's endowment to itself. Select an arbitrary cycle in the graph, label the agents represented by the arcs in this cycle as selected, and decrease (e.g., halve) the price of the items possessed by the remaining unselected agents.
6. Go to step 2.

Figure 3.10: The TTC algorithm, as a protocol for the auctioneer

Our motivation is twofold. First, we will use TTC later on to make an important observation with respect to the quality of Walrasian allocations in TR markets.

Second, presentation of the TTC algorithm enables us to point out that, even though TTC computes a Walrasian equilibrium, it should not be labeled as an auctioneering protocol for the Walrasian auction. In Sect. 3.1.2, we described the Walrasian auction as a relaxation-based approach to reallocation, in which the auctioneer provides incentives for agents to relax their proposals in the form of prices. The TTC algorithm does not provide incentives. It *forces* the agents to adjust their proposals, by deliberately keeping them from attaining certain goods. In this respect, the TTC algorithm is more akin to composition/revision (cf. Sect. 3.2.2) than to tâtonnement.

In Fig. 3.10, the TTC algorithm is specified in the form of an auction protocol that tells the auctioneer how to change the prices in response to the agents' bids. Since we are dealing with reassignment, the bids submitted by the agents are elementary proposals that reflect the agents' first preferences under the going prices.

If agents are never indifferent between different items, the TTC algorithm described in Fig. 3.10 reaches a Walrasian equilibrium after at most n price adjustments, where n is the number of agents. To see this, we make three assertions.

(i) Every arc in a tool graph of unselected agents points to a node within the tool graph:
If a cycle is selected in some round, any arcs that point from a node not in this cycle to a node in the cycle disappear in the next round, since the agent in possession of the item represented by the outside node can no longer afford the item represented by the node in the cycle. Hence, there are no arcs in the tool graph of unselected agents which point to a node of an already selected top-trading cycle.

(ii) There exists a cycle in each tool graph of unselected agents:
In view of assertion (i), and the fact that the tool graphs are finite, any path that starts at an arbitrary node in the graph either returns to a node on the path, or it ends at some node which points to itself. In the first case, there is a cycle of at least two arcs. In the second case, there is a 1-arc cycle at the terminal node. Hence, there always is a top-trading cycle (comprising at least one agent) in the tool graph of unselected agents.

(iii) Once an agent is labeled as selected, it sticks to its last-submitted bid:
Once an agent is labeled as selected, subsequent price changes only *lower* the prices of items that were already affordable to it. Hence, the budget of an selected agent (i.e., the items it can afford) never changes. Because the agent's bid at the time of its selection represents the agent's (unique!) first preference from this budget, it will stick to this bid in all subsequent rounds.

Assertion (ii) implies that one top-trading cycle, comprising at least one agent, is selected in each round, after which the agents in the cycle are labeled as selected. Hence, there are no unselected agents left after at most n rounds. The reassignment defined by the bids of the selected agents is a Walrasian allocation, because each of the three conditions in Def. 3.12 on page 69 is satisfied:

1. Since each agent receives an item from within its top-trading cycle, and all items in a top-trading cycle have the same price, each of the allocated items is affordable.
2. The motivation of assertion (iii) implies that no agent prefers any item within its budget over the allocated item.
3. Since the reassignment is defined by a set of disjoint cycles, there is conservation of items.

TTC, as described above, is a simple algorithm. However, this is solely due to the triviality of untyped reassignment. Since TTC comes down to composition/revision with random cluster composition, any *generalization* of TTC from untyped reassignment to TR problems is bound to be intractable (cf. Sect. 3.2.2).

3.3.7 Economic measures for allocation quality

The quality of Walrasian allocations is the object of study in the subfield of *welfare economics*. This field combines the two main economic themes: cooperation and competition (Hildenbrand & Kirman, 1988). In economics, the quality of solutions rendered by the competitive Walrasian auction is defined in terms of notions which stem from cooperative game theory.

These notions have two characteristics that distinguish them from solution quality notions in other mathematical fields, such as combinatorial optimization and control theory. First, they are *properties* rather than measures for solution quality. This is a consequence of the fact that they are based on *ordinal* agent-level utilities (i.e., preferences), instead of cardinal agent-level utilities or costs. Second, their usage implies that quality is equated with *stability*: A solution is considered optimal if no group of agents is motivated and capable to revolt against it.

Definition 3.13 (Pareto optimal)
An allocation is Pareto optimal if no other allocation exists that is as least as good for all agents, and better for at least one. △

Definition 3.14 (weakly Pareto optimal)
An allocation is weakly Pareto-optimal if no other allocation exists that is better for all agents. △

Definition 3.15 (individually rational)
An reallocation is individually rational if the original allocation is not better than the resulting one for any agent. △

In the following definition of core, the phrase "the group G can attain the allocation x" means that x can be arrived at from the initial allocation by exchanging tools within G. In this context, a group of agents can be *any* nonempty subset of the agent community, including sets of one agent, and the entire community.

Definition 3.16 (core)
The core of an exchange economy consists of those allocations that cannot be improved upon, in the sense that no group of agents in the community can attain an allocation that is better than a core allocation for each agent in the group. △

Definition 3.17 (strong core)
The strong core consists of those allocations that cannot be improved upon, in the sense that no group of agents in the community can attain an allocation that is at least as good for all members of the group, and better for at least one. △

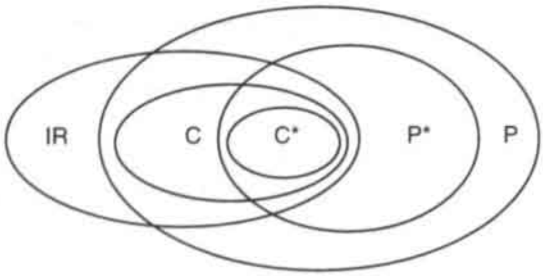


Figure 3.11: Venn diagram of the core (C), strong core (C*), Pareto-optimal set (P*), weakly Pareto-optimal set (P), and the set of individually rational allocations (IR).

These five concepts are related in the manner illustrated in Fig. 3.11. The strong core is a subset of the core. Core membership implies individual rationality and weak Pareto optimality. Strong-core membership implies individual rationality and Pareto optimality. It is easy to see that these inclusion relationships hold, from the observation that the conditions that must be satisfied to assert individual rationality or Pareto optimality are in fact *special cases* of the conditions for core membership. To ensure that an allocation is in the core, we must check for *each* subgroup in the community that this group will not revolt by refusing to exchange tools with agents that are not in the group. Individual rationality involves the special case in which only revolts by individual agents matter, and Pareto optimality pertains to the case where the absence of revolt need only be ensured for the community as a whole.

In the context of indivisible commodities, these optimality concepts are related to Walrasian equilibria in the following manner. In an untyped assignment market, the relation between the core C, the strong core C^*, and the set W of Walrasian allocations is (Wako, 1984)

$$C^* \subseteq W \subseteq C$$

Furthermore, if all agent preferences are strict,[9] $W = C^*$ (Roth & Postlewaite, 1977). Since $C^* \subseteq P^*$ (see Fig. 3.11), this implies that, in the absence of agent indifference, every Walrasian allocation is Pareto optimal.

3.3.8 Effectiveness of Walrasian allocations in TR

As pointed out in Sect. 2.2.8, our chosen measures of solution quality are the concepts of community utility and allocation effectiveness. Consequently, the above concepts of Pareto optimality and core membership are valuable to us only if they can be

[9] We speak of a strict preference if no agent is ever indifferent between different tool types.

related to these relative-utilitarian concepts. This is generally possible in money markets, but *not* in barter trade markets where all goods are indivisible.

We illustrate this with an example, in which we use the TTC algorithm to compute a Walrasian allocation.

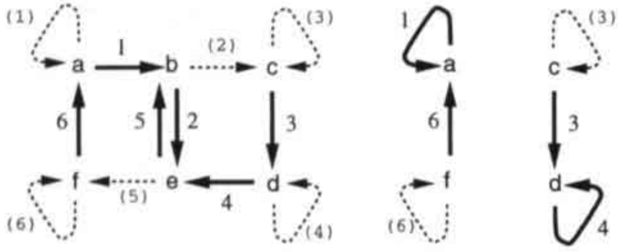
Example 3.18 (A Walrasian allocation of poor quality)
Consider an assignment market involving six agents, each endowed with a tool of a different type. We denote the agents by $1, 2, \ldots, 6$ *and the tool types by* $a, b, \ldots, f$. *Let the initial allocation be such that agent* 1 *possesses* a, *agent* 2 *possesses* b, *et cetera. We denote this initial allocation by the list* $e = [abcdef]$. *Reallocations are represented by permutations of* e. *As an example, the reallocation where agents* 1 *and* 2 *exchange their tools renders the allocation* $[bacdef]$. *The relative agent utilities are represented (as percentages) in the matrix* U *in table 3.1. The matrix reveals*

U	*1*	*2*	*3*	*4*	*5*	*6*
a	*10*	.	.	.	.	*100*
b	*100*	*10*	.	.	*100*	.
c	.	*80*	*10*	.	.	.
d	.	.	*100*	*10*	.	.
e	.	*100*	.	*100*	*10*	.
f	.	.	.	.	*90*	*10*

Table 3.1: The utility matrix (dots denote negligible utilities).

that none of the agents currently possess tools that are of significant value to them, and that, for agents 2 *and* 5, *acceptable alternatives exist for the tools they like best.* △

Fig. 3.12a depicts the top-trading-cycle pattern for this economy. An arrow from x to y, labeled i, expresses that agent i, currently possessing tool type x likes tool type y best. Second-best options of the agents are represented by dotted arrows. For clarity, we have represented agents which do not submit a second bid by an arc of their endowment to itself. The only top-trading cycle in the primary TTC pattern (i.e., the pattern of solid arrows) involves agents 2 and 5. Consequently, the TTC algorithm will select this cycle as the first top-trading cycle, and lower the price of all tool types, except b and e. The secondary TTC pattern (see Fig. 3.12b) that forms in response to this price change contains only singleton cycles (of agents 1 and 4). It does not really matter which one of these is selected. The result is always essentially the same: The agent that has expressed a preference for the associated tool type is no longer able to afford it in the next round, and, since there is no other tool it prefers over its initial endowment, it will, in turn, refrain from bidding. Thus,



(a) primary pattern

(b) secondary pattern

Figure 3.12: The top-trading-cycles pattern of the example economy.

all agents other than 2 and 5 drop out of the auction in subsequent rounds, each sticking to its initial endowment. When only agents 2 and 5 are left, an equilibrium is reached. Therefore, the only Walrasian allocation is $[aecdbf]$.

In this case, the strong core coincides with W. This follows from the observation that, agents 2 and 5 maximize their subjective utility only if they exchange their tools, and without the cooperation of these two agents, the other agents cannot improve on their initial utility. W does *not* coincide with the core, which, apart from the above Walrasian allocation, contains the allocations $[becdfa]$ and $[acdebf]$.

With the Walrasian allocation $w = [aecdbf]$, the sum of relative agent utilities equals $10+100+10+10+100+10 = 240$. Hence, the community utility equals $U(w) = \frac{240}{6} = 40\%$. There exists, however, an allocation *outside* W with a much higher community utility. If agents 2 and 5 are prepared to concede just a little, opting for their acceptable alternatives c and f, the allocation $[bcdefa]$ with a community utility of $\frac{100+80+100+100+90+100}{6} = \frac{570}{6} = 95\%$ emerges. This non-Walrasian allocation, which is optimal in the relative-utilitarian sense, is not even in the core. This indicates that neither the notion of Walrasian equilibrium, nor the more general notions of core, strong core, and Pareto optimality constitute adequate solution quality notions in the present context.

It is important to understand *in what situations* the core allocations rendered by the Walrasian auction are poor solutions in a relative-utilitarian sense. Why, for example, does the auction do such a poor job in the problem instance described in Ex. 3.18? In this example, the low performance is partly due to the fact that two agents hold the key to wealth or starvation of the community as a whole. A small concession on their part would allow the entire community to prosper. If they do not concede, all is not lost, provided that one of the other agents is prepared to sacrifice the little utility is has left. In that case, only this agent would starve.

However, with the common semantics of rationality, self-centeredness is the dominant characteristic of a rational agent. This leaves no room for heroism or compassion. Hence, the rational agents 2 and 5 in Ex. 3.18 will close the deal that optimizes their utility, impervious to the hardship this inflicts on the others, and none of the other agents act as a savior for its peers.

In economic terms, one could say that the Walrasian auction is liable of bringing about poor allocations, due to the absence of provisions against *trust formation*. One way to prevent the formation of such trusts can be found in MOP literature: the addition of money to the market.

Ygge (1998, pp. 43-44) presents a theorem that relates Pareto optimality to utilitarian optimality. The theorem states that any *separable*[10] optimization problem can be associated with an artificial market such that all Pareto-optimal allocations in the market correspond with solutions to the optimization problem. The definition of 'separable optimization problem' (Def. 3.1 in (Ygge, 1998)) also covers *discrete* optimization. This suggests that the theorem applies to problems with indivisible goods also. However, we have just provided a counterexample showing that the effectiveness of Pareto-optimal allocations can be very low in an untyped assignment market.

The key to this paradox lies in the nature of the market. The market in the theorem is a market with money (which does not occur in the original optimization problem), and the agents in this market are assumed to have a utility for goods plus money that is quasi-linear in money. This amounts to an agent-level utility of the form

$$\tilde{v}(\beta, m) = v(\beta) + m \tag{3.16}$$

Here, m denotes the amount of money possessed by the agent, β its tool endowment, and v the *value* it attributes to this tool endowment.

It is illuminating to inspect the effect of adding money to our Walrasian trust market (Ex. 3.18 on page 77). Note that we cannot immediately employ (3.16). The sum $v(\beta) + m$ in this equation entails that the agent's utility $v(\beta)$ for the tool bag β is already expressed in terms of the money m. This is not the case in our example, since the utilities in Table 3.1 are relative utilities representing agent satisfaction, rather than absolute utilities representing tool value. The relation between these relative utilities $u_i(x)$ and the underlying absolute utilities $v_i(x)$ is of the form $v_i(x) = f_i \cdot u_i(x) + t_i(x)$, where the factors f_i and the terms t_i are unknown real numbers (cf. Eq. 2.10 on page 39). In other words, we cannot reconstruct v_i from u_i unambiguously. We can, however, choose random values for f_i and t_i to attain absolute utilities which are *consistent* with the matrix U of relative utilities in Table 3.1. For simplicity, we assume that $f_i = 1$ and $t_i = 0$ for all agents i. Thus we

[10] An optimization problem is separable if it can be reformulated as an allocation problem.

arrive at absolute utilities which are the same (in, say, dollars) as the percentages listed in U.

Suppose that we add money to this example market by presenting every agent with \$1000, and specify tool-type prices in terms of dollars. If we start with a unity price of, say, \$100 for every tool type, and apply a standard tâtonnement protocol, this causes the prices of b and e to rise, while the prices of f and c drop. After the first adjustment of the prices, the prices could thus be \$110 for b and e, \$80 for c and f, and \$100 for all other items.

What are the rational bids for the agents in this situation? Agent 2 has two options. It can repeat its bid on e which, when exchanged for b, would incur a utility improvement of \$90 (\$100 minus \$10). Or it could submit a bid on c instead of e. This would incur a utility improvement of \$70 (\$80 minus \$10), *and* a profit due to the price difference of \$30 (\$110 minus \$80). Hence, the total gain obtained with this transaction is \$100, ten dollars more than the gain associated with a bid on e. Apparently, agent 2 receives sufficient monetary compensation to opt for c instead of the inherently more valuable item e. It is left to the reader to check that a similar remark pertains to agent 5, and that the other agents submit the same bids as they did in Fig. 3.12(a).

With these bids, gross supply equals gross demand. In other words, the market reaches a Walrasian equilibrium. More importantly, the associated allocation is optimal in the (relative-)utilitarian sense.

3.4 Chapter Summary and Conclusions

In this chapter, we have explored two strategies to solve TR problems in a distributed fashion, based on proposal composition and proposal relaxation, respectively. Composition entails the *selection* of a subbag of the submitted proposals that is globally feasible, whereas relaxation constitutes an attempt to change the entire proposal profile into one that is globally feasible by *adapting* proposals. As such, the composition strategy is basically a bottom-up strategy, while the relaxation strategy can be qualified as a top-down approach.

In Sect. 3.2.2, we sketched how composition can be combined with proposal revision to arrive at an approach to TR which is informationally decentralized. We then focused on the question how to ensure that the approach is also *operationally* decentralized, that is, how to prevent the central coordination module from becoming a computation bottleneck. In view of the simplicity of proposal revision, this question pertains specifically to the *composition* of proposals.

The analysis in Sect. 3.2.3 has revealed that, if we allow the agents to submit general proposals of the form $\{\alpha \rightleftharpoons \beta\}$ where α and β are tool bags, composition is not tractable. This is not only the case for the rather demanding form of composition formalized in Def. 3.2 of UUCP* (i.e., the search for a *maximal*, globally feasible subbag of proposals), but also for seemingly easier variants such as UUCP$^{\oplus}$ (the search for a single, maximal *cluster* of jointly executable proposals). It is even true for the *least* demanding form of composition one can think of: We proved that UUCP$^{\bullet}$, the decision problem associated with *random* composition (i.e., the search for an *arbitrary*, globally feasible subbag of proposals) is NP-complete. Hence, it seems that, to arrive at a composition/revision approach that is operationally decentralized, we must constrain the kind of proposals that agents can submit.

This led us to *elementary* composition, where agent are only allowed to submit elementary proposals. We proved that ECP*, the elementary version of UUCP*, is tractable, by reformulating it as linear assignment.

As for the question which *variant* of elementary composition (i.e., ECP$^{\bullet}$, ECP$^{\odot}$, ECP$^{\oplus}$, or ECP*) offers the best prospect of attaining a high allocation effectiveness with composition/revision, we argued that ECP$^{\bullet}$ and ECP$^{\odot}$ are unlikely to outperform the other variants if there are considerable differences between the agent-level utilities of first preferences and second-best alternatives. We then employed the graphic representation of composition problems in the form of *tool graphs* to show that, in such cases, ECP* can be superior to ECP$^{\oplus}$ to an arbitrary extent on specific problem instances, and is likely to perform better on average.

We also conceived a top-down approach to ECP*, based on tool-graph analysis. This method, referred to as tool-graph balancing, is efficient for problems with little discrepancy between supply and demand, but computationally complex in the converse case, if the tool-graph contains bottlenecks. However, unlike the solution of ECP* instances via reformulation as assignment problems, tool-graph balancing can be used to identify the proposals for which relaxation is most desirable.

While switching from UUCP* to ECP* takes much of the burden from the mediator's shoulders, it does not simplify the overall reallocation problem. The burden of problem solving is merely shifted from the mediator to the agents. It is an open question how well agents can cope with the added difficulty that they are not allowed to submit composite proposals. We will pay more attention to this issue in subsequent chapters (viz. Chapters 5, 7, and 8), but a complete and final answer is outside the scope of this thesis, because our mechanism-evaluation *experiments* are confined to reassignment. Since reassignment problems involve only one-for-one exchanges, the restriction to elementary proposals does not constitute a constraint here.

As we explained in Sect. 3.1.2, the persuasive style of relaxation employed in CDN is likely to turn the mediator into a bottleneck in terms of computation as well as design. Consequently, we have focused on the *incentive* style of relaxation that is prominent in micro-economics. In particular, we investigated the potential of the Walrasian exchange auction toward TR problems. In Sect. 3.3.4, we identified three minimal demands to be met for the Walrasian auction to constitute an adequate approach.

1. a theorem which proves that — under suitable conditions — a Walrasian equilibrium exists;
2. an efficient auction protocol to attain such an equilibrium;
3. a guarantee that the equilibrium allocations are of acceptable quality in a relative-utilitarian sense.

Our discussion in Sect. 3.3 has revealed that there is little support in economic literature for *any* of these requirements. Theory on equilibrium existence in exchange markets with indivisible goods exists only for an insignificant subclass of TR: the *reassignment* of *untyped* goods.

The TTC auction protocol to attain equilibria in these markets is such that it is questionable whether one can speak of a Walrasian auction: instead of a tâtonnement procedure based on the *scarcities* of goods, the TTC protocol requires the auctioneer to look at the submitted proposals themselves. Furthermore, instead of providing relaxation incentives, the TTC protocol *forces* the agents to revise their bids. As we argued in Sect. 3.3.6, this protocol is, in fact, a composition/revision method with random cluster composition. Consequently, even if we were able to generalize TTC from untyped assignment markets to general TR problems, this would be of little value, since the task of the auctioneer would not be tractable.

Finally, the quality of Walrasian allocations can be expressed in terms of game-theoretic notions such as Pareto optimality and core membership. However, in Sect. 3.3.8, we proved that this does not provide any guarantee with respect to the quality of the allocations in relative-utilitarian terms. By means of an example, we showed that the *effectiveness* of a Walrasian allocation in an assignment market can be *arbitrarily low*, even if the allocation is Pareto optimal. We also showed that the absence of *money* is a crucial factor in this respect. In this context, 'money' should be interpreted as

> a perfectly divisible good, that can compensate every agent for any conceivable loss of goods, and is possessed by every agent in abundance.

Even if all other commodities are indivisible, money (of the above kind) can be used within a Walrasian auction as a vehicle for utility comparison between agents, which enables an agent to convey to some other agent "Hey, I need this more badly than you do" without any explicit or direct communication between the two.

Although we believe to have shown convincingly that there is insufficient *theoretical basis* to regard the Walrasian auction as a promising approach for TR, this does not constitute proof that such an approach is *bound* to be inadequate. The fact that there is no lower bound for the effectiveness of Walrasian allocations is certainly discouraging, but it does not imply that the effectiveness of Walrasian allocations is unacceptably low *on average*. Also, the lack of theoretical results on equilibrium existence can be a matter of the state of the art. For all we know, powerful theorems on equilibrium existence in important subclasses of TR may lie just beyond the horizon.

This has prompted us to take up our own investigation on the existence of Walrasian equilibria (and adequate auction protocols) in TR markets. Unfortunately, the outcome of this investigation, laid down in Chapter 4, confirms the pessimistic tone of the conclusions drawn in the current chapter. Consequently, readers not specifically interested in the Walrasian auction, should feel free to skip Chapter 4, and move on to Chapter 5, where we continue our quest for an adequate general approach to TR in a more constructive manner.

Chapter 4

Walrasian Equilibria in TR markets

4.1 Chapter Overview

This chapter addresses the issue of equilibrium existence in TR. We aim to answer the question whether Walrasian equilibria exist sufficiently often in TR markets for the Walrasian auction to be applicable. In particular, we would like to know whether there exist subclasses of TR, in which *every* problem instance possesses a Walrasian equilibrium. One such subclass is known already. As we mentioned in Sect. 3.3.5, a Walrasian equilibrium always exists in an untyped assignment market (Shapley & Scarf, 1974). However, the problem class of untyped assignment markets constitutes a degenerate case of tool reallocation. Hence, the question is whether there are other, more relevant subclasses of TR, for which the Walrasian auction is an adequate mechanism.

A Venn diagram of classes of reallocation problems is shown in Fig. 4.1. The enveloping class RR is the class of reallocation problems, with either divisible or indivisible goods. The subclass TR of RR comprises the tool reallocation problems, which feature indivisible goods. Within TR, AM denotes the class of assignment markets, tool reallocation problems where no agent possesses or desires more than one tool. The subclass SPAM of AM consists of those assignment markets where all agents have strict preferences.[1] Finally, UAM denotes the class of *untyped* assignment markets, in which the notion of tool *type* is meaningless, because no two tools in the market are of the same type.

The boundary between TR and RR\TR is drawn as a dotted line, to indicate that it is not sharply defined. The problem class RR\TR should be interpreted as the class of reallocation problems with commodity spaces for which the set $\mathbb{R}^k$ is an appropriate

[1] We speak of a strict preference if no agent is ever indifferent between different tool types.

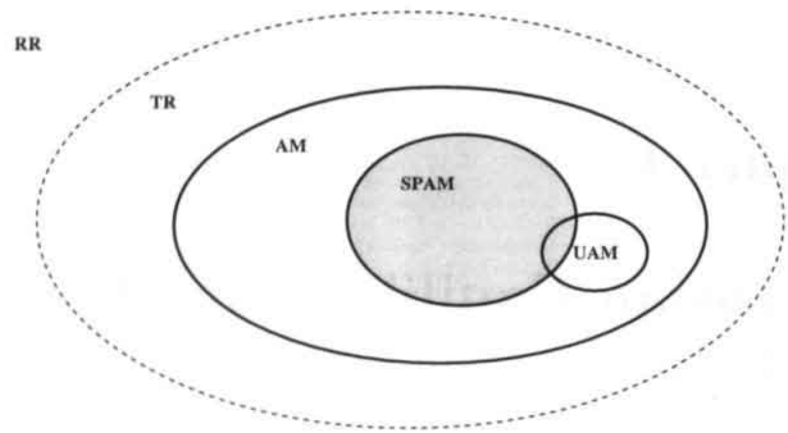


Figure 4.1: Venn diagram of reallocation problem classes:
RR = reallocation of (divisible or indivisible) goods;
TR = reallocation of tools (i.e., indivisible goods);
AM = reassignment of tools;
SPAM = reassignment of tools with strict agent preferences;
UAM = reassignment of untyped tools.

model, and TR as the problem class, for which this is not the case (cf. Sect. 1.1.3). Due to the inherent vagueness of this distinction, it is not feasible to draw hard conclusions about the existence of Walrasian equilibria in TR as a whole.

In view of this, our investigation focuses on the class AM, and more particularly on SPAM. The reason that we look at SPAM in particular is that, even if we could guarantee the existence of an equilibrium in every problem in AM\SPAM, this would be a little value, because agents preferences must be strict for any form of tâtonnement to be effective. This issue will be discussed more elaborately in Sect. 4.7.

The method we use to gain insight in the existence of Walrasian equilibria in SPAM is the following. First, we show that prices are essentially ordinal in a Walrasian reassignment auction. We employ this to arrive at a characterization of Walrasian markets (i.e., markets which possess a Walrasian equilibrium) in SPAM in terms of a relationship between the initial assignment and the preferences of the agents. The concept which embodies this relationship is called market stratifiability. The analysis of this concept leads to the conclusion that, though multiple Walrasian equilibria can exist for a SPAM instance, the associated Walrasian *allocation* is the same for all equilibria. In addition, this theoretical analysis enables us to draw some *negative* conclusions about the existence of Walrasian equilibria in certain subclasses of SPAM.

While our theoretical analysis does not lead directly to conclusions about the frequency of Walrasian equilibria in SPAM as a whole, it does render an efficient tâtonnement algorithm for SPAM. The algorithm is a generalization of the TTC algorithm to *typed* reassignment. It is not only able to find a Walrasian equilibrium if it exists, but also to falsify the existence of Walrasian equilibria. We employ this gTTC algorithm to estimate the *Walrasian density* (i.e., the percentage of Walrasian markets) within SPAM. Rather than doing this once, for SPAM as a whole, we estimate Walrasian densities in various subspaces. The advantage of such a scheme over a single estimation for SPAM is that it provides insight in the factors that influence Walrasian density. In particular, our scheme provides information on the influence of problem scale. In Sect. 4.8, we utilize this information to draw some (tentative) conclusions on the Walrasian density that can be expected in subclasses of TR\AM.

An anecdotal example of success and failure in a Walrasian reassignment auction is presented in Sect. 4.2 below. The example is enlighting, but not required to understand the formal analysis in subsequent sections. Hence, readers who prefer mathematical poetry over prose can skip this section.

4.2 A Walrasian Three-Story Story

The IGI insurance firm is located at the third floor of a former fire station in Paris, Texas. Apart from the director, Mr. Woo, the company has eight employees. Every workday at nine, they assemble in the lobby on the first floor, waiting for Mr. Woo to unlock the elevator door, and escort them to their working place. The elevator is not exactly hi-TEX, but it still works, and the management considers it quite safe as long as it is not used more than once a day. Consequently, at the end of the day, the employees use the fire poles between floors to return to their families.

Mr. Woo does his best to keep the company going, not an easy job in times of management participation. The powerful Insurance Laborers Union in particular, is an indepletable source of insomnia. Only two months ago, they harassed him with this color problem. Hardly a *viable* problem, if you ask Mr. Woo, but nobody did.

The trouble started with one of the firm's lesser valued employees complaining about the dull gray color of his desk. It killed his creativity, and gave the firm a bad image. Bollocks! Yet, it wasn't long before the other employees were affected. Within a week or so, all employees were firmly convinced that this was the prime cause of the turnover falling low. The Union was called in, and, after lengthy negotiations, a Bill of New Desks came out. As a result, the firm was now blessed with a more colorful image, comprising two blue, three pink, two green and one purple desk. Each of the four departments had chosen its own vivid color. The workers were delighted.

Some of the customers seemed less enthusiastic. Many a brow was raised upon glancing at the new furniture. Moreover, the enthusiasm of the employees faded rapidly, up to the point that *all* of the workers were again dissatisfied with the color of their desk.

Yesterday, the dreaded Union knocked on the door again. This time, they demanded freedom of desk choice for each worker. Mr. Woo, soon weary of the negotiation, settled for a half-way compromise. In the morning, each worker was to sit at his/her own desk. In the afternoon, they could all pick a desk of their choice.

"Well, at least it didn't take a whole week again", Henry Woo said to himself as he turned over in bed to switch off the light. Then he froze in his movement. What if more employees prefer the color blue than there are blue desks? What if everybody chooses to sit at the purple desk? They'll chat away for the whole of the afternoon. Work will pile up. Angry customers! Falling turnover! Disaster!

Mr. Woo jumped out of bed, into his slippers, and paced back and forth in the bedroom. He knew that the Union could not be persuaded to review the settlement. But something had to be done to prevent his personnel from jabbering away the day. What? He looked at his wife who was snoring happily, her rollers sparkling softly in the light of the bed-side lamp. No use waking her up. Troubled, he went downstairs and called Mr. Sotheby, his best friend from the good old days at Oxford.

Having heard the story, Mr. Sotheby, who was deeply moved by the problems of his former roommate, hesitantly said that he might know of a solution that would not require a breach of agreement. "I'm not *quite* sure that it will work, though", he said with his whining voice, so typical of Walrasian auctioneers. Mr. Woo didn't care. He gladly welcomed *any* suggestion.

S: Well, I was thinking... At present, the first and second floors are vacant, is it not?

W: Yes...

S: And the elevator cannot be used in the afternoon, can it?

W: No...

S: Then there may be a Top-Trading-Cycles solution.

W: A what?

S: Top-Trading-Cycles. It's an algorithm to find a Walrasian equilibrium in an exchange economy with only one-for-one exchanges.

W: Oh?

S: The idea is that of a Walrasian auction. You try to diminish excess demand for some good by raising its price, and get rid of excess supply by lowering it. Only, you do not really work with prices, more with *levels*.

W: Ah?

S: In your case, lowering prices would come down to relocating some of the desks to the lower storeys. This would prevent the agents at those desks from overcrowding desks at the storeys above them..., since they cannot climb up the fire poles.

W: Yes! Of course! Why didn't I think of that?

S: Easy now, old chap. As I think I mentioned before, I'm not *quite* sure that it will work. Such a Top-Trading-Cycles solution always exists. But, in your case, with eight desks, it *may* require *seven* storeys. And you only have three.

W: What do I need seven stories for? There are only four departments!

S: You mean to say that you do not want to split departments?

W: No, of course not!

S: Aahh.... I see. So where one pink desk goes, all pink desks go. That does complicate matters, though.... It means that we are dealing with a *typed* assignment market. Fortunately, I read something about those only yesterday. In a Ph.D. thesis, mind you. Had a description of an algorithm in it. *Very* simple.

W: So tell me. Which department should go where?

S: I can't tell you that! It depends on your workers' preferences. All I can tell you is what *procedure* you should follow. But it's extremely simple, really. You wait and see what happens at noon, if they all move to their favorite desk. Take notice at which departments there are empty desks, and move all of such departments downward. Then the next day...

W: The next day? How many days will this procedure take?

S: Difficult to say. Probably not too much, since there are only four tool types... colours, I mean. And at least two of these departments, eh... colours, must remain at the third floor to have equilibrium there.

W: Why is that?

S: Didn't you tell me that *all* of your employees are dissatisfied?

W: Yes.

S: Well, then, if we were to leave only one department at the third, all of the – dissatisfied – employees of this department would slide down the pole toward their favorite desk, wouldn't they? And the resulting empty desks at the top floor imply overcrowding – at least at *some* desk – down below.

W: Oh, yeah. Of course.

S: So, where was I. Ah, yes, the second day. Well, you simply keep moving any departments with empty desks to the floor below. Then at some point, you'll either have equilibrium at the third, or no department left there. In the latter case, there's no solution. Otherwise, you perform the same operations again at the floor below.

W: Right.
Still, Henry, I don't really fancy the idea of having to relocate for a week or so. They won't like it one bit. If I relocate once, I could make them believe that it has nothing to do with desk colors and stuff. But they'll surely feel that I try to trick them out of the Union deal if I keep moving desks. So is there not a way to do it all in one move?

S: In one move? ... Well, you could *simulate* the whole process beforehand. But you would need to know all of their preferences, of course.

W: Can you provide the solution if I provide the information?

S: Sure! Just send it to me by email tomorrow, and I'll send the answer back. Mind you, I do need to know *all* of their preferences, not only their first choice, but also their second, third, et cetera. At least up to the colour of their own desk.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

From woo@igi.com Tue Nov 17 16:49:36 1998
To: harold@sotheby.com.uk
Subject: color problem

WORKER       COLOR   PREFERENCE ORDER (decreasing)
Mr. Smith    blue    green, pink, purple, blue
Ms. Brown    pink    green, blue, pink, purple
Mr. Fenn     blue    green, pink, purple, blue
Mrs. Bucket  green   blue, purple, pink, green
Mr. McGull   green   blue, green, pink, purple
Ms. Take     pink    blue, green, pink, purple
Ms. Gross    pink    green, purple, pink, blue
Mr. deVries  purple  pink, blue, green, purple

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
From harold@uk.com.sotheby Tue Nov 17 10:28:04 1998
To: woo@igi.com
Subject: Re: color problem

Congratulations!
Just move the purple and the pink desks to the 2nd, and
... bingo! (Better not receive any customers there :-)
Harold

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


From woo@igi.com Wed Nov 18 21:23:31 1998
To: harold@sotheby.com.uk
Subject: Re: color problem

Thank you very much!
There's a tiny (I hope...) problem, though.
The solution you suggested didn't work. As I looked at the
specification again, I noticed that I had accidentally
exchanged the pink and the purple desk in Ms. Take's preference.
Could you look at it once more? (Sorry for the inconvenience)
Henry

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


From harold@uk.com.sotheby Thu Nov 19 04:29:56 1998
To: woo@igi.com
Subject: Re: color problem

Dear Henry,
That small change makes a big difference. There is no solution.
As I looked into this thesis again, I saw some statistics. It
appears that the chance of equilibrium existence is very dim
anyway. With eight employees and four departments, and a
3-2-2-1 colour distribution (as in this case) the probability
of existence is less than 6%. Walrasian auctions are of little
use for this kind of problem, I guess...
All the best,
Harold
```

4.3 Characterization of Walrasian TR Markets

4.3.1 Preferences and prices

Upon considering the application of a Walrasian auction to tool reallocation, a first observation is the following. If a Walrasian equilibrium exists in some market in which agent i attributes utility $u_i(a)$ to a tool of type a, then that equilibrium continues to exist if u_i is changed to $u_i' = f_o u_i$, where f is a homomorphism, that is, $f(x) > f(y) \Leftrightarrow x > y$. In other words, the existence of Walrasian equilibria depends on the agents' (ordinal) tool-bag *preferences* rather than their (cardinal) tool-bag *utilities*. In this section, we provide some basic definitions related to preferences.

Definition 4.1 (preorder)
A preorder $\succsim$ on a set S is a relation with the following properties.

reflexivity: $(\forall x \in S)\ x \succsim x$

transitivity: $(\forall x, y, z \in S)\ x \succsim y\ \wedge\ y \succsim z \ \ \Rightarrow\ \ x \succsim z$

We use $x \sim y$ as a shorthand for $x \succsim y \wedge x \precsim y$.[2] *Note that $\sim$ defines an equivalence relation on S, that is, a relation which is reflexive, transitive, and symmetric ($x \sim y \ \Leftrightarrow\ y \sim x$). To express that $x \succsim y\ \wedge\ x \not\sim y$, we use $x \succ y$.* △

Within mathematical economics, a preorder is often referred to as a preference ordering (Kelly, 1978). We deviate from this terminology, because we use preorders for tool prices as well as tool preferences. The term 'preference ordering' would be inappropriate and potentially confusing in the context of tool pricing.

Definition 4.2 (order, weak order, anti-symmetric)
A preorder $\succsim$ on S is an order if it is anti-symmetric, that is, iff

$$(\forall x, y \in S)\ x \sim y\ \Leftrightarrow\ x = y$$

Usually, we will denote an order relation by $\succ$ or $\succeq$ instead of $\succsim$. To state that some preorder is definitely not an order, we use the term weak order. △

Relationships between a price preorder and a preference preorder are key elements in the theory developed in this section. Consequently, we need two different symbols for the two preorders. We choose to use $\succsim$ for preference preorders and $\gtrsim$ for price preorders. If we use $a > b$ to express that tool type a is in a higher price class than tool type b, this may tempt the reader to assume that the price preorder is an order. We therefore use $a \gnsim b$ instead of $a > b$ to express that $a \gtrsim b \wedge a \nsim b$, if the price preorder is not known to be an order.

In the following, a *strict* preference is a preference that amounts to an order, and a weak preference amounts to a weak order.

[2]Obviously, $x \precsim y\ \Leftrightarrow\ y \succsim x$.

Definition 4.3 (total preorder, total order)
An preorder $\succsim$ on S is total iff

$$(\forall x, y \in S)\ x \succsim y\ \vee\ y \succsim x$$

△

Throughout the thesis, we will assume that preferences and price orders are *total* preorders. We denote the set of total preorders over the set R by PREORD(R), and the set of total preorders over some finite set of size m by PREORD[m].[3] The subspace of total orders over a set of size m is isomorphic with the space of permutations over m elements, and will therefore be denoted by PERM[m].

Any equivalence relation $\sim$ on a set S defines a partition $S/_\sim$ of S into equivalence classes. Since any total preorder $\succsim$ *incorporates* an equivalence relation $\sim$ (defined by $x \sim y \Leftrightarrow (x \succsim y \wedge y \succsim x)$), the total preorder $\succsim$ also defines a partition $S/_\sim$. Moreover, the preorder $\succsim$ induces a total order on the partition $S/_\sim$. We therefore refer to $S/_\succsim$ as the *ordered partition* induced on S by $\succsim$.

Definition 4.4 (maximal elements)
Let $\succsim$ be an preorder on S. Then the set of maximal elements of $T \subset S$ under $\succsim$ is defined as

$$\text{best}_\succsim(T) = \{x \in T \mid (\neg\exists y \in T)\ y \succ x\}$$

For a total order $\succeq$ on S, $\text{best}_\succeq(T)$ contains exactly one element for any $\emptyset \neq T \subset S$. This unique maximal element is denoted by $\max_\succeq(T)$. △

Up to this point, we have provided general definitions of concepts related to orderings, which are applicable to reallocation problems in general. The definitions that follow presume that the context is reassignment.[4]

Definition 4.5 (preference profile)
Let I be a set of agents, and R a set of tool types. A preference profile P of I over R is a set $\{\succsim_i\}_{i \in I}$ of total preorders over R. For assertions involving $\succsim_i$, we employ the following notation.

$a \succ_5 b$	*"Agent 5 prefers tool type a over b."*
$a \sim_5 b$	*"Agent 5 is indifferent between tool types a and b."*
$a \succsim_5 b$	*"Agent 5 either prefers a over b, or is indifferent."*

The shorthand notation [a.b.c, bc.a] denotes the preference profile of $\{1,2\}$ over $\{a,b,c\}$, defined by $a \succ_1 b \succ_1 c \ \wedge\ b \sim_2 c \succ_2 a$. △

[3] PREORD[m] can be defined formally as a normal divisor of the equivalence relation "of the same size" on the set of preorders over all sets of m elements.

[4] The defined concepts can be generalized to reallocation, but this is not always true for the notation.

In our analysis of assignment markets, we will often *restrict* preference profiles and assignments to a subset $R' \subset R$ of tool types, or a subset $I' \subset I$ of agents.

An initial assignment $e : I \to R$ can be restricted to pertain only to the agents in a subset $I' \subset I$, (direct domain restriction), or only to the agents endowed with a tool type in $R' \subset R$ (indirect domain restriction). We use the symbol $\restriction$ for such domain restrictions. The formal definition below is based on the fact that a function $f : X \to Y$ is defined in mathematics as a subset of $X \times Y$. This implies that the statement $e(i) = r$ is synonymous with $\langle i, r \rangle \in e$.

Definition 4.6 (assignment restriction)
Let $e : I \to R$ denote an assignment of tools in R to the agents in I. Let $R' \subset R$, and $I' \triangleq \{i \in I \mid e(i) \in R'\}$.
Then the restrictions of e to I', and to R' are defined as

$$e\restriction_{I'} \triangleq e \cap (I' \times R)$$

$$e\restriction_{R'} \triangleq e \cap (I' \times R')$$

Evidently, grace to the definition of I', these two restrictions are alternative definitions of the same object. △

To arrive at a formal definition of profile restriction, we observe that a preference preorder on a set R is a relation on R, which — like a function from R to itself — is a subset of $R \times R$. Hence, a preference *profile* for the agent community I and the set R of tool types can be viewed as a subset of $I \times R \times R$. In other words, if the profile P involves the preorders $\succsim_i$, then $a \succsim_i b$ can also be stated as $\langle i, a, b \rangle \in P$. Hence, we can define profile restriction similarly to assignment restriction in Def. 4.6.

Definition 4.7 (profile restriction)
Let $e : I \to R$ denote an assignment of tools in R to the agents in I, and P a profile of preferences over R of the agents in I. Let $R' \subset R$, and define $I' \triangleq \{i \in I \mid e(i) \in R'\}$.
Then the restrictions of P to I', and to R' are defined as follows.

$$P\restriction_{I'} \triangleq P \cap (I' \times R \times R)$$

$$P\restriction_{R'} \triangleq P \cap (I' \times R' \times R')$$

△

Example 4.8 (restriction of a preference profile)
Let $R' = \{a, c\} \subset \{a, b, c, d\} = R$, $I' = \{1, 2, 3\} \subset \{1, 2, 3, 4, 5\} = I$, and $P = [abc.d,\ a.b.dc,\ bd.c.a,\ c.d.ab,\ cd.ab]$.
Then $P\restriction_{I'} = [abc.d,\ a.b.dc,\ bd.c.a]$ and $P\restriction_{R'} = [ac,\ a.c,\ c.a]$. △

4.3.2 Equilibria in assignment markets

In this section, we develop formal definitions for assignment markets and Walrasian equilibria in such markets. On the basis of these definitions we will try to acquire insight in the likelihood of existence of Walrasian equilibria in AM.

To promote this, we strive for a characterization of markets with a Walrasian equilibrium that is *structural* rather than *behavioral*. Conditions for equilibrium existence formulated in terms of behavioral constructs like the excess demand function[5] may be useful if one seeks for suitable model assumptions, but it is difficult to verify the satisfaction of behavioral conditions for specific problem instances, let alone to employ them adequately to assess the likelihood of equilibrium existence in subclasses of TR. In this respect, structural conditions, that is, conditions on constructs in the problem specification, such as the preference profile and the initial allocation, are more valuable.

As a first step toward a structural characterization of Walrasian markets in AM, we provide general definitions for markets and Walrasian equilibria in TR, based on the earlier definitions for TR problems (Def. 2.36 on page 42) and Walrasian equilibria (Def. 3.12 on page 69).

Definition 4.9 (TR market)
A TR market M is a tuple $\langle I, R, \delta, P\rangle$, where

- $I = \{1, \ldots, n\}$ *is a set of agents,*
- $R = \{r_1, \ldots, r_m\}$ *is a set of tool types,*
- $\delta : I \rightarrow \mathfrak{B}(R)$ *is an initial allocation, and*
- $P \in \text{PREORD}(\mathcal{P}(\underline{\downarrow}\, \mathit{Im}\, \delta))^n$ *is a profile of preferences over the set of subbags of* $\underline{\downarrow}\, \mathit{Im}\, \delta$*, the community tool bag.*

△

Def. 4.9 of TR markets differs only marginally from Def. 2.36 of TR problems. The utility profiles are replaced with preference profiles. Furthermore, while a TR problem instance is denoted by a pair $\langle \delta, \bar{U}\rangle$ of an initial allocation and a utility profile, a TR market is denoted as a 4-tuple $\langle I, R, \delta, P\rangle$. The latter difference is a matter of convenience. In our analysis, we often look at portions of a market that involve only agents from a subset $I' \subset I$ or only tool types from a subset $R' \subset R$. Such market portions are easier to denote if I and R are part of the denotation.

The following definition of Walrasian TR equilibrium reformulates Def. 3.12 in terms of bags. Apart from that, the only difference between the two definitions is that

[5] An example of such a condition is that of gross substitutability, which plays an prominent role in economic theory on markets with divisible goods.

the prices in the definition of Walrasian TR equilibria are rational instead of real numbers.

Definition 4.10 (Walrasian equilibrium/allocation/price assignment)
Let $\langle I, R, \delta, \{\succsim_i\}_{i \in I}\rangle$ be a TR market, and let $p : R \to \mathbb{Q}_+$ denote a price assignment to tool types. Define the budget set of agent i under price assignment p as

$$B_i(p) = \{\beta \in \mathfrak{B}(R) \mid \sum_{x \in \beta} p(x) \leq \sum_{x \in \delta(i)} p(x)\} \tag{4.1}$$

Then the pair $\langle \omega, p\rangle$ with $\omega : I \to \mathfrak{B}(R)$ and $p : R \to \mathbb{Q}_+$ is a Walrasian equilibrium iff

(i) $(\forall i \in I) \;\; \omega(i) \in B_i(p)$
(All allocated tool bags are affordable)

(ii) $(\forall i \in I)(\forall \beta \in \mathfrak{B}(R)) \;\; \beta \succ \omega(i) \;\Rightarrow\; \beta \notin B_i(p)$
(Any bag preferred over an allocated one is not affordable.)

(iii) $\underline{\downarrow} \mathit{Im}\, \omega = \underline{\downarrow} \mathit{Im}\, \delta$
(Conservation of tools.)

If $\langle \omega, p\rangle$ is a Walrasian equilibrium, ω is called a Walrasian allocation, and p a Walrasian price assignment. △

In an assignment market, each agent possesses exactly one tool of a certain type, and may be interested in exchanging this tool for a tool of another type. Consequently, in the context of AM, Def. 4.9 of TR markets can be simplified to the definition below.

Definition 4.11 (assignment market)
An assignment market is a tuple $\langle I, R, e, P\rangle$, where
$I = \{1, \ldots, n\}$ is a set of agents,
$R = \{r_1, \ldots, r_m\}$ is a set of tool types,
$e : I \to R$ is an initial allocation, and
$P \in (\text{PREORD}[m])^n$ is a profile of preferences over R. △

Example 4.12 (Assignment market representation)
To describe a specific AM instance $\langle I, R, e, P\rangle$, it suffices to specify the initial assignment e and the preference profile P. We represent (initial or final) assignments as character strings enclosed in brackets. Thus, [abbc] represents the assignment of a type-a tool to agent 1, a type-b tool to agents 2 and 3, and a type-c tool to agent 4. An example market $\langle I, R, e, P\rangle$ is represented by underlining the endowment of each agent in the representation of its preference. Thus, the market $\langle I, R, e, P\rangle$ with $P =$ [a.bc, b.a.c, c.a.b, b.ca] and $e =$ [abbc] is denoted by [$\underline{a}$.bc, $\underline{b}$.a.c, c.a.$\underline{b}$, b.$\underline{c}$a]. △

For assignment markets, Def. 4.10 (of Walrasian TR equilibria) can be simplified considerably. To begin with, the budget equation (4.1) simplifies to

$$B_i(p) = \{x \in R \mid p(x) \leq p(e(i))\} \tag{4.2}$$

This reflects that, as we move from TR to AM, it is no longer relevant for an agent to what *extent* the price of tool type a exceeds that of tool type b, but only *whether* $p(a) > p(b)$. In other words, in the context of AM, a price assignment is essentially ordinal, and can therefore be defined as a *preorder* over R. A formal proof for this statement is provided in Def. 4.13 and Prop. 4.14 below.

Definition 4.13 (ordinal equivalence)
Two price assignments p and q are ordinally equivalent ($p \backsim q$) if the ordinal relationships between prices of any two tool types are the same with both assignments, that is, iff

$$(\forall r \in R)\ \ p(i) < p(j) \Leftrightarrow q(i) < q(j)$$

△

Proposition 4.14
Let $M = \langle I, R, e, P\rangle$ be an assignment market with a Walrasian equilibrium $\langle w, p\rangle$, and let q be a price assignment such that $q \backsim p$. Then $\langle w, q\rangle$ is also a Walrasian equilibrium of M.

Proof.
Def. 4.13 implies that the budget set, redefined by (4.2)(!), does not change if we replace p by q, if $q \backsim p$. Hence, conditions (i) and (ii) remain valid if we replace p by q. Finally, condition (iii) in Def. 4.10 is not affected since it does not involve p.

■

The following proposition reveals that the equation defining the budget set can be simplified even more, to an extent that obviates its definition altogether.

Proposition 4.15
If $\langle w, p\rangle$ is a Walrasian TR equilibrium in an assignment market $M = \langle I, R, e, P\rangle$, then

$$(\forall i \in I)\ \ p(w(i)) = p(e(i))$$

Proof.
Let, for any $I' \subseteq I$, $e(I')$ denote the bag of tools of the subcommunity I', that is, $e(I') \triangleq \biguplus_{i \in I'}\{e(i)\}$. Define $w(I')$ similarly. Extend $p(\cdot)$ to a price assignment to tool *bags*, in the usual manner: $(\forall \beta \in \mathfrak{B}(R))\ p(\beta) \triangleq \sum_{x \in \beta} p(x)$.
From condition (i) in Def. 4.10, and the redefinition of the budget set in (4.2), we know that

$$(\forall i \in I)\ \ p(w(i)) \leq p(e(i)) \tag{4.3}$$

Hence,

$$(\forall I' \subset I)\ p(w(I')) \leq p(e(I')) \tag{4.4}$$

Also, by condition (iii) of Def. 4.10, $w(I) = \underline{\downarrow} Im\, w = \underline{\downarrow} Im\, e = e(I)$, and hence,

$$p(w(I)) = p(e(I)) \tag{4.5}$$

Choose $i_0 \in I$ arbitrarily, and define $I_0 \triangleq I \backslash \{i_0\}$. Then by (4.4) and (4.5),

$$p(w(i_0)) = p(w(I)) - p(w(I_0)) \geq p(e(I)) - p(e(I_0)) = p(e(i_0)) \tag{4.6}$$

At the same time, (4.3) tells us that $p(w(i_0)) \leq p(e(i_0))$. It follows that $p(w(i_0)) = p(e(i_0))$. ∎

As a consequence of Prop. 4.15, Def. 4.10 of Walrasian TR equilibria can be reformulated for assignment markets as follows.

Definition 4.16 (Walrasian AM equilibrium)
Let $\langle I, R, e, \{\succsim_i\}_{i\in I}\rangle$ be an assignment market. Then the pair $\langle w, \gtrsim\rangle$ with $w : I \to R$ and $\gtrsim$ a price preorder on R, is a Walrasian equilibrium iff the following conditions are met.

(i) $(\forall i \in I)\ \ w(i) \sim e(i)$

(ii) $(\forall i \in I)(\forall r \in R)\ \ r \succ_i w(i) \ \Rightarrow\ r \gtrsim e(i)$

(iii) $\underline{\downarrow} Im\, w = \underline{\downarrow} Im\, e$

△

With Def. 4.16, we have attained a characterization of Walrasian assignment markets that is substantially simpler than the corresponding definition (Def. 4.10) for TR markets. However, the characterization is not a structural one, since it involves existential quantification over the unknown variable w. The notion of market stratifiability, which we will develop shortly, gets rid of this unknown variable by reformulating conditions (i) and (ii) in Def. 4.16 in terms of the *structural* notions of equilibrial market, market segment, and submarket.

Definition 4.17 (equilibrial market)
A market is equilibrial if it possesses a Walrasian equilibrium with one and the same price for all goods. △

The above definition of equilibrial market is general, but not structural: implicitly, it involves the same existential quantification over w as Def. 4.16. However, as Prop. 4.18 below states, equilibriality can be reformulated as a structural property in the case of assignment markets in which all agent preferences are strict.

Proposition 4.18 (equilibrial assignment markets)
Let $M = \langle I, R, e, P \rangle$ be an assignment market with a preference profile P of strict preferences $\succ_i$ on R. Then M is equilibrial iff the bag of first preferences $\{\max_{\succ_i}(R) \mid i \in I\}$ equals the community tool bag $\underline{\downarrow} \mathit{Im}\, e$.

Proof.
$\Rightarrow$:
An agent in a Walrasian auction always bids on an endowment that is a maximal element (with respect to its preference) within its budget set, and the budget set in an assignment problem equals R if all tool prices are the same. Furthermore, if agent i has a strict preference, there is a unique maximal element in the budget set for every price preorder. Hence, in this case, the bid $b(i)$ submitted by agent i is $b(i) \triangleq \max_{\succ_i}(R)$. Because the market is equilibrial, while the agents' bids under any price preorder are defined unambiguously, the bids $b(i)$ constitute, together with the price preorder $\sim$ (of equal prices for all tool types) a Walrasian equilibrium $\langle b, \sim \rangle$. Hence, by condition (iii) of Def. 4.16, $\underline{\downarrow} \mathit{Im}\, b = \underline{\downarrow} \mathit{Im}\, e$. In other words, the bag of first preferences equals the community tool bag.

$\Leftarrow$:
The proof for the converse implication is analogous, and left to the reader.

■

A market is always *closed*, that is, agent preferences only involve tool types which are present within the community. To define stratifiability, we need to speak about parts of a market which do not necessarily have this property. We refer to such parts of a market as *market segments*. A market segment comprises a nonempty *subset* of the tools types that occur in the enveloping market, and *all* agents that (initially) possess one of these tool types. Since the agents in a market segment still have their original preferences, a market segment is generally not closed in the above sense.

Definition 4.19 (market segment)
Let $M = \langle I, R, e, P \rangle$ be an assignment market.
Then $M' = \langle I', R', e', P' \rangle$ is a market segment of M iff

1. $R' \subseteq R \wedge R' \neq \emptyset$
2. $I' = \{i \in I \mid e(i) \in R'\}$
3. $e' = e{\restriction}_{I'}$
4. $P' = P{\restriction}_{I'}$

△

By viewing conditions 2, 3, and 4 in Def. 4.19 as definitions of I', e', and P', we can associate any nonempty subset R' with a market segment. We denote the market segment induced by R' in this manner by $M\restriction_{R'}$. Note that the correspondence between market segments of M and subsets of R is one-to-one.

The closure of a market segment M' of M is referred to as the *submarket* associated with M'.

Definition 4.20 (submarket)
The submarket associated with a market segment $M' = \langle I', R', e', P' \rangle$ of M is the market

$$\lceil M' \rceil = \langle I', R', e', P'\restriction_{R'} \rangle$$

$\triangle$

The intersection and union of market segments (and submarkets) are defined as follows.

Definition 4.21 (intersection of submarkets/market segments)
Let $M = \langle I, R, e, P \rangle$ be an assignment market, and let $M^1 = \langle I^1, R^1, e^1, P^1 \rangle$ and $M^2 = \langle I^2, R^2, e^2, P^2 \rangle$ be market segments of submarkets of M. Let $I_\cap = I^1 \cap I^2$. Then the intersection of M^1 and M^2 is defined as

$$M^1 \cap M^2 = \langle I_\cap, R^1 \cap R^2, e\restriction_{I_\cap}, P\restriction_{I_\cap} \rangle$$

$\triangle$

Definition 4.22 (union of submarkets/market segments)
Let $M = \langle I, R, e, P \rangle$ be an assignment market, and let $M^1 = \langle I^1, R^1, e^1, P^1 \rangle$ and $M^2 = \langle I^2, R^2, e^2, P^2 \rangle$ be market segments of submarkets of M. Let $I_\cup = I^1 \cup I^2$. Then the union of M^1 and M^2 is defined as

$$M^1 \cup M^2 = \langle I_\cup, R^1 \cup R^2, e\restriction_{I_\cup}, P\restriction_{I_\cup} \rangle$$

$\triangle$

Note that the union of the market segments $M\restriction_{R^1}$ and $M\restriction_{R^2}$ is the market segment $M\restriction_{R^1 \cup R^2}$. In view of the one-to-one correspondence between subsets of R and the market segments induced by these subsets, this implies that the set of submarkets of any market M is closed under segment union. The same applies to segment intersection, provided that the two subsets of R are not disjoint.

The same correspondence between subsets of R and market segments of $M = \langle I, R, e, P \rangle$ allows us to extend the ordered partition $R/_{\gtrsim}$ induced by a preorder $\gtrsim$ on R to an ordered partition of M into market segments. We refer to such a partition of M as a *segmentation* of M.

Definition 4.23 (market segmentation, head, tail)
A segmentation $S_M = \langle M^1, \ldots, M^k \rangle$ of M is a list (i.e., an ordered tuple) of disjoint market segments M^i of submarkets of M, such that $\cup_{i=1,\ldots,k} M^i = M$.
Segments are specified in descending order. In other words, M^1 is the maximal element in the segmentation $\langle M^1, \ldots, M^k \rangle$. In accordance with the usual definitions of the head and the tail of a list, we define the head of S_M as M^1, and the tail of S_M as $\langle M^2, \ldots, M^k \rangle$. They are denoted by $hd\, S_M$ and $tl\, S_M$, respectively. Expressions like $hd\, tl^3\, S_M$ should be interpreted as $hd\, tl\, tl\, tl\, S_M = hd\langle M^4, \ldots, M^k \rangle = M^4$. △

The following definition of *stratifiable market* is the key definition of this chapter.

Definition 4.24 (stratifiable market, stratification, top stratum)
An assignment market $M = \langle I, R, e, P \rangle$ is stratifiable if there exists a price preorder $\gtrsim$ on the set R of tool types, such that the segmentation $M_{\gtrsim} = \langle M^1, \ldots, M^m \rangle$ meets the following constraints.

1. *The submarkets $\lceil M^i \rceil$ associated with the market segments of $M_{\gtrsim}$ are equilibrial markets.*
2. *If an agent prefers a tool type over all tool types within its own segment, then this favorite type resides within a higher segment:*

$$(\forall i \in I)(\forall y \in R)\ ((\forall x \sim e(i))\, y \succ_i x) \Rightarrow y \gtrapprox e(i)$$

A segmentation $M_{\gtrsim} = \langle M^1, \ldots, M^m \rangle$ which satisfies conditions 1 and 2 above is called a stratification of M, and the segments of $M_{\gtrsim}$ are the strata of M. M^1 is called the top stratum of M. △

Proposition 4.25
Let M denote an assignment market.
Then M is Walrasian $\Leftrightarrow$ M is stratifiable.

Proof.
$\Rightarrow$:
Suppose that M possesses a Walrasian equilibrium $\langle w, \gtrsim \rangle$. We prove that $\langle M^1, \ldots, M^k \rangle \triangleq M_{\gtrsim}$ is a stratification of M, that is

(1) Any submarket $\lceil M^j \rceil$ is an equilibrial market (for $j = 1, \ldots, k$)

(2) $(\forall i \in I)(\forall y \in R)\ ((\forall x \sim e(i))\, y \succ_i x) \Rightarrow y \gtrapprox e(i)$

Because the stratum M^j comprises all tools of the price-equivalence class $R^j \in R/\gtrsim$, it follows from condition (i) in Def. 4.16 that, with the budget constraint in effect, all trade within M^j concerns tools from R^j. Since this is the case for *all* strata of $M_{\gtrsim}$, condition (iii) in Def. 4.16 implies that there is conservation of tools *within each stratum*. Since the fact that the submarket $\lceil M^j \rceil$ is closed implies that the budget constraint is in effect, $\lceil M^j \rceil$ is an equilibrial submarket. Hence, condition (1) is met. Condition (2) follows immediately from conditions (i) and (ii) in def. 4.16.

$\Leftarrow$:
This is proven analogously to the converse implication, so we do not go into detail. Condition (i) in def. 4.16 follows from conditions (1) and (2) above. In turn, this implies, together with conditions (1) and (2) that condition (ii) in Def. 4.16 is met also. Finally, condition (iii) is a direct consequence of condition (1).

■

We have shown that market stratifiability is a necessary and sufficient condition for Walrasian equilibria to exist in an assignment market M. Stratifiability is a structural notion, provided that the preferences of all agents are strict (cf. Prop. 4.18). In view of this, we focus on assignment markets with strict preference profiles in the analysis that follows.

4.3.3 Equilibria in strict-preference assignment markets

Definition 4.26 (SPAM)
A Strict-Preference Assignment Market (or SPAM) is an assignment market in which the preferences of all agents are strict, that is $P = \{\succ_i\}_{i \in I} \in (\text{PERM}[m])^n$. *The problem space of all SPAMs is denoted by* SPAM. △

If it is clear that we are dealing with SPAMs, dots in preference specifications (cf. Ex. 4.12) are superfluous. Hence, in descriptions of SPAM instances, we will omit the dots from the preference profile. The preference $a \succ b \succ c \succ d \succ e$ will, for example, be denoted by *abcde* instead of *a.b.c.d.e* if it occurs within a SPAM instance.

The main advantage of assuming that preferences are strict lies in the fact that, with this assumption, the bid of an agent in a Walrasian auction is determined unambiguously. As a consequence, any stratification of a SPAM is associated with a *single* Walrasian allocation.

Proposition 4.27
Let $M = \langle I, R, e, P \rangle$ *be a stratifiable SPAM with stratification* $M_{\gtrsim}$, *and let* $\langle w, \gtrsim \rangle$ *be a Walrasian equilibrium for* M. *Then* w *is unambiguously defined by*

$$(\forall i \in I)\, w(i) = \max_{\succ_i}(\{r \in R \mid r \sim e(i)\})$$

Proof.
By Prop. 4.15, every tool exchange takes place *within* a stratum, that is, $w(i) \in \{r \in R \mid r \sim e(i)\}$. Furthermore, in a SPAM, the preference of each agent is strict, so for any set $R' \subseteq R$, each agent's first preference $\max_{\succ_i}(R')$ unambiguously

determines a tool type. Thus, conditions (i) and (ii) in Def. 4.16 can be combined into $w(i) = \max_{\succ_i}(\{x \in R \mid x \sim e(i)\})$, which determines w unambiguously.

■

The above proposition shows that each stratification of a SPAM M is associated with one and only one Walrasian allocation. To answer the question *how many* Walrasian allocations exist, it would be most convenient if we were able to show that even *different* stratifications of a SPAM always correspond with one and the same Walrasian allocation. The route that we will take to prove this is the following.

1. We define a canonical stratification, and show that, for any stratifiable SPAM, a unique canonical stratification exists.
2. We specify a set of transformation rules that, when applied to an arbitrary stratification of a stratifiable SPAM, will render the associated canonical stratification.
3. We prove that the final (Walrasian) allocation associated with a stratification is invariant under these transformation rules.

As a first step toward the definition of the canonical stratification, we visualize a segmentation as a graph in such a way that we can determine whether the segmentation is a stratification by merely inspecting the graph. The visualization is arrived at by defining a relation between the segments of a segmentation in terms of the (first) preferences of the agents in those segments.

Definition 4.28 (envy between market segments)
Let $\langle M^1, \ldots, M^k \rangle$ be a segmentation of an assignment market M. Then segment M^i envies segment M^j if there is an agent in M^i who prefers some tool type in segment M^j over all tool types in its own segment. Formally,

$$M^i \lessdot M^j \quad \Leftrightarrow \quad i \neq j \wedge (\exists a \in I^i)(\exists r^* \in R^j)(\forall r \in R^i)\ r^* \succ_a r$$

△

While the relation induced between market segments by a price preorder is an order, that is, a relation which is reflexive, transitive and ant-symmetric, the envy relation has none of these three properties. However, $\overline{\lessdot}$ is anti-symmetric (and transitive, of course), if the segments are strata of a stratification (by condition (ii) in Def. 4.24). Because of this property, we can use $\overline{\lessdot}$ to rephrase condition 2 in Def. 4.24 as "$\overline{\lessdot}$ is an anti-symmetric relation". This implies that, if we picture $\lessdot$ graphically (by representing $M^i \lessdot M^j$ as an arc from node M^i to node M^j), a market stratification is recognizable as a *cycle-free* digraph.

This is illustrated in Fig. 4.2 for the assignment market specified in Fig. 4.2a. It is left to the reader to deduce from the problem description in Fig. 4.2a that the segmentation pictured in Fig. 4.2b is a stratification. Once we have established this, we can deduce from the *graph* in Fig. 4.2b that the segmentation shown in Fig. 4.2c is a stratification while that in Fig. 4.2d is not.

This is so, for the following reasons. The two bottom nodes (d and e) in Fig. 4.2b can be joined into one segment without introducing cycles in the digraph, grace to the fact they are $\bar{\lessdot}$-unrelated. If we join segments d and e, the graph in Fig. 4.2b transforms into the graph of Fig. 4.2c.

Nodes e and a are $\bar{\lessdot}$-related via node $\{b, c\}$. Consequently, if we join these nodes to arrive at Fig. 4.2d, we introduce a cycle ($\{a, e\} \bar{\lessdot} \{b, c\}$ and $\{b, c\} \bar{\lessdot} \{a, e\}$) in the graph. Hence, the segmentation in Fig. 4.2d is not a stratification.

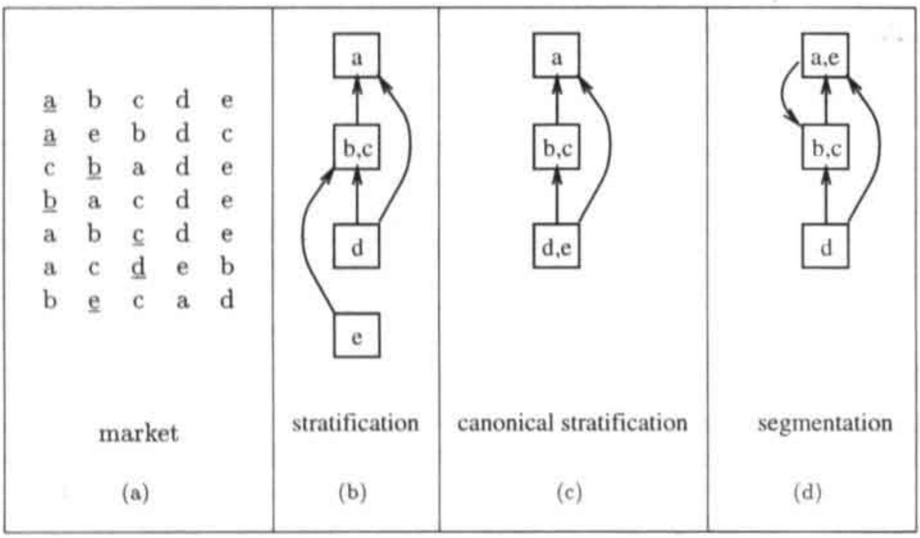


Figure 4.2: A Walrasian market with two of its stratifications, and a segmentation which is not a stratification.

If we define the *weight* of a stratification $\langle M^1, \ldots, M^k \rangle$ by $\sum_i \frac{|R^i|}{i}$, the stratification in Fig. 4.2c is heavier than the stratification in Fig. 4.2b, since term $\frac{|R^3|}{3} = \frac{2}{3}$ in the weight formula of the former stratification exceeds the sum $\frac{|R^3|}{3} + \frac{|R^4|}{4} = \frac{1}{3} + \frac{1}{4}$ in the formula associated with the latter stratification. This illustrates that we can turn a stratification into a heavier one by moving one or more tool types to a higher stratum. In fact, the stratification in Fig. 4.2c is is the heaviest stratification that exists for this problem instance. We refer to this stratification as the *canonical* stratification..

Below, we present a formal definition of the canonical stratification of a SPAM. The definition is constructive and recursive, defining the head (i.e., the top stratum) of the stratification, together with the market from which the tail can be computed. The definition uses the notions of *reopening* and *equilibrial market segment*. Reopening a submarket amounts to supplanting its preference profile with the original profile (i.e., that of the enveloping market). An equilibrial market segment is a segment with an *equilibrial* associated submarket, which is effectively equal to the segment itself, in the sense that all of the agents' *first* preferences lie within the segment. Consequently, none of these first preferences are affected when the preference profile P' of the segment is restricted to the set R' of tool types in the segment.

Definition 4.29 (Equilibrial Market Segment (EMS))
Let $M' = \langle I', R', e', P' \rangle$ be a market segment of a SPAM $M = \langle I, R, e, P \rangle$, where $P = \{\succ_i\}_{i \in I}$. Then M' is an equilibrial market segment (EMS) within M iff

(i) The submarket $\lceil M' \rceil$ associated with M' is an equilibrial market

(ii) $(\forall i \in I')\ \max_{\succ_i}(R) \in R'$

The set of EMSs in a SPAM M is denoted by $\text{EMS}(M)$. △

Example 4.30 (EMS)
The segment $M' = [c\underline{a}db,\ ab\underline{c}d]$ is an EMS of the SPAM $M = [a\underline{b}cd,\ c\underline{b}ad,\ c\underline{a}db,\ ab\underline{c}d,\ bca\underline{d}]$, because

(i) $\lceil M' \rceil = [c\underline{a},\ a\underline{c}]$ is an equilibrial submarket of M, and

(ii) the first preferences of both of the agents in M' are in R':

- $\max_{\succ_3}(\{a, b, c, d\}) = c \in \{a, c\}$
- $\max_{\succ_4}(\{a, b, c, d\}) = a \in \{a, c\}$

△

Reopening a submarket amounts to supplanting its preference profile with the original profile (i.e., that of the enveloping market).

Definition 4.31 (reopening)
Let $M = \langle I, R, e, P \rangle \in$ SPAM, and let $M' = \langle I', R', e', P' \rangle$ be a submarket of M. Then the reopening $\mathfrak{R}_M(M')$ of M' to M equals

$$\mathfrak{R}_M(M') = \langle I', R', e', P \rangle$$

The reopening of a segmentation is defined componentwise, that is,

$$\mathfrak{R}_M(\langle M^1, \ldots, M^k \rangle) \triangleq \langle \mathfrak{R}_M(M^1), \ldots, \mathfrak{R}_M(M^k) \rangle$$

△

Definition 4.32 (canonical stratification)
Let $M = \langle I, R, e, P \rangle$ be a stratifiable SPAM, and let E denote the union of all equilibrial market segments of M. Then the head $M^1 = \langle I^1, R^1, e^1, P^1 \rangle$ of the canonical stratification C_M equals E, and the tail of C_M equals the reopening to M of the canonical stratification of the market $\lceil M \setminus M^1 \rceil$. △

Since, by Def. 4.29, the head M^1 of any stratification $M_{\gtrsim}$ of M is a (nonempty) EMS of M, Def. 4.32 is well-posed. The canonical stratification, thus defined, is obviously unique for any SPAM. From Prop. 4.27, we know that it is associated with a unique Walrasian allocation. In the following, we show that *all* of the stratifications of the market lead to the same Walrasian allocation.

As a first step, we describe the structure of a canonical stratification in more detail, discerning one or more *minimal* equilibrial market segments within each EMS.

Definition 4.33 (Minimal EMS (MEMS))
Let M be an assignment market. Then M' is a minimal equilibrial market segment (MEMS) of M iff

1. $M' \in$EMS(M), *and*
2. $\neg(\exists M'' \in$ EMS$(M))\ M'' \subsetneq M'$

The set of MEMSs in a SPAM M is denoted by MEMS(M). △

Lemma 4.34
Let $M^1, M^2 \in$MEMS(M) with $M^1 \neq M^2$. Then $M^1 \cap M^2 = \emptyset$.

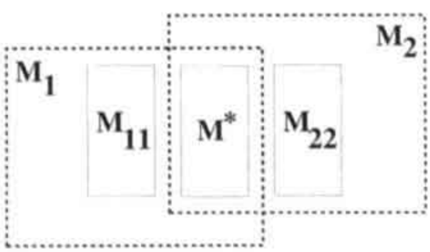


Figure 4.3: Two MEMSs M^1 and M^2 of a SPAM M are always disjoint.

Proof.
Let M^1 and M^2 be different MEMSs of the SPAM M. Define $M^* = M^1 \cap M^2$, $M_{11} = M^1 \setminus M^*$, and $M_{22} = M^2 \setminus M^*$ (cf. Fig. 4.3). Since M^1 and M^2 are MEMSs, neither M_{11} nor M_{22} is empty. If, for example, M_{11} were empty, then M^1 would be an equilibrial market segment within M^2, which contradicts the assumption that $M^2 \in$ MEMS(M). If $M^* \neq \emptyset$ then $M^* \lessdot M_{22}$, since, otherwise, M^* would be

an equilibrial market segment within M^2 (which, again, would contradict $M^2 \in$ MEMS(M)). However, $M^* \subset M^1$, and consequently, by Def. 4.28 of $\lessdot$, $M^* \lessdot M_{22}$ implies that $M^1 \lessdot M^2$. This contradicts the fact[6] that $M^1 \in$ MEMS(M). So $M^* = \emptyset$.

■

Lemma 4.34 implies that the fine-grain structure of a canonical stratification looks like Fig. 4.4: The top stratum M^1 of the canonical stratification of a SPAM M is the direct sum of all MEMSs of M. Stratum M^2 is the direct sum of the *reopenings* of the MEMSs of $\lceil M \backslash M^1 \rceil$, ... *et cetera*. We will refer to the components M^i_j (which are reopenings of MEMSs) as the *atoms* of C_M. The arrows denote $\lessdot$-relationship between these atoms.

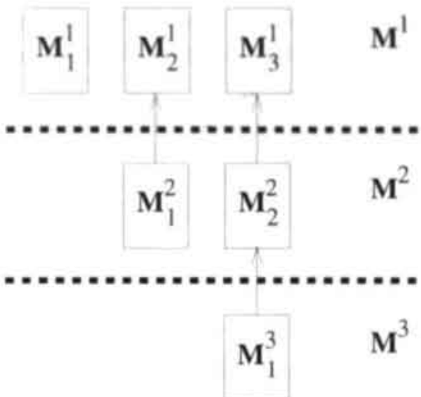


Figure 4.4: The fine-grain structure of a canonical stratification.

To some extent, Fig. 4.4 also applies to ordinary (noncanonical) stratifications. For an ordinary stratification $S_M = \langle M^1, \ldots, M^m \rangle$, each of the strata M^j is the reopening of an EMS of $\lceil \bigcup_{h \geq j} M^h \rceil$, and consequently, the direct sum of atoms. The difference between the canonical stratification and an ordinary (noncanonical) one is that, in an ordinary stratification, a stratum M^i need not contain *all* of the atoms (reopenings of MEMSs) of the submarket $\mathcal{M}^i = \lceil \bigcup_{j \geq i} M^j \rceil$. We could, for example, move the atom M^1_1 from M^1 to M^3, and the modified segmentation would still be a stratification. In view of our goal to arrive at an arbitrary stratification by applying a transformation to the canonical stratification, two questions arise.

1. Does every relocation of atoms in the canonical stratification render a stratification?

2. Can every stratification be constructed from the canonical stratification via atom relocation?

[6] $M^1 \in$ MEMS$(M) \Rightarrow M^1 \in$ EMS(M)

The answer to the first question is negative. The relocation must be such that all of the $\lessdot$-arrows point upward and cross stratum boundaries. If we, for example, relocate atom M_2^1 (cf. Fig. 4.4) into stratum M^2 or M^3, the modified segmentation $\langle M^1, M^2, M^3 \rangle$ is no longer a stratification.

The answer to the second question is affirmative. As the following lemma states, the set of atoms of any stratification of a SPAM M equals that of the canonical stratification C_M of M. In other words, it is not possible to construct a stratification by recombining fragments of atoms in C_M into new atoms which do not occur in C_M.

Lemma 4.35
Let C_M be the canonical stratification of a SPAM market M, and let S_M be an arbitrary stratification of M. If $\mathfrak{A}_C$ and $\mathfrak{A}_S$ denote the atom sets of C_M and S_M respectively, then $\mathfrak{A}_C = \mathfrak{A}_S$.

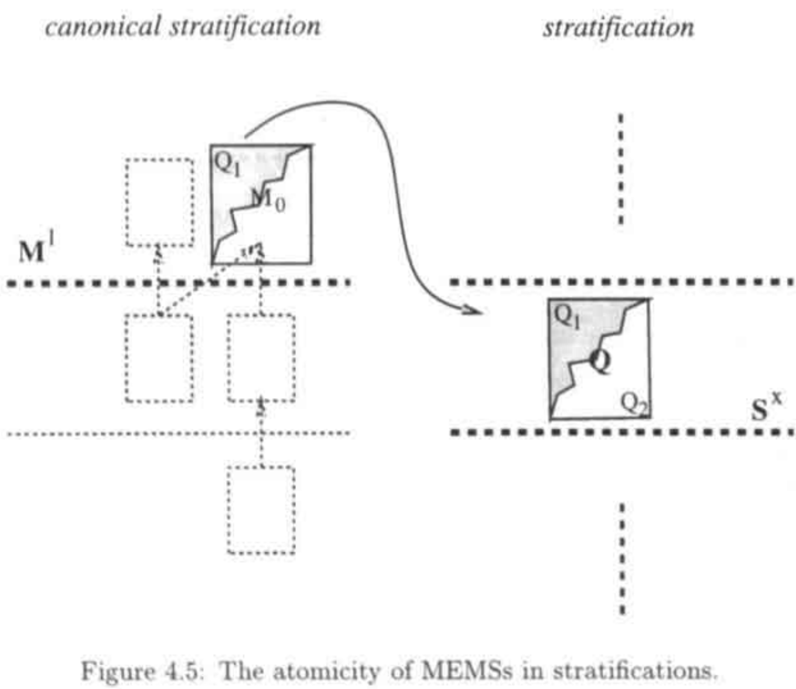


Figure 4.5: The atomicity of MEMSs in stratifications.

Proof (with induction to the stratum number i).
Let $M^i = M_1^i \oplus \ldots \oplus M_{l(i)}^i$ (cf. Lemma 4.34) where each of the M_j^i are atoms of C_M. Let $S_M = \langle S^1, \ldots, S^n \rangle$ be an arbitrary stratification of M, and let $\mathfrak{A}_S(S^p)$ denote the set of atoms which occur in stratum S^p. Then

$$(\forall i : 1 \leq i \leq k)(\forall j : 1 \leq j \leq l(i))(\exists p : 1 \leq p \leq n)\ M_j^i \in \mathfrak{A}_S(S^p)$$

First step ($i = 1$):
We prove that every atom in the top stratum of the canonical stratification C_M occurs, as an atom, in some stratum of any stratification S_M. Let $M_0 = \langle I^0, R^0, e^0, P\rangle$ be an arbitrary MEMS of $M = \langle I, R, e, P\rangle$. Since $M_0 = \Re_M(M_0)$, M_0 is, in fact, an (arbitrary) atom in the top stratum M^1 of C_M. Since C_M and S_M are stratifications of one and the same market M, all of the tools occurring in M_0 must occur in some stratum S^p of S_M. Let S^x be the highest stratum (i.e., the one with minimal x) of S_M which contains one or more tools from M_0, and let $Q = \langle J, F, a, P\rangle$ be an atom of S^x in which such tools occur. Let $\widehat{Q} = \langle J, F, a, Z\rangle$ be the MEMS of $\lceil \bigcup_{y \geq x} S^y \rceil$ from which Q is computed (via $Q = \Re_M(\widehat{Q})$). The situation is depicted in Fig. 4.5. Define $F_1 = F \cap R^0$ and $F_2 = F \setminus R^0$, and let $Q_1 = \langle J_1, F_1, a\restriction_{J_1}, P\restriction_{J_1}\rangle$ be the subsegment of Q which contains F_1, and $Q_2 = \langle J_2, F_2, a\restriction_{J_2}, P\restriction_{J_2}\rangle$ the complementary subsegment. Obviously, $F_1 \neq \emptyset$. Since $Q_1 \subset M_0$ and M_0 is a segment within the *top stratum* of C_M, the first preference of agents in Q_1 is not constrained by C_M, that is, $(\forall i \in J_1)\ \max_{\succ_i}(R) \in R^1$. Moreover, in view of the fact that M_0 is an *EMS* of M, all of the (unconstrained) first preferences of agents in Q_1 are in $R^0 \subset R^1$. Since no tools from R^0 occur in any stratum above S^x in S_M, the agents in Q_1 are not *effectively* constrained by S_M either. Formally,

$$(\forall i \in J_1)\ \max_{Z_i}(R) = \max_{\succ_i}(R) \in R^0 \tag{4.7}$$

We now turn to the MEMS $\widehat{Q}$ from which the atom $Q \in S^x$ was constructed. $\widehat{Q}$ is an EMS in $\lceil \bigcup_{y \geq x} S^y \rceil$, so the first preferences $\max_{Z_i}(R)$ of all agents in $\widehat{Q_1}$ must lie in F. Because of (4.7), we know that the first preferences $\max_{\succ_i}(R)$ of the agents within $Q_1 \subseteq Q$ also lie in F. In fact, since $F_2 \cap R^0 = \emptyset$, they must be in F_1. This implies (again, by (4.7)) that, within the equilibrial segment $\widehat{Q}$ of S_M, none of the first preferences of the agents in Q_1 lie in F_2. Consequently, $\widehat{Q_1}$ is an equilibrial subsegment within $\lceil \bigcup_{y \geq x} S^y \rceil$. Since $\widehat{Q_1} \subset \widehat{Q}$ and $\widehat{Q} \in \text{MEMS}(\lceil \bigcup_{y \geq x} S^y \rceil)$, this implies that $\widehat{Q_1} = \widehat{Q}$ and hence $Q_1 = Q$.

Induction step ($1, 2, \ldots, i-1 \to i$):
Let $\mathfrak{A}_S(S^j)$ denote the subset of atoms in stratum S^j of the stratification S_M, and $\mathfrak{A}_C(M^j)$ the corresponding set in stratum M^j of C_M. Then the induction hypothesis states that, for all $X \in \mathfrak{A}_C(M)$,

$$\begin{aligned} &(\exists j \in \{1, \ldots, i-1\})\ X \in \mathfrak{A}_C(M^j) \ \Rightarrow \\ &(\exists p \in \{1, \ldots\})\ X \in \mathfrak{A}_S(S^p) \end{aligned}$$

Consequently, we can remove the atoms in $\bigcup_{j=1}^{i-1} \mathfrak{A}_C(M^j)$ from C_M as well as S_M, and arrive at a canonical stratification $C_{\widetilde{M}}$ and an ordinary stratification $S_{\widetilde{M}}$ of $\widetilde{M} = \lceil \bigcup_{j \geq i} M^j \rceil$. We now face exactly the same task as in the first step of the proof. We must prove that

All atoms in the top stratum (here: M^i) of the canonical stratification $C_{\widetilde{M}}$ of a market $\widetilde{M}$ must occur *as a whole* in some stratum of the ordinary stratification $S_{\widetilde{M}}$ of $\widetilde{M}$.

This can be proven by using the same line of reasoning as in the first step ($i = 1$).

■

Corollary 4.36
Every stratification of a stratifiable SPAM M contains the same atoms.

Lemma 4.37
If $\mathfrak{A}_S(M)$ denotes the set of atoms of the stratification S_M of M, then the Walrasian allocation associated with S_M is defined by

$$a \in \mathfrak{A}_S(M) \wedge \lceil a \rceil = \langle I^*, R^*, e^*, P^* \rangle \Rightarrow (\forall i \in I^*)\, w(i) = \max_{P_i^*}(R^*)$$

Proof.
Let A be an arbitrary atom in $\mathfrak{A}_S(M)$, and let $M^k = \langle I^k, R^k, e^k, P^k \rangle$ be the stratum in which A resides. Let $\hat{A} = \langle I^*, R^*, e^*, P^k \rangle$ denote the MESM of $\lceil \bigcup_{h \geq k} S^h \rceil$ from which A was computed (via $A = \mathfrak{R}_M(\hat{A})$). By Prop. 4.27, the Walrasian allocation associated with S_M is defined by

$$(\forall i \in I)\, w(i) = \max_{P_i}(\{r \in R \mid r \sim e(i)\})$$

Hence,

$$(\forall i \in I^*)\, w(i) = \max_{P_i}(R^k)$$

Since $P^k = P\lceil_{R^k}$, this implies that

$$(\forall i \in I^*)\, w(i) = \max_{P_i^k}(R) \tag{4.8}$$

By condition (ii) of Def. 4.29 (EMS), $(\forall i \in I^*)\ \max_{P_i^k}(R) \in R^*$. Since $\lceil A \rceil = \lceil \hat{A} \rceil$, $P^* = \hat{P}\lceil_{R^*}$. Hence,

$$(\forall i \in I^*)\ \max_{P_i^*}(R^*) = \max_{P_i^*}(R) = \max_{P_i^k}(R)$$

Combining this with (4.8) renders

$$(\forall i \in I^*)\, w(i) = \max_{P_i^*}(R^*)$$

■

Proposition 4.38
In any stratifiable SPAM, a unique Walrasian allocation exists.

Proof.
Corollary 4.36 states that all of the stratifications of a stratifiable SPAM M contain the same atoms, while Lemma 4.37 entails that the Walrasian allocation associated with a stratification depends only on the set of atoms of that stratification. Hence, every stratification of a stratifiable SPAM is associated with the same Walrasian allocation.

■

4.4 Verification of Stratifiability in SPAM

Having characterized the Walrasian SPAMs in terms of their preference profile and initial endowment as *stratifiable* markets, and gathered some insight into the structure of stratifications, the time has come to tackle the second question of interest: 'How can we check (within an acceptable time) whether a Walrasian equilibrium exists in a given SPAM.'

A first, rather obvious opportunity to answer this question is furnished by applying Prop. 4.14 to SPAMs. This leads to the conclusion that every price order $\gtrsim$ unambiguously[7] defines a final allocation w by

$$(\forall i \in I)\; w(i) = \max_{P_i}(\{r \in R \mid e(i) \gtrsim r\}) \tag{4.9}$$

Hence, we could answer the question whether a Walrasian equilibrium exists in a given SPAM involving m tool types in finite time by simply setting tool prices according to all of the $|\text{PREORD}[m]|$ price orders consecutively and checking whether the final allocation w determined by (4.9) satisfies the conservation of tools constraint (condition (iii) of Def. 4.16). In the mean time, we know from Prop. 4.27 that this requires each agent to be endowed (in the final allocation) with a tool type from its own segment, which would speed up the verification of Walrasianess some more.

However, for this procedure to be tractable, $|\text{PREORD}[m]|$ should not grow too rapidly with m. Proposition 4.39 below reveals that this requirement is not met. As an illustration, $|\text{PREORD}[m]| = 3, 13, 75, 541, 4683, 47293$, and 545835 for $m = 2, 3, 4, 5, 6, 7$, and 8 respectively.

[7] The unambiguity is guaranteed within SPAM, but not in AM in general, since $best_{\succsim_i}(\cdot)$ is generally a set of more than one tool if $\succsim_i$ is a weak preference.

Proposition 4.39

$$|\text{PREORD}[m]| = \sum_{n=1}^{m}\sum_{k=0}^{n}(-1)^k \binom{n}{n-k} (n-k)^m \tag{4.10}$$

Proof.
Let $S_{m,n}$ denote the number of different partitions of m elements into n equivalence classes. These numbers, known as "the Stirling numbers of the second kind", equal (Grimaldi, 1985)

$$S_{m,n} = \frac{1}{n!}\sum_{k=0}^{n}(-1)^k \binom{n}{n-k} (n-k)^m \tag{4.11}$$

The space of *strict* total orders over a set of size m can be identified with the space $\text{PERM}[m]$ of permutations of the set $\{1, 2, \ldots, m\}$, and therefore contains $m!$ elements. The set of total preorders can be partitioned into subsets $O_{m,n}$ of preorders featuring n equivalence classes (for $n = 1, \ldots, m$). Then

$$|\text{PREORD}[m]| = \sum_{n=1}^{m} |O_{m,n}| = \sum_{n=1}^{m} n! \cdot S_{m,n} \tag{4.12}$$

Eq. 4.12 expresses that the total number of orders equals the sum of the cardinalities of each of these subsets, where the cardinality of $O_{m,n}$ equals the number of strict total orders over n elements, multiplied by $S_{m,n}$, the number of different partitions of m elements into n equivalence classes. Substituting (4.11) into (4.12) yields (4.10).

■

As such, testing for market stratifiability by enumeration of all conceivable price preorders is not a feasible option. Fortunately, we have by now gained sufficient insight into the properties of stratifications to come up with a better alternative. The algorithm is based on a property of the top stratum M^1 of a canonical stratification $\langle M^1, \ldots, M^k \rangle$, which we will now prove. To this avail, we first define the supply and demand for tool types in a market *segment*.

Definition 4.40 (supply, demand, and oversupply)
Let $M = \langle I, R, e, P \rangle$ be a market segment of a SPAM.
Then the demand for $r \in R$ in M equals

$$\sigma^+(r, M) = |\{i \in I \mid \max_{\succ_i}(R) = r\}| \tag{4.13}$$

and the supply of $r \in R$ in M equals

$$\sigma^-(r, M) = |\{i \in I \mid e(i) = r\}| \tag{4.14}$$

A tool $r \in R$ is called oversupplied in M if its supply exceeds its demand. △

Proposition 4.41
In a stratifiable SPAM M with canonical stratification $\langle M^1, \ldots, M^k \rangle$, R^1 does not contain any tool types which are oversupplied in any market segment $\mathcal{M}$ such that $M^1 \subseteq \mathcal{M} \subseteq M$.

Proof.
Since $M^1 \subseteq \mathcal{M}$,

$$\sigma^+(r, M^1) \leq \sigma^+(r, \mathcal{M}) \tag{4.15}$$

Since $r \in R^1$,

$$\sigma^-(r, M^1) = \sigma^-(r, \mathcal{M}) \tag{4.16}$$

Since M^1 is an equilibrial market segment of M,

$$\sigma^+(r, M^1) - \sigma^-(r, M^1) = 0$$

Hence, by (4.15), (4.16),

$$0 = \sigma^+(r, M^1) - \sigma^-(r, M^1) \leq \sigma^+(r, \mathcal{M}) - \sigma^-(r, \mathcal{M}) \Rightarrow$$

$$\Rightarrow \sigma^-(r, \mathcal{M}) \leq \sigma^+(r, \mathcal{M})$$

■

Proposition 4.41 provides an efficient algorithm to check whether a Walrasian allocation exists for a SPAM $M = \langle I, R, e, P \rangle$. The algorithm attempts to compute a canonical stratification of M through iterative computation of $hd\, tl^k\, M$ for $k = 0, 1, \ldots$ When a newly computed tail is empty, the stratification is complete. When the head appears to be empty while the tail is not, the market is not stratifiable.

This gTTC algorithm is specified in pseudocode in Table 4.1.

The most important steps in gTTC are those of the FORALL statement, which comprise the removal of any oversupplied tool types from the current top stratum R^1. This is justified by Prop. 4.41.

In the actual implementation of gTTC in the IG testbed, the WHILE-loop surrounding the FORALL statement in Table 4.1 is optimized, by performing a transitive-closure operation before recomputing σ. This operation repeatedly moves any tool types initially assigned to agents in I^1 with a first preference in R^2 from R^1 to $R2$.

In the form listed in Table 4.1, gTTC only reveals *whether* the market possesses a stratification. However, it can easily be made to render the price preorder of the stratification (presuming that it exists) by adding a print statement (viz. a command to print the string of tool types in R^1, followed by a dot) just before the statement $R^1 := R^2$ in the outer loop of Table 4.1.

The pseudocode below presumes that the market M is a tuple $\langle I, R, e, P\rangle$, and that every update of R^1 triggers a daemon which updates M^1 accordingly, that is, $M^1 := \langle I^1, R^1, e^1, P^1\rangle$, where

- $I^1 \triangleq \{i \in I \mid E(i) \in R^1\}$
- $e^1 \triangleq e\restriction_{R^1}$
- $P^1 \triangleq (P\restriction_{R^1 \cup R^2})\restriction_{I^1}$

```
BOOLEAN FUNCTION stratifiable ( MARKET M );
TOOL-TYPE-SET R¹, R²;
TOOL-TYPE r;
MARKET M¹;
SCARCITY-PROFILE σ;
BOOLEAN success;

BEGIN
        R¹ := ∅; R² := R; success:= TRUE;
        WHILE ( R² ≠ ∅ ) DO
        BEGIN
                R¹ := R²; R² := ∅;
                σ := scarcities-in(M¹);
                WHILE ( σ ≠ ∅ ) DO
                BEGIN
                        FORALL ( r ∈ σ⁻ ) move-from-R¹-to-R²(r);
                        σ := scarcities-in(M¹);
                END;
                IF ( R¹ = ∅ ) THEN
                BEGIN
                        success := FALSE; R² := ∅;
                END;
        END;
        RETURN success;
END;
```

Table 4.1: The gTTC algorithm to determine market stratifiability.

The definition of P^1 in the preamble of the algorithm effectively prevents the agents in $I^1 \cup I^2$ from submitting a bid on the tools in $R \backslash (R^1 \cup R^2)$. In a Walrasian auction, this is effectuated by redefinition of the price assignment. Hence, by decreasing the prices of the tool types in R^1 after each traversal of the outer loop in gTTC *instead* *of* redefining P^1, we can turn gTTC into an auction protocol for the auctioneer in a Walrasian auction. Note that this is a tâtonnement protocol, that is, a price-adaptation procedure which is purely based on the scarcity information in σ.

Obviously, the label gTTC was chosen in view of the reminiscence with TTC. Contrary to TTC, gTTC is a tâtonnement protocol. Yet, one can say that gTTC generalizes TTC to *typed* reassignment problems, for if both are applied to the same *untyped* reassignment problem, the outcomes are identical. To see this, we make the following observations.

Since Prop. 4.41, on which gTTC is based, is a statement about oversupply of tool types in the top stratum of the *canonical* stratification, the price preorder computed by gTTC is the canonical one. In other words, the top stratum of a stratification rendered by gTTC is always the *largest* equilibrial market segment of M. This implies that, if we apply gTTC to an *untyped* assignment market, the top stratum of the resulting stratification equals the solution rendered by elementary LBF-composition (i.e., ECP* defined in Def. 3.7 on page 55) for this market. Consequently, the (Walrasian) allocation rendered by gTTC is identical to the outcome of LBF composition/revision.

As we argued in Sect. 3.3.6, TTC is tantamount to composition/revision with random cluster composition (ECP$^{\odot}$). At first sight, this would seem to imply that the solutions rendered by TTC and gTTC are not necessarily the same. This is true for typed assignment markets in general, for which gTTC renders at most one allocation, while TTC can render many different ones. However, on *untyped* assignment markets, there is a Walrasian allocation, which — if the preference profile is strict — is always rendered by both. Indeed, while the set of possible outcomes of LBF composition/revision is generally a subset of the set of outcomes of random cluster composition/revision, the outcomes of both are unique and identical in the case of untyped reassignment. This is due to the fact that the indegree of a node (which denotes the supply of the associated tool type) in a tool graph associated with an untyped reassignment problem always equals 1. As a consequence, cycles in such a tool graph are always arc-disjoint. Consequently, the order in which cycles are selected does not matter.

4.5 Pruning and Structuring SPAM

The algorithm described in the previous section enables us to test whether a particular SPAM is a Walrasian market. It can verify/falsify the existence of Walrasian equilibria in 10,000 markets of moderate size (with, say, 12 agents and 5 tool types) within a few seconds (on a SUN-4). By taking a random sample from SPAM, and applying the test to each market in the sample, we could acquire an estimate of the 'Walrasian density' within SPAM, that is, the probability that a randomly selected market possesses a Walrasian equilibrium. However, SPAM is an infinite problem space (without restriction of the number of agents in a market). Presenting an estimate based on markets with $|I| < n$ as an estimate of Walrasian density in SPAM would therefore be incorrect. Moreover, we can acquire more insight if several subspaces of SPAM are investigated separately. Hence, we will compute estimates $\widehat{d}(n,m)$ for the density $d(n,m)$ within $\text{SPAM}(n,m)$, that is, the subspace of SPAMs $\langle I, R, e, P\rangle$ with $|I| = n$ and $|R| = m$, for various n and m. In fact, the subspaces from which we take our samples are much smaller than $\text{SPAM}(n,m)$, for two reasons. First, not all problem instances that are formally in $\text{SPAM}(n,m)$ constitute viable problems, and second, we can filter out all alphabetic variants of a sample problem. This reduces the effective problem space considerably, as we will show in the next two sections.

4.5.1 Restriction to viable problem instances

Not every assignment market in SPAM constitutes a *viable* problem. As an example, a SPAM in which each agent already possesses its favorite tool type is not much of a problem. More generally, a market featuring n agents and m tool types does not constitute a viable problem in $\text{SPAM}(n,m)$ if *any* of these agents possess their first preference. If k of the n agents are initially endowed with their first preference, then the market is a (viable) problem in $\text{SPAM}(n-k,m)$; not in $\text{SPAM}(n,m)$.

This leads to the following formal definition for the problem space $\text{SPAM}^*(n,m)$ of *viable* SPAMs with n agents and m tool types.

Definition 4.42 (viable SPAM)
A SPAM $M = \langle I, R, e, P\rangle$ is an element of the problem space $\text{SPAM}^*(n,m)$ iff no agent already possesses its most-preferred tool type:

$$(\forall i \in I)\, max_{\succ_i}(R) \neq e(i)$$

Furthermore, we define $\text{SPAM}^* \triangleq \bigcup_{n \geq m \geq 2} \text{SPAM}^*(n,m)$ △

The restriction of the subclasses of SPAM to viable problem instances implies that certain 'constellations' cannot occur. It is, for instance, not possible that all agents in a SPAM^* instance have the same preference.

Example 4.43 (identical preferences)
A SPAM in which the preference of all agents are identical is not viable in $\text{SPAM}(n,m)$ *for any* $n, m > 0$.
Consider, for example, $M = \langle I, R, e, P\rangle$ *with* $|I| = 5$, $|R| = 4$, *and* $P_i = [abcd]$ *for* $i = 1, \ldots, 5$. *Since our definition of markets requires these to be closed,*[8] *there*

[8] Here, "closed" means that $P = P\lceil_R$ and $(\forall r \in R)\, r \in \underline{\downarrow}\, Im\, e$.

must be some agent in I (initially) endowed with a type-a tool. This agent already possesses its first preference and is therefore not a viable participant in the market. However, after removing all such phantom agents, there is no type-a tool left in the community ($a \notin \underline{\downarrow} Im\, e$). Consequently, the tool type a must be removed from the preference profile also. The new profile consists of the (identical) preferences [bcd]. The same line of reasoning can be applied to the other tool types until — ultimately — there are no agents or tool types left.

△

Ex. 4.43 specifies a preference profile P, such that the market $\langle e, P\rangle$ is not a viable problem for *any* allocation e. Such profiles are rare. In fact, Ex. 4.43 (i.e., a profile of identical preferences) is the only case we can think of. However, there are many *combinations* of a preference profile P and a community tool bag Γ which are incompatible, in the sense that there exist no viable market $\langle e, P\rangle$ such that $\underline{\downarrow} Im\, e = \Gamma$. The following proposition characterizes these combinations in terms of Γ and the bag π of first preferences of P.

Proposition 4.44
Let R be a set of m tool types, P a profile of n strict preferences over R, and $\Gamma \in \mathfrak{B}(R)$ a tool bag with $|\Gamma| = n$.
Define $\pi \in \mathfrak{B}(R)$ by $\pi \triangleq \{\text{best}_{P_i}(R) \mid i \in I\}$, and let $\text{SPAM}^*(\Gamma, P)$ denote the space of viable markets $\langle e, P\rangle$ with preference profile P and community tool bag $\underline{\downarrow} Im\, e = \Gamma$. Let E_P denote the set of viable allocations for P, that is, $E_P \triangleq \{e \in R^I \mid \langle e, P\rangle \in \text{SPAM}^*(\Gamma, P)\}$.
Then a necessary and sufficient condition for $E_P \neq \emptyset$ is

$$(\forall r \in R)\ \ \pi(r) + \Gamma(r) \leq n \tag{4.17}$$

Proof.
The necessity of (4.17) can easily be established. Define, for $i = 1, \ldots, n$, $h(i) = \max_{P_i}$. Suppose that $E_P \neq \emptyset$. Then choose $e \in E_P$, $r \in R$ arbitrarily, and define

$$\begin{aligned} I_r &\triangleq \{i \in I \mid e(i) = r\} \\ J_r &\triangleq \{i \in I \mid h(i) = r\} \\ K_r &\triangleq \{i \in I \mid h(i) \neq r\} \end{aligned}$$

Then $|J_r| = \pi(r)$ and $|I_r| = \Gamma(r)$. (†)
Furthermore,

$$\begin{aligned} I = J_r \oplus K_r \quad &\Rightarrow \\ |I| = |J_r| + |K_r| \quad &\Rightarrow \end{aligned}$$

$$|K_r| = |I| - |J_r| = n - \pi(r) \tag{‡}$$

Since e is a *viable* allocation,

$$\begin{array}{lll} (\forall i \in I) \ \ (e(i) = r \Rightarrow h(i) \neq r) & & \Rightarrow \\ (\forall i \in I_r) & h(i) \neq r & \Rightarrow \\ I_r \subseteq K_r & \Rightarrow & |I_r| \leq |K_r| \end{array}$$

With (†) and (‡), $\Gamma(r) = |I_r| \leq |K_r| = n - \pi(r)$.
Hence, $\Gamma(r) + \pi(r) \leq n$.

We prove sufficiency by induction for $m \geq 2$.
The initial step ($m = 2$):
Let $R = \{r_1, r_2\}$ and let Γ and π be defined according to the description above. Then (4.17) comes down to

$$\left\{ \begin{array}{lll} \pi(r_1) & +\Gamma(r_1) & \leq n \\ \pi(r_2) & +\Gamma(r_2) & \leq n \end{array} \right. \tag{4.18}$$

Since $\pi, \Gamma \in \mathfrak{B}(\{r_1, r_2\})$, and $|\pi| = |\Gamma| = n$, $\pi(r_1) + \pi(r_2) = n = \Gamma(r_1) + \Gamma(r_2)$. With (4.18), this implies that

$$\left\{ \begin{array}{lll} \pi(r_1) & \leq n - \Gamma(r_1) & = \Gamma(r_2) \\ \pi(r_2) & \leq n - \Gamma(r_2) & = \Gamma(r_1) \end{array} \right.$$

In other words, the number of tools of type r_2 suffices to allocate an r_2 to each of the agents with a first preference for r_1 and vice versa. This defines a viable allocation.

Induction step: $(\{2, \ldots, m\} \rightarrow m + 1)$
Let $R = \{r_1, \ldots, r_{m+1}\}$, and $I = \{1, \ldots, n\}$. Define π and $\Gamma \in \mathfrak{B}(R)$ as before, and suppose that (4.17) holds, that is,

$$\left\{ \begin{array}{lll} \pi(r_1) & +\Gamma(r_1) & \leq n \\ \vdots & & \\ \pi(r_{m+1}) & +\Gamma(r_{m+1}) & \leq n \end{array} \right. \tag{4.19}$$

We will show that there exist a *partial* allocation involving (at least) the tools of type r_{m+1} and the agents with a first preference for type r_{m+1}, which is viable, and transforms the set of inequalities (4.19) into a set of at most m inequalities. By the induction hypothesis, there exists a viable allocation involving the tool types in this secondary set of inequalities. Combining this allocation with the partial one renders a viable allocation that involves all of the $m + 1$ tool types.
The partial allocation is constructed as follows. Define the *slack* S_i of tool type r_i as

$$S_i \triangleq n - \pi(r_i) - \Gamma(r_i)$$

Then the total slack in R equals

$$S_R \triangleq \sum_{i=1}^{m+1} S_i = \sum_{i=1}^{m+1} (n - \pi(r_i) - \Gamma(r_i)) = (m+1)n - n - n = (m-1)n$$

The partial allocation involves $\pi(r_{m+1}) + \Gamma(r_{m+1})$ allocation steps, where each step comprises the allocation of one tool type to one agent. When we allocate a tool of type r_{m+1} to an agent with a first preference for $r_i \neq r_{m+1}$, we must update (subtract one from) the values of n and $\pi(r_i)$. This does not alter the slacks S_i and S_{m+1} but it reduces the slack of each of the tool types in $R \backslash \{r_i, r_{m+1}\}$ by 1, so the total slack S_R is reduced by $m-1$.
The same holds when we allocate a tool of type $r_i \neq r_{m+1}$ to an agent with a first preference for r_{m+1}. The associated updates of n and $\Gamma(r_i)$ do not alter S_i and S_{m+1}, but they reduce the total slack by $m-1$.
Thus, each *step* in the construction of the partial allocation involves a decrease of n by one, and a decrease of the total slack by $m-1$. This implies that the equation $S_R = (m-1)n$ remains valid throughout the construction. If we ensure that, in each step, the selected tool type r_i has *minimal* slack (i.e., $S_i = \min\{S_1, \dots, S_m\}$), then the step does not invalidate (4.19) as long as there is *at most one* tool type with zero slack. We will show that this condition is fulfilled during the entire construction process. Suppose that two or more tool types in $R \backslash \{r_{m+1}\}$ have zero slack. Then the total slack of $(m-1)n$ must stem from at most $m-1$ tool types (including r_{m+1}). Since $S_i \leq n$ by definition, this would imply that $S_i = n$ for all of the tool types, which, in turn, implies that $\pi(r_i) = \Gamma(r_i) = 0$. In other words, the situation that we cannot perform a step without invalidating (4.19) can only occur when we have already completed a viable allocation involving all $m+1$ tool types.

■

Next to 'nonproblems' like the instance in Example 4.43, Def. 4.42 of problem viability also excludes problems that are already solved, that is, problem instances where the initial allocation is a Walrasian allocation. This is expressed formally in the following proposition.

Proposition 4.45
Let W_M denote the set of Walrasian allocations of the market $M = \langle I, R, e, P \rangle \in$ SPAM. Then*

$$e \notin W_M$$

Proof.
If the market is not Walrasian, $W_M = \emptyset$, so $e \notin W_M$. Otherwise, $e \in W_M \Leftrightarrow W_M = \{e\}$. Suppose that this is the case.

Let r be a tool type in the top stratum of a stratification associated with e. Then any agent i with $e(i) = r$ will possess its most-preferred tool type in the Walrasian allocation. Since $W_M = \{e\}$, this most-preferred tool type appears to be $e(i)$. Hence, e is not a viable initial allocation. This contradicts $M \in$ SPAM*. So $e \in W_M$ cannot be true.

■

4.5.2 Exclusion of alphabetic variants

The software that was developed to test the stratifiability of a market in SPAM is sufficiently fast to consider the possibility of *exact* computation of the density $d(\Gamma, P)$ in a subspace of all markets with a certain, specified preference profile P, and a certain, specified tool bag Γ.[9] Such exact computation involves the generation of all initial allocations e of Γ over I, such that the market $\langle e, P\rangle$ constitutes a viable problem in SPAM(n, m).[10] Then the density $d(\Gamma, P)$ equals the fraction of these initial allocations for which the associated market is stratifiable. By selecting a random sample of preference profiles, and averaging over the associated densities $d(\Gamma, P)$, an estimate $\hat{d}(\Gamma)$ of $d(\Gamma)$ can be acquired. Computation of estimates $\hat{d}(\Gamma)$ for *all* tool bags Γ with $|\Gamma| = n$ and $|\check{\Gamma}| = m$ would provide us with an impression of the density in SPAM$^*(\Gamma)$ as a function of Γ. However, even for relatively small values of n and m, the number of such tool bags is too large for such computation to be feasible.

Fortunately, it is not necessary to compute estimates for *all* of these bags. Provided that the size of the profile sample is sufficiently large, $\hat{d}(\Gamma)$ will be approximately the same for many different bags Γ, because $d(\Gamma_1) = d(\Gamma_2)$ if Γ_1 and Γ_2 are *alphabetic variants*.

Example 4.46 (Alphabetic variants)
Consider the markets $M_1 = [ab\underline{c},\ b\underline{a}c,\ ac\underline{b},\ cb\underline{a}]$ and $M_2 = [cb\underline{a},\ b\underline{c}a,\ ca\underline{b},\ ab\underline{c}]$. It is obvious, that M_1 is stratifiable iff M_2 is, for the only difference between the two markets is that the tool type which is called "a" in M_1 is called "c" in M_2, and vice versa. The associated tool bags are $\Gamma_1 = \{a : 2, b, c\}$ and $\Gamma_2 = \{a, b, c : 2\}$. If we were to compute the densities $d(\Gamma_1)$ and $d(\Gamma_2)$ exactly, *that is, use a 'sample' which consists of* all *possible profiles of 4 agents over $R = \{a, b, c\}$, then the outcomes would be identical. As illustrated in Fig. 4.6, any profile P_1 that forms a stratifiable market with the initial allocation $e_1 \in \mathcal{A}(\Gamma_1, I)$ corresponds in a one-to-one fashion with an initial allocation $e_2 \in \mathcal{A}(\Gamma_2, I)$ and a profile P_2, where $\langle e_2, P_2\rangle$ is the alphabetic variant of $\langle e_1, P_1\rangle$ which is arrived at by exchanging the a's and the c's. Consequently, an estimate for Walrasian density in* SPAM$^*(n, m)$ *based on only one tool bag Γ_1 is just as reliable as an estimate based on a number of tool bags Γ_1, Γ_2, $\cdots$, if all of these bags are alphabetic variants.* △

[9] In practice, exact computation is feasible only for small dimensions, say $m < 5$, and $n < 15$.
[10] Here, $n = |\Gamma|$ and $m = |\check{\Gamma}|$.

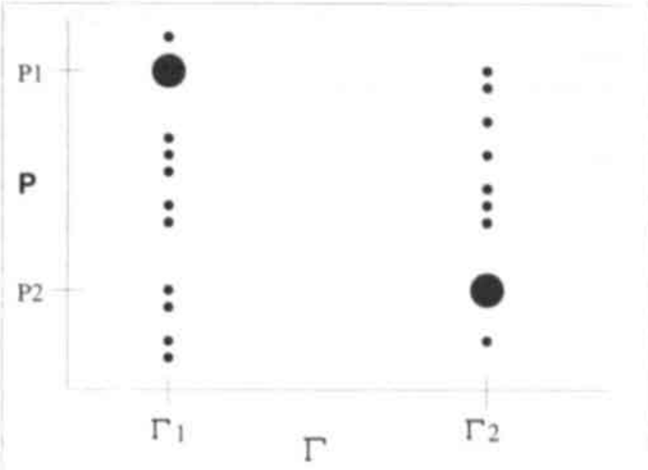


Figure 4.6: For two tool bags Γ_1 and Γ_2 which are alphabetic variants, the associated tool type renaming σ defines mappings between preference profiles and initial allocations such that $\langle e_1, P_1\rangle$ and $\langle e_2, P_2\rangle$ (where $e_2 = \sigma(e_1)$ and $P_2 = \sigma(P_1)$) are alphabetic variants, in other words, essentially the same markets.

Consequently, we need only investigate one tool bag from each equivalence class under alphabetic variation. As pointed out in Sect. 2.2.2, each of these equivalence classes corresponds with a *multiplicity type* of tool bags. More precisely, the set of multiplicity types of size m and sum of elements n is the normal divisor under alphabetic variation of the set of tool bags involving n tools and m tool types. For the reader's convenience, we recapitulate (and expand) Def. 2.13 of multiplicity type.

Definition 4.47 (multiplicity type)
Let $\beta \in \mathfrak{B}(S)$ *be denoted by* $\{x_1 : m_1, \ldots, x_n : m_n\}$. *Then the multiplicity type* $\xi(\beta)$ *of* β *is the bag*

$$\xi(\beta) \triangleq \{m_1, \ldots, m_n\}$$

$\mathfrak{B}_\alpha(S)$ *denotes the set of bags over* S *with multiplicity type* α, *and* $\text{SPAM}^*(\alpha)$ *denotes the set of markets* $\langle I, R, e, P\rangle \in \text{SPAM}^*$, *such that* $\xi(\Gamma) = \alpha$. △

Example 4.48 (multiplicity type)
Let $\alpha = \{a, a, a, b, b, c, d, d, f\}$, *and* $\beta = \{a, a, b, b, b, c, c, d, e\}$. *Then* $\xi(\alpha) = \xi(\beta) = \{3, 2, 2, 1, 1\}$, *and* $\xi(\xi(\alpha)) = \{2, 2, 1\}$. △

The notion of multiplicity type enables us to partition the subspaces $\text{SPAM}^*(n, m)$ into a relatively *small* number of subspaces $\text{SPAM}^*(\alpha)$ within which we can apply our algorithm to compute (estimates of) Walrasian density. The proposition below provides the means to calculate the factor by which the problem space is reduced if we use multiplicity types to get rid of alphabetic variation.

Proposition 4.49
Let α be a multiplicity type with $n = \sum_{x\in\alpha} x$, and let S be an arbitrary set of cardinality $|\alpha|$. Then the number of bags β of over S with $\xi(\beta) = \alpha$ (i.e., the number of alphabetic variants associated with α) equals

$$B(\alpha) = \frac{|\alpha|!}{\prod_{x\in\xi(\alpha)} x!}, \tag{4.20}$$

where $\xi(\alpha)$ denotes the multiplicity type of the multiplicity type α.

Proof.
Let us first look at the case $\xi(\alpha) = \{1 : m\}$, that is, α contains $m = |\alpha|$ different elements. Then the number of bags of multiplicity type α equals the number of ways in which we can label m distinguishable objects using m different labels. Obviously, this can be done in $m!$ different ways. If, however, k elements of α are all identical (and the others differ), we must divide $m!$ by $k!$ to compensate for counting identical permutations more than once. In general, if $\alpha = \{a_1 : m_1, \ldots, a_p : m_p\}$ (i.e., $\xi(\alpha) = \{m_1, \ldots, m_p\}$), then we must divide by $m_1! \cdot \ldots \cdot m_p!$ (i.e., $\prod_{x\in\xi(\alpha)} x!$), to compensate for counting identical permutations more than once.

■

Table 4.2 shows the 9 multiplicity types associated with SPAM$^*(10,4)$, and the numbers of different bags (alphabetic variants) and allocations associated with each type. The second column in the table specifies the number of bags (alphabetic variants) associated with a multiplicity type α. This number is given by (4.20) in Prop. 4.49.

As an example, the first line in Table 4.2 features the multiplicity type $\alpha = \{7, 1, 1, 1\}$. The number of bags associated with this multiplicity type equals the number of ways to label the 4 'containers' which make up the multiplicity type with 4 different labels. Note that, initially, the containers can only be distinguished by their size. The container of size 7 in α is thus distinguishable from the three containers of size 1. These three are indistinguishable, however. This amounts to a number of different possible labelings (corresponding with the bags $\{a : 7, b, c, d\}$, $\{b : 7, a, c, d\}$, $\{c : 7, a, b, d\}$, and $\{d : 7, a, b, c\}$) of $4!/3! = 4$. This number equals the value specified by (4.20), since $\xi(\alpha) = \xi(\{7, 1, 1, 1\}) = \{1, 3\}$.
From the first and third column of the table, we conclude that the number of allocations (modulo alphabetic variation), and hence, the size of the subspaces SPAM(α) increases with increasing 'balance' within the multiplicity type. The imbalanced

multiplicity type	nr of bags	nr of allocs/bag			nr of allocs
$\{7,1,1,1\}$	4	$10!/(7!)$	=	720	2880
$\{6,2,1,1\}$	12	$10!/(6!2!)$	=	2520	30,240
$\{5,3,1,1\}$	12	$10!/(5!3!)$	=	5040	60,480
$\{4,4,1,1\}$	6	$10!/(4!^2)$	=	6300	37,800
$\{5,2,2,1\}$	12	$10!/(5!2!^2)$	=	7560	90,720
$\{4,3,2,1\}$	24	$10!/(4!3!2!)$	=	12,600	302,400
$\{3,3,3,1\}$	4	$10!/(3!^3)$	=	16,800	67,200
$\{4,2,2,2\}$	4	$10!/(4!2!^3)$	=	18,900	75,600
$\{3,3,2,2\}$	6	$10!/(3!^2 2!^2)$	=	25,200	151,200
9	84			95,640	818,520

Table 4.2: Multiplicity types of bags of size 10 over $\{a, b, c, d\}$.

multiplicity type $\{7, 1, 1, 1\}$ is associated with only 720 allocations per alphabetic variant, whereas the balanced type $\{3, 3, 2, 2\}$ is associated with 25200 allocations.

Another observation made from Table 4.2 is that using multiplicity types instead of tool bags to partition SPAM$^*(n, m)$ reduces the number of subspaces for which the Walrasian density must be computed (presuming that we wish to perform an exhaustive investigation) by a factor $\frac{84}{9} > 9$.

The number $M(n, m)$ of *multiplicity types* associated with SPAM$^*(n, m)$ can be computed via a simple recurrent relation, as shown in Prop. 4.50 below. Values for various n and m, listed in Table 4.3, demonstrate that $M(n, m)$ rises relatively slowly with n and m, in comparison with the number of allocations per bag as a function of multiplicity type imbalance.

Proposition 4.50
The number of multiplicity types α with $\sum \alpha = n$ and $|\alpha| = m$ is defined by the recurrent relation

$$D(n, m) = \sum_{i=1}^{\min\{m, n-m\}} D(n - m, i) \tag{4.21}$$

with boundary conditions $D(n, 1) = D(n, n) = 1$

Proof.
The number of multiplicity types α with $\sum \alpha = n$ and $|\alpha| = m$ equals the number of ways in which n indistinguishable objects can be divided over m indistinguishable containers, with no container left empty. As such, the boundary values $D(n, 1)$ and $D(n, n)$ are obviously 1. When we have to divide the n objects over m containers, with $1 < m < n$, then m objects are needed to ensure that no container is empty.

For the remaining $n-m$, we can choose. Either we stack all of them in one container (D(n-m,1) ways to do this), or we divide them over 2 containers (D(n-m,2) ways to do this), ..., or we divide them over all of our m containers ((D(n-m,m) ways to do this). Of course, if $n-m < m$, we can divide the $n-m$ objects over at most $n-m$ containers, since each of these containers is to receive at least one object. This is exactly what is expressed by Eq. 4.21.

■

$n \setminus m$	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
3	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	2	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	3	2	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-
7	4	3	2	1	1	-	-	-	-	-	-	-	-	-	-	-	-
8	5	5	3	2	1	1	-	-	-	-	-	-	-	-	-	-	-
9	7	6	5	3	2	1	1	-	-	-	-	-	-	-	-	-	-
10	8	9	7	5	3	2	1	1	-	-	-	-	-	-	-	-	-
11	10	11	10	7	5	3	2	1	1	-	-	-	-	-	-	-	-
12	12	15	13	11	7	5	3	2	1	1	-	-	-	-	-	-	-
13	14	18	18	14	11	7	5	3	2	1	1	-	-	-	-	-	-
14	16	23	23	20	15	11	7	5	3	2	1	1	-	-	-	-	-
15	19	27	30	26	21	15	11	7	5	3	2	1	1	-	-	-	-
16	21	34	37	35	28	22	15	11	7	5	3	2	1	1	-	-	-
17	24	39	47	44	38	29	22	15	11	7	5	3	2	1	1	-	-
18	27	47	57	58	49	40	30	22	15	11	7	5	3	2	1	1	-
19	30	54	70	71	65	52	41	30	22	15	11	7	5	3	2	1	1
20	33	64	84	90	82	70	54	42	30	22	15	11	7	5	3	2	1

Table 4.3: The number of multiplicity types $M(n, m)$.

The employment of multiplicity types instead of tool bags to partition SPAM$^*(n, m)$ does not influence the precision of the density estimates, and tends to reduce the computational complexity of density estimation considerably. We abstain from presenting a table similar to Table 4.3 for $B(n, m)$ to provide support for this statement, in the conviction that a few examples suffice. One example can be derived from Table 4.2: for SPAM$^*(10, 4)$ the use of multiplicity types reduces the computational complexity by a factor of about 9 (84/9). The reduction factor tends to grow rapidly with the size of the agent population: For SPAM$^*(12, 5)$, it is about 25 (330/13), while it is only 3 for SPAM$^{(}5, 3)$.

Skeptical readers are invited to perform their own computations with help of Prop. 4.51 below, which specifies a formula for the number $B(n, m)$ of different *tool* bags occurring in SPAM$^*(n, m)$ markets, that is, the number of bags β of size $|\beta| = n$ and carrier size $|\check{\beta}| = m$.

Proposition 4.51
Let S be an arbitrary set of cardinality m, let $\xi(\beta)$ denotes the multiplicity type of the bag β, and $\Xi(n,m)$ denote the set of multiplicity types α with $|\alpha| = m$ and $\sum_{x\in\alpha} x = n$. The number $B(n,m)$ of bags $\beta \in \mathfrak{B}(S)$ of size $|\beta| = n$ and $|\check{\beta}| = m$ can be expressed in terms of $\Xi(n,m)$ as

$$B(n,m) = \sum_{\alpha\in\Xi(n,m)} \frac{|\alpha|!}{\prod_{x\in\xi(\alpha)} x!} \tag{4.22}$$

Proof.
Let $\mathfrak{B}_\alpha(S)$ denotes the set of bags over S with multiplicity type α. Then

$$\mathfrak{B}_{\alpha_1}(S) \cap \mathfrak{B}_{\alpha_2}(S) = \emptyset \quad \text{if} \quad \alpha_1 \neq \alpha_2.$$

As such,

$$B(n,m) = \sum_{\alpha\in\Xi(n,m)} |\mathfrak{B}_\alpha(S)| \tag{†}$$

The size of the set $\mathfrak{B}_\alpha(S)$ does not depend on S. In fact, for any set S of size $|\alpha|$, $|\mathfrak{B}_\alpha(S)| = B(\alpha)$, with $B(\alpha)$ defined as in Prop. 4.49, that is,

$$|\mathfrak{B}_\alpha(S)| = B(\alpha) = \frac{|\alpha|!}{\prod_{x\in\xi(\alpha)} x!} \tag{‡}$$

Substitution of (‡) into (†) renders (4.22).

■

A second useful aspect of the notion of multiplicity type (next to the reduction of the computational complexity of density estimation) is illustrated by the following proposition.

Proposition 4.52
Let $M \in \text{SPAM}^*(\alpha)$, and let, for any $x \in \alpha$, $\alpha_x \subset \alpha$ denote the subbag of multiplicities in $\alpha \setminus \{x\}$ that do not exceed x, that is, $\alpha_x \triangleq \{y \in \alpha \setminus \{x\} \mid y \leq x\}$. Suppose that

$$(\forall x \in \alpha)\; x > \sum_{y\in\alpha_x} y \tag{4.23}$$

Then M is not a Walrasian market.

Proof.
Let $M = \langle I, R, e, P\rangle \in \text{SPAM}^*(\alpha)$, and let $\Gamma = \underline{\downarrow} \mathit{Im}\, e$ denote the market's tool bag. Without loss of generality, we may assume that the tool types $r_1, \ldots, r_m$ are numbered according to nonincreasing multiplicity in $\Gamma = \underline{\downarrow} \mathit{Im}\, e$. That is, if

$\alpha = \{k_1, \ldots, k_m\}$, where $k_i = \Gamma(r_i)$ then $k_1 \geq k_2 \geq \ldots \geq k_m$.
Then (4.23) can be expressed in terms of the elements k_i of α as

$$(\forall i \in \{1, \ldots, m\})\ k_i > \sum_{j=i+1}^{m} k_j \tag{4.24}$$

Suppose that M is stratifiable. Let $M_{\succsim} = \langle M^1, \ldots \rangle$ be an arbitrary stratification of M, and let w denote the Walrasian allocation of M.
We will show by induction, for tool type $r_i\,(i = 1, \ldots, m)$, that none of the tool types resides in the top stratum of $M_{\succsim}$. Since the top stratum of a stratification is not empty by definition, it follows that M is not stratifiable, and, hence, not Walrasian.

First step (tool type r_1): Let I_1 denote the set of agents, which are initially endowed with a tool of type r_1, and let Γ_1 denote the subbag of tools from Γ which are not of type r_1. Formally, $I_1 \triangleq \{i \in I \mid e(i) = r_1\}$, and $\Gamma_1 \triangleq \downarrow Im\, e\restriction_{\overline{I_1}}$.
Now let us suppose that r_1 is in the top stratum M^1 of the (arbitrary) stratification $M_{\succsim}$ of M. Since M is a viable market, this implies that $(\forall i \in I_1)\ \max_{\succ_i}(R) \in R\backslash\{r_1\}$. Hence, $(\forall i \in I_1)\ w(i) \in R\backslash\{r_1\}$. Thus, in the Walrasian allocation, each of the k_1 agents in I_1 must possess one of the $\sum_{j=2}^{m} k_j$ tools in Γ_1. But this is not possible, since, by (4.24), $k_1 > \sum_{j=2}^{m} k_j$. So $r_1 \notin M^1$.

Induction step ($\{r_1, \ldots, r_{i-1}\} \to r_i$): If none of the tool types $r_1, \ldots, r_{i-1}$ are in the top stratum of any stratification of M, then (due to Prop. 4.15) none of these tool types can be allocated to an agent in the top stratum. (†)
Now suppose that $r_i \in M^1$. Let I_i denote the set of agents possessing a tool of type r_i, that is $I_1 = \{j \in I \mid e_j = r_i\}$. Then all of the k_i agents in I_i must receive a tool type other than r_i (because of viability), which is not an element of $\{r_1, \ldots, r_{i-1}\}$ (because of (†)). In other words, each of the k_i agents initially endowed with tool type r_i must be allocated a tool from the bag $\Gamma_i \triangleq \downarrow Im\, e\restriction_{\cup_{j>i} I_j}$ of tool types other than r_i. This, again, is not possible, since $|\Gamma_i| = \sum_{j=i+1}^{m} k_j < k_i = |I_i|$ (by (4.24)). Hence, $r_i \notin M^1$.

■

Example 4.53 (A multiplicity type without Walrasian equilibria)
If $M = \langle I, R, e, P \rangle$ is a market in SPAM* *such that $\Gamma \triangleq \downarrow Im\, e = \{a{:}4, b{:}2, c\}$, then we can be sure that no Walrasian equilibrium exists, whatever the values of P and e are.* △

The usefulness of the notion of multiplicity type is not restricted to cases to which Prop. 4.52 applies. It also tells us something about the *degree of improvement* to be expected from a Walrasian auction in cases that *come close to* Eq. 4.23.

Example 4.54 (A low-prospect multiplicity type)
If $M = \langle I, R, e, P \rangle$ is a market in SPAM* *such that $\Gamma \triangleq \downarrow Im\, e = \{a : 12, b : 3, c : 3\}$,*

then we can use the same line of reasoning as in the proof of Prop. 4.52 to deduce that, if a Walrasian equilibrium exists, the associated reallocation will not involve any type-a tools. Consequently, none of the 12 agents initially endowed with a type-a tool will gain anything by partaking in a Walrasian auction. △

If $|R| = 4$ instead of 3 as in the above example, then we cannot draw such a conclusion. This is a consequence of the fact that the top stratum of a stratifiable market in SPAM* must involve at least two tool types, but not necessarily more. Suppose, for example, that $\Gamma = \{a : 12, b : 4, c : 3, d : 3\}$. If each of the 'c-possessors' has a first preference for type d and vice versa, a top stratum consisting of c and d possessors could exist. In this case, the market is Walrasian if the preferences of the a- and b-possessors are such that the number of a-possessors preferring b over a equals the number of b-possessors preferring a over b.[11] And the equilibrium constitutes a gain for at least one a-possessor if this number is positive. Of course, we can deduce that *at most* 4 of the 12 a-possessors will profit from a Walrasian auction. Similar conclusions can be drawn for imbalanced tool bags if $|R| > 4$.

Example 4.55 (Another low-prospect multiplicity type)
Let $M = \langle I, R, e, P\rangle \in$ SPAM* *with* $\xi(M) = \langle 12, 4, 4, 2, 1\rangle$. *Then at least 9 of the twelve agents possessing the most frequent tool type will not profit from a Walrasian auction.* △

This follows immediately from the fact that (in view of Prop. 4.52) the tools with multiplicity 12 cannot be in the top stratum, and the fact that the smallest top stratum that can be formed consists of the two tools with multiplicity 4, leaving at most 3 tools to satisfy agents which initially possess a tool type of multiplicity 12.

4.5.3 Inhomogeneity and eccentricity

In order to picture the dependency of Walrasian density on the multiplicity type of a market, we would like to arrange the multiplicity types along an axis. Such arrangement suggests the existence of a scalar metric on multiplicity types. In view of the above account on the relatively low merits of Walrasian equilibria in markets with 'strongly imbalanced' tool bags, we would like the metric to reflect this informal notion of imbalance. As such, it is clear that $\{1, 1, 1, 2, 10\}$ should qualify as "less balanced" than $\{3, 3, 3, 3, 3\}$. But what about $\{2, 2, 3, 3, 5\}$ and $\{1, 1, 2, 4, 4\}$? Which of these is more balanced? Or are they equally balanced? There seems to be no objective criterion to define *the* most appropriate order, let alone a natural *metric*, on multiplicity types.

[11] The market is also stratifiable if *all* a(b)-possessors prefer their a(b) over b(a), but in this case none of these agents will gain anything.

We have tried out different scalar metrics for multiplicity imbalance. None of these appeared to allow for a *simple* characterization of the relation between imbalance and Walrasian density (like "The Walrasian density increases with multiplicity imbalance"). Below, we use two of these metrics. The first, the *inhomogeneity* of a multiplicity type, measures its variance as a series of numbers. It is relative fine-grained, in the sense that it is often (but not always) one-to-one, mapping different multiplicity types to different inhomogeneity values. The second one, the *eccentricity*, is coarser. It reflects only the maximum difference of two of the numbers in the multiplicity type. Both metrics are scaled so as to range over the closed interval $[0,1]$. Formal definitions are provided below.

The variance in a series of m positive natural numbers k_i whose sum equals n is maximal if the series is of the form $n-m+1, 1, \ldots, 1$, as in the multiplicity type $\langle 5,1,1,1\rangle$. In this case the variance equals

$$\sigma^2 \triangleq \frac{1}{n}\sum_1^m (x_i - \bar{x})^2 =$$

$$\frac{1}{n}((x_1-\bar{x})^2 + \sum_2^m (x_i-\bar{x})^2 =$$

$$\frac{1}{n}((n-m+1-\frac{n}{m})^2 + (m-1)(1-\frac{n}{m})^2)$$

This leads to the following definition for inhomogeneity.

Definition 4.56 (inhomogeneity)
Let α be a multiplicity type, and let $\sum\alpha$ and $\bar{\alpha}$ be defined in the natural manner, that is,

$$\sum\alpha \triangleq \sum_{x\in\alpha} x$$

and

$$\bar{\alpha} \triangleq \frac{\sum\alpha}{|\alpha|}$$

Then the inhomogeneity $inh(\alpha)$ of a multiplicity type α equals

$$\frac{1}{C}\sum_{x\in\alpha}(x-\bar{\alpha})^2$$

where the scaling factor C equals

$$(\sum\alpha - |\alpha| + 1 - \bar{\alpha})^2 + (|\alpha|-1)(\bar{\alpha}-1)^2$$

When we speak of the inhomogeneity of a market, we mean the inhomogeneity of its multiplicity type. △

In an assignment market with n agents, m tool types, and multiplicity type α, $n = \sum_{x \in \alpha} x$ and $m = |\alpha|$. Furthermore, the difference between the largest and the smallest element of α attains a maximum of $n - m$ at $\alpha = \{n - m + 1, 1, \ldots, 1\}$. This leads to the following definition of eccentricity.

Definition 4.57 (eccentricity)
The eccentricity ecc(α) *of a multiplicity type* α *equals*

$$\frac{1}{C}(\max \alpha - \min \alpha)$$

where the scaling factor C *equals*

$$\sum_{x \in \alpha} x \; - |\alpha|$$

The eccentricity of a market is the eccentricity of its multiplicity type. △

Without proof, we mention that both these metrics are one-to-one at their extremes, that is, for any $n > m$ and $m > 2$, we can speak of *the* most eccentric multiplicity type within SPAM$^*(n, m)$, and *the* most homogeneous one. Moreover, the most homogeneous multiplicity type is also the least eccentric one, and vice versa. In other words, the two metrics coincide in their extremes. We denote the subset of markets in SPAM$^*(n, m)$ with *minimal eccentricity* (or inhomogeneity) by SPAM$^*(n, m, _)$.

Example 4.58 (Minimal and maximal eccentricity)
Let SPAM$^*(n, m)$ *denote the class of markets* $M \in$ SPAM* *with* n *agents and* m *tool types. Then* $M \in$ SPAM$^*(12, 4)$ *has minimal eccentricity (viz.* 0*) if* $\xi(M) = \{3, 3, 3, 3\}$*, and maximal eccentricity (viz.* $\frac{9-1}{12-4} = 1$*) if* $\xi(M) = \{9, 1, 1, 1\}$*. Likewise, a market in* SPAM$^*(10, 4)$ *has minimal eccentricity (viz.* $\frac{1}{6}$*) if* $\xi(M) = \{3, 3, 2, 2\}$*, and maximal eccentricity (viz. 1) if* $\xi(M) = \{7, 1, 1, 1\}$*.* △

Note that, in general, the minimal eccentricity of assignment markets with n agents and m tool types is 0 if $n \bmod m = 0$, and $\frac{1}{n-m}$ otherwise.

4.6 Estimation of Walrasian Densities in SPAM

We shall present estimates $\tilde{d}(\alpha)$ of the density of Walrasian markets in different subspaces SPAM$^*(\alpha)$ of markets with (a community tool bag of) multiplicity type α. The density $d(\alpha)$ represents the probability that a market which is randomly chosen from SPAM$^*(\alpha)$ turns out to possess a Walrasian equilibrium. As such, the density $d(\alpha)$ can be regarded as the mean value of the stochastic variable W (for Walrasianess) in the sample space SPAM$^*(\alpha)$, where $W = 1$ if a sample market is Walrasian, and $W = 0$ if it is not.

4.6.1 Sampling method

We have computed the estimates $\hat{d}(\alpha)$ of the Walrasian density in SPAM$^*(\alpha)$ by averaging over estimates $\hat{d}(\Gamma, P)$ for various preference profiles P, and a fixed $\Gamma \in \mathfrak{B}_\alpha(R)$. Each of these $\hat{d}(\Gamma, P)$ estimates is the percentage of (viable) problem instances $\langle e, P\rangle$ (in a sample of initial assignments which are drawn randomly such that $\underline{\downarrow} Im\, e = \Gamma$) which turn out to possess a Walrasian equilibrium.

The reason that we compute the estimates $\hat{d}(\alpha)$ by averaging over the estimated densities computed from samples taken from subspaces of SPAM$^*(\alpha)$, instead of simply taking a single sample from SPAM$^*(\alpha)$ is that the latter method tends to render less reliable estimates. In this respect, we have been inspired by the statistical method known as *stratified sampling*(Hoel, 1984). Stratified sampling entails the estimation of the mean $\bar{X}$ of a stochastic variable X in some sample space Y as a *weighted average* of estimates $\hat{m}_i$ for $\bar{X}$ in subspaces Y_i of the sample space, where the weights are proportional to the sizes of the sample subspaces. Formally,

$$\hat{m} = \sum_i \frac{|Y_i|}{|Y|} \hat{m}_i$$

The advantage of stratified sampling lies in the fact that it generally increases the reliability of $\hat{m}$.

Our approach *differs* from stratified sampling in two respects. First, the subspaces SPAM$^*(\Gamma, P)$ from which we draw our samples do not form a partition of the overall space SPAM$^*(\alpha)$. We draw sample problems from the subspaces SPAM$^*(\Gamma, P)$ associated with only some 1000 preference profiles, because the total number of different preference profiles (well over 10^{120} for markets with 12 agents and 5 tool types (cf. Prop 4.39 on page 112)) is far too large to compute an estimate for every subspace.

The second difference between our method and stratified sampling is that we do not weigh the estimates $\hat{d}(\Gamma, P)$ in proportion to the size of the subspace to which they apply. We are indeed dealing with sample subspaces of different size: Prop. 4.44) shows that the size of a subspace SPAM$^*(\Gamma, P)$ for fixed Γ depends on P.

We investigated the impact of subspace size differences on the estimates of $d(n, m, \xi)$ by comparing estimates computed with proportional weighing, with estimates derived with uniform weighing. As illustrated in Fig. 4.7 below, subspace size differences are insignificant for maximally homogeneous multiplicity types, but they can have considerable impact on the density estimates for inhomogeneous multiplicity types.

Nevertheless, we have employed *uniform* instead of proportional weighing in the computation of $\hat{d}(n, m, \alpha)$ from $\hat{d}(\Gamma, P)$. There are two reasons for this. First, the size of SPAM$^*(\Gamma, P)$ is the number of allocations e which are compatible with Γ (i.e., $\underline{\downarrow} Im\, e = \Gamma$), and viable with respect to P. Whereas the number of allocations

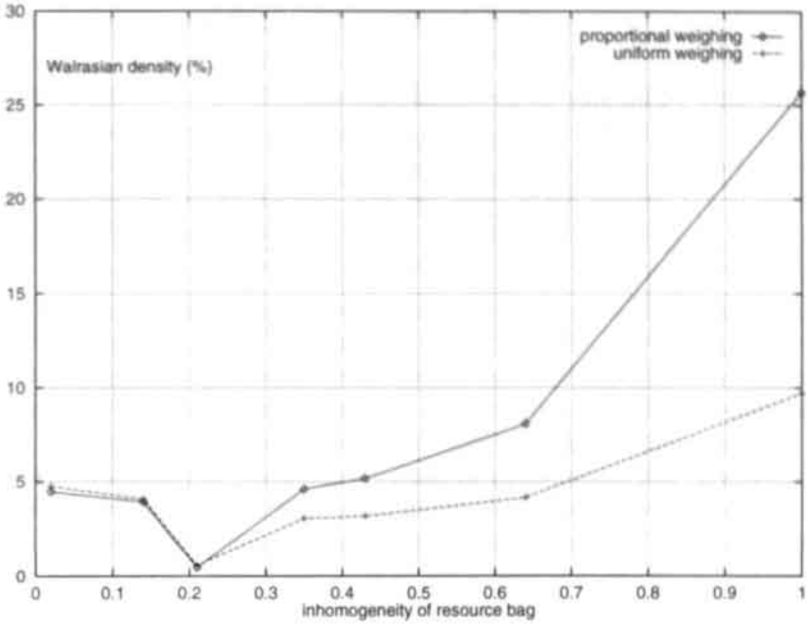


Figure 4.7: Differences in density estimates acquired via different weighing methods ($n = 10, m = 4$).

compatible with a given multiplicity type is easy to compute, we know of no easy way to determine the number of *viable* allocations that comply with α. Of course, we could simply generate these allocations and count them. This is what we did to determine the density with proportional sampling for Fig. 4.7. But this would be a very costly matter for higher values of n and m.

The second reason to use uniform instead of proportional weighing is the following. What we are ultimately interested in, is the probability that a *realistic* reassignment problem possesses a Walrasian equilibrium. We have excluded the least realistic problems (viz. those which are not viable), but among the remaining subspaces, there may still be some that are less likely to occur in practice than others. Hence, the *weight* (in the sense of importance) of $\text{SPAM}^*(\Gamma, P)$ should not equal its size in terms of the number of viable, compatible allocations, but the probability that the profile P and the tool bag Γ are encountered in practice.

One could argue that, in the absence of information about these probabilities, we should make use of the information that we do have (or, at least, can acquire with some effort), namely the size of the sample spaces in terms of the number of associated allocations. We provide a counterargument against this statement in the form of example profiles for which we claim uniform weighing to be *more appropriate* than proportional weighing, even when the sizes of the problem spaces are known.

Example 4.59 (The size of problem spaces)
Consider the profiles $P^1 = [abc,\ abc,\ bac,\ bca]$ *and* $P^2 = [abc,\ acb,\ bac,\ bca]$, *and the allocations* $e^1 = [bcaa]$ *and* $e^2 = [cbaa]$. *The two profiles differ only in the preference of the second agent, and the two allocations in the endowment of the first and the second agent. Note that the two allocations have the same tool bag:* $\Gamma = \underline{\downarrow} \mathit{Im}\, e^1 = \underline{\downarrow} \mathit{Im}\, e^2 = \{a:2, b, c\}$ *of multiplicity type* $\{2,1,1\}$.

If we view profile P^1 *as a* tuple *of preferences* $\langle P_1^1, P_2^1, P_3^1, P_4^1 \rangle$ *(as we have been doing most of the time), then* $M^1 = \langle e^1, P^1 \rangle$ *and* $M^2 = \langle e^2, P^1 \rangle$ *constitute different markets. However, if the endowments are not specified, then all we know of an agent is its preference. Consequently, we can* identify *an agent with its preference. Two agents with the same preference thus become two instances of the same agent. An agent population becomes a* bag *of agents (=preferences). And an allocation becomes a* multifunction[12] *from a bag of agents to a bag of tool types. In this framework, the markets* M^1 *and* M^2 *are identical. In fact, they* do *constitute essentially the same reallocation problem, so regarding them as identical markets seems quite reasonable. This also implies that, in the context of profile* P^1, e^1 *and* e^2 *are one and the same allocation. In the multifunction notation of Def. 2.15, both allocations are denoted by* $\{\langle abc, b\rangle, \langle abc, c\rangle, \langle bac, a\rangle, \langle bca, a\rangle\}$. *In the context of profile* P^2 *however, the allocations* e^1 *and* e^2 *differ, no matter whether we regard allocations as multifunctions or not, because profile* P^2 *consists of four different preferences.*

Summarizing, allocations that differ in the context of profile P^2, *may be identical in the context of* P^1. *As a consequence of this and the fact that profile* P^2 *consists of four different markets, whereas two of the four preferences in* P^1 *are identical, the size (in terms of the number of viable markets) of the respective associated market spaces differ. In this case,* e^1 *and* e^2 *are the only viable allocations for profile* P^2. *Hence, the size of* $\text{SPAM}^*(\{a,a,b,c\}, P^2)$ *equals 2, whereas the size of* $\text{SPAM}^*(\{a,a,b,c\}, P^1)$ *equals 1.* △

As such, when employing proportional sampling, one would attribute twice as much weight to the estimate associated with profile P^2 as to that associated with profile P^1. It seems questionable at the very least whether it is realistic to expect that profile P^2 will occur twice as often in practice as profile P^1. After all, people tend to exhibit at least some degree of unanimity in their preferences. Many people prefer a Mercedes over a Toyota, and a Toyota over a Skoda. Consequently, if the number of agents is large in comparison with the number of tool types, profiles consisting of n different preferences may be relatively rare.[13]

[12] See Sect. 2.2.2 for a definition of multi-functions.

[13] For one thing, they do not even *exist* if $n > m!$.

One could suspect that the reduction factor 2 in the above example is an artefact of the extremely small sizes of the respective problem spaces. To take away such suspicion, we present one additional example.

Example 4.60 (The size of problem spaces (II))
Let S^1 =SPAM$^*(\Gamma, P^1)$ and S^2 =SPAM$^*(\Gamma, P^2)$ be subspaces of SPAM$^*(6,6)$ *with* $\Gamma = \{a, b, c, d, e, f\}$. *Suppose that the profiles share the same bag* $\pi = \{a : 3, b : 3\}$ *of* first *preferences, but differ in the sense that P^1 consists of six* different *preferences, whereas P^2 is a bag with only two different element types* ($P^2 = \{P^2_1 : 3, P^2_2 : 3\}$). *To count the number of viable initial allocations for P^1, we observe that the viability constraint entails that the type-a tool must be allocated to one of the three agents that have type b as their favorite, and vice versa. There are $3 \times 3 = 9$ ways to do this. For each of these, the other four tool types can be distributed freely over the remaining four agents. Thus, the number of viable allocations for profile P^1 equals $9 \times 4! = 216$. Viability with respect to profile P^2 also entails that the type-a tools must be allocated to the type-b fans, and vice versa. However, due to the fact that all the type-a fans are identical (as are all the type-b fans) this represents only one possibility. Similarly, the number of ways to distribute the remaining four tool types over the remaining agents now comes down to the number of ways to split the four tool types in two groups of two. This can be done in $\binom{4}{2} = 6$ ways. Consequently, there are only six different viable allocations for P_2. All in all, the problem space* SPAM$^*(\Gamma, P^1)$ *is 36 times larger than* SPAM$^*(\Gamma, P^2)$. △

In view of the fact that we employ uniform weighing to compute $\hat{d}(\xi)$, the term "density estimate" is somewhat inappropriate. $\hat{d}(\xi)$ does not estimate the density in SPAM$^*(\xi)$, but the average of the densities in its subspaces SPAM$^*(\Gamma, P)$. However, for the sake of conciseness, we choose to use the phrase "density estimate" instead of "estimate of the average of the densities in subspaces SPAM$^*(\Gamma, P)$ of SPAM$^*(\xi)$".

4.6.2 Walrasian density in markets of minimal eccentricity

We now have gathered sufficient insight into the structure of SPAM* to estimate the density of Walrasian markets $\langle I, R, e, P\rangle$ within subspaces of SPAM*. We will present density estimates $\widehat{d}(m, n, \xi)$ for various m, n, and ξ. Here, $n = |I|$ denotes the number of agents, $m = |R|$ the number of tool types, and ξ the multiplicity type of the market.

In view of the low number of potential beneficiaries of a Walrasian auction in a strongly eccentric market, we start out with markets of minimal eccentricity.

Fig. 4.8 shows density estimates $\hat{d}(n, m, \_)$ for $4 \leq n \leq 20$ and $3 \leq m \leq n$. These estimates are the averages of single-profile density estimates $\hat{d}(\Gamma, P)$ over approximately 1000 randomly generated profiles. For $m < 5, n < 15$, each of the single-profile estimates $\hat{d}(\Gamma, P)$ equals the *exact* value of Walrasian density (i.e., the fraction

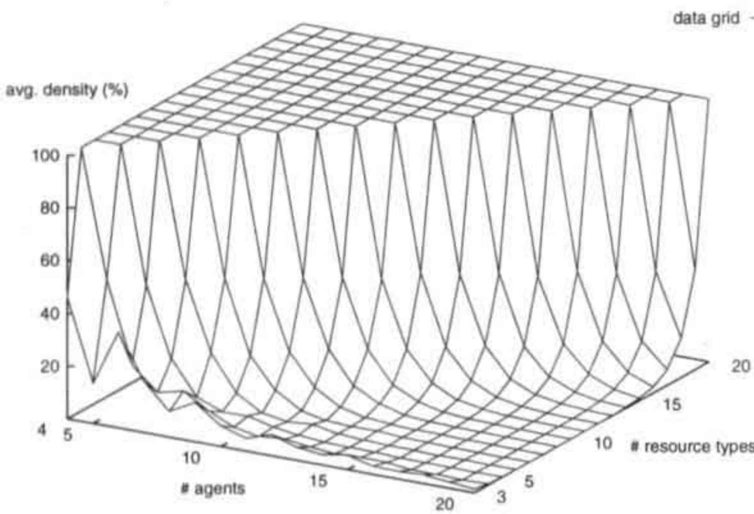


Figure 4.8: Density as a function of problem dimension (for markets of minimal eccentricity).

of the viable allocations that form a Walrasian market with P). For other values of n and m, the density estimates $\hat{d}(\Gamma, P)$ equal the fraction of 1000 randomly generated, viable initial allocations that form a Walrasian market with P.

The grid points of the high plane (100% density at $m > n$) have no meaning. They are plotted only to support the reader in seeing the 3D surface. The plot shows that the density in SPAM$^*(n, m, _)$, which is 100% if $n = m$, decreases rapidly if we decrease m. At very small values of m (the number of tool types), the density tends to rise somewhat again. The same observation can be made if we look at increasing numbers of agents with a fixed number of tools (except for the fact that the density keeps decreasing with rising n). Figure 4.9 offers a different perspective on the very same data. Logarithmic scaling is used to enhance the relief for low densities. Furthermore, the surface is observed from a different point of view. We now stand, as it were, to the right of the tool-types axis in Fig. 4.8, looking toward the origin. From this plot, and its 2D projection on the tool-types/density plane (in Fig. 4.10), we conclude that, if we start out from $n = m$, the density decreases exponentially with decreasing m, (approximately) until we reach the line $n = 2m$. Here the surface levels out, and finally rises somewhat again at low values of m.

The plot in Fig. 4.10 is difficult to read at low values of m. Fig. 4.11 offers a clearer

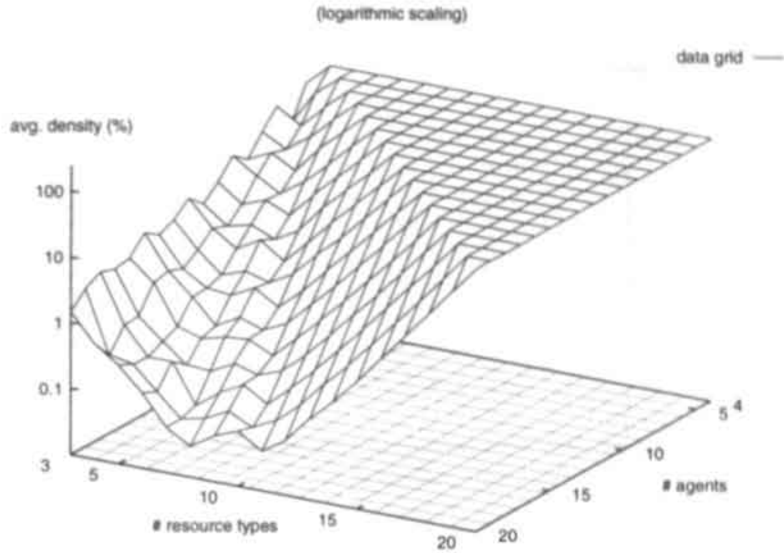


Figure 4.9: An alternate view at the data of Fig. 4.8.

view. Here, we have pictured $d(n, m, \_)$ as a function of n for various m. The plot reveals that the density decreases exponentially with increasing n, especially for relatively large values of m.

The plot in Fig. 4.12 shows density isoclines at 80, 20, 5, 1, 0.5, and 0.1%. From this plot, we can conclude that, beyond $n = 15$, the density drops below 5% for the bulk of values for m. The picture suggests that it is to be expected that, for values $n \gg 20$, finding a Walrasian market with $n \approx 2m$ via random generation of preference profile and initial allocation, is like looking for the proverbial needle in the haystack.

The plots suggest that Walrasian density in large, typed assignment markets (of minimal eccentricity) is too low for a Walrasian auction to be a useful instrument in this context. However, since the plots merely picture *estimates*, we would like to have an impression of the reliability of these. Despite the relatively large number (viz. approximately one million) of markets used for each data point in the plots, the reliability tends to be low, especially for small values of m. This is due to the fact that the estimates of $d(\Gamma, P)$, which form the basis of the estimates $\hat{d}(n, m, \xi)$, generally cover the whole interval between 0 and 100%, and the two extremes are often more frequent than the intermediate values. In fact, inspection of the numerical data revealed that, for $m = 3$, *all* of the estimates $\hat{d}(\Gamma, P)$ were either 0 or 100%.

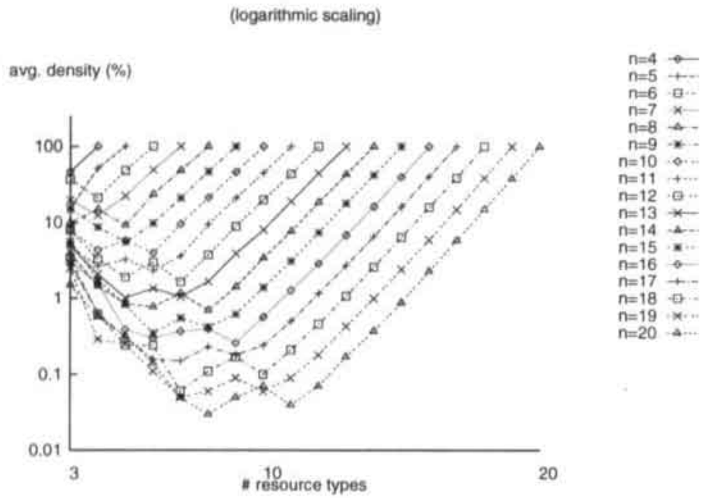


Figure 4.10: Densities $d(n, \cdot, \_)$ for various values of n.

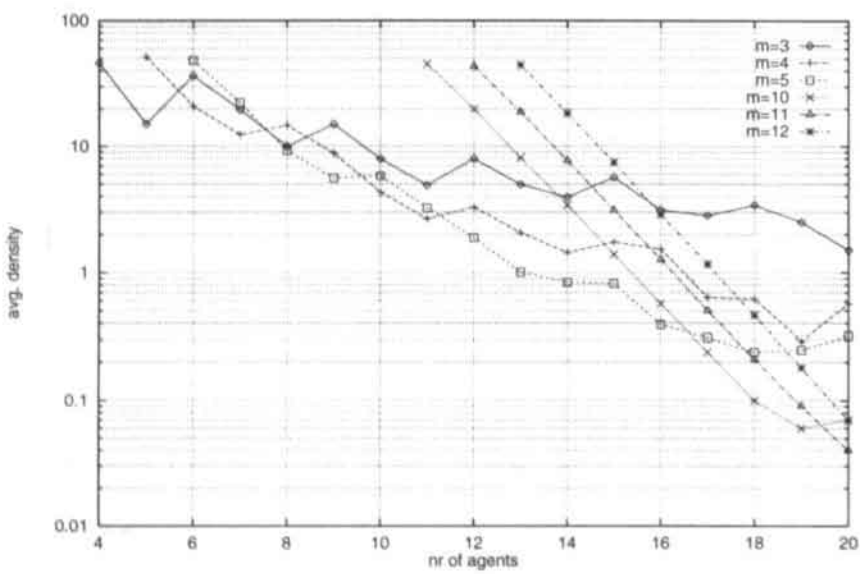


Figure 4.11: Densities $d(\cdot, m, \_)$ for various values of m.

As we will show later (in Prop. 4.62) this is not a coincidence: Within the problem space SPAM$^*(n, 3)$, every density $d(\Gamma, P)$ is either 0% or 100%.

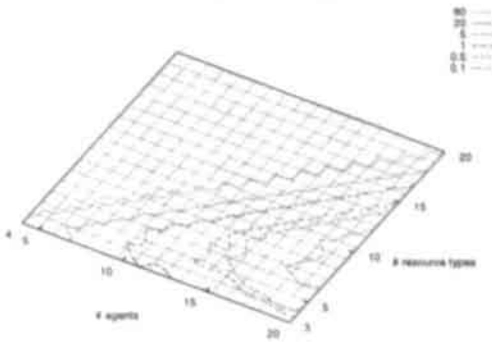


Figure 4.12: Density isoclines associated with Fig. 4.8.

profiles	agents	types	viable	mean	min	max	stdev	none	all
904	5	3	15.11	15.15	0.00	100.00	35.86	767	137
989	5	4	26.20	51.64	0.00	100.00	35.06	54	141
984	10	3	1.93	7.93	0.00	100.00	27.02	906	78
999	10	4	6.45	4.33	0.00	100.00	11.74	360	9
1000	10	5	100.00	5.82	0.00	100.00	10.42	90	5
1000	10	6	100.00	3.99	0.00	100.00	5.95	52	1
1000	10	7	100.00	9.64	0.00	100.00	8.17	1	1
1000	10	8	100.00	21.11	0.90	72.20	12.82	0	0
1000	10	9	100.00	46.14	1.60	86.60	18.97	0	0
992	15	3	100.00	5.65	0.00	100.00	23.08	936	56
1000	15	4	100.00	1.74	0.00	100.00	10.84	924	11
1000	15	5	100.00	0.82	0.00	100.00	5.18	633	2
1000	15	6	100.00	0.35	0.00	13.40	0.97	467	0
1000	15	7	100.00	0.55	0.00	9.20	1.07	298	0
1000	15	8	100.00	0.42	0.00	8.20	0.80	331	0
1000	15	9	100.00	0.62	0.00	6.00	0.71	126	0
1000	15	10	100.00	1.39	0.00	9.60	1.25	35	0
1000	15	11	100.00	3.11	0.00	12.10	2.30	4	0
1000	15	12	100.00	7.44	0.10	27.10	4.59	0	0
1000	15	13	100.00	17.85	1.20	50.00	9.03	0	0
1000	15	14	100.00	41.93	2.90	75.90	15.32	0	0

Table 4.4: Context of density estimates at minimal eccentricity.

Tables 4.4 and 4.5 show some statistics on the data from which the density estimates $\hat{d}(n, m, \_)$ at $m = 5$, and $n = 5, 10, 15, 20$ in Fig. 4.8 were computed. Each line in the table represents data computed from a sample of approximately 1000 preference

profiles	agents	types	viable	mean	min	max	stdev	none	all
997	20	3	100.00	1.50	0.00	100.00	12.17	982	15
1000	20	4	100.00	0.57	0.00	100.00	6.47	958	4
1000	20	5	100.00	0.32	0.00	100.00	4.55	914	2
1000	20	6	100.00	0.14	0.00	100.00	3.17	874	1
1000	20	7	100.00	0.05	0.00	8.00	0.35	875	0
1000	20	8	100.00	0.03	0.00	2.20	0.10	829	0
1000	20	9	100.00	0.05	0.00	4.60	0.22	773	0
1000	20	10	100.00	0.07	0.00	3.60	0.19	685	0
1000	20	11	100.00	0.04	0.00	0.60	0.08	770	0
1000	20	12	100.00	0.07	0.00	1.20	0.12	585	0
1000	20	13	100.00	0.17	0.00	1.70	0.21	355	0
1000	20	14	100.00	0.37	0.00	2.40	0.37	151	0
1000	20	15	100.00	0.87	0.00	5.60	0.70	44	0
1000	20	16	100.00	2.27	0.00	9.40	1.54	6	0
1000	20	17	100.00	5.84	0.20	21.30	3.40	0	0
1000	20	18	100.00	14.82	1.40	41.00	7.45	0	0
1000	20	19	100.00	37.76	4.30	72.30	13.33	0	0

Table 4.5: Some context of density estimates at minimal eccentricity(continued).

profiles. For $m = 3, 4$ and $n < 15$, the density estimates are the average of the *exact* values of the single-profile densities (i.e., the fraction of *all* viable initial allocations that turned out to constitute a Walrasian market in combination with the profile). We originally planned to perform this procedure for all values of n and m, but this is not feasible due to the large numbers of viable allocations at higher values for n and m.[14] The other density estimates are therefore based on *estimated* single-profile densities, that is, the 'Walrasian fraction' in a random sample of 1000 viable initial allocations. For low values of n and m (e.g., $n = 6; m = 5$), the number of different viable allocations can be less than or close to 1000, so the allocation samples may contain some allocations more than once.

The column headings of Table 4.4 should be interpreted as follows:

profiles: the number of profiles on the basis of which an average density was computed;[15]

agents: the number of agents;

types: the number of tool types;

[14]For $n = 20; m = 19$ the maximum number of viable initial allocations for a profile approximately equals 10^{17}.

[15]This number equals 1000 minus the number of generated profiles for which no viable allocation appeared to exist.

viable: the average percentage of allocations that formed a viable market together with the profile;[16]

mean: the average Walrasian density;

min: the minimum Walrasian density in the sample;

max: the maximum Walrasian density in the sample;

stdev: the standard deviation in the sample;

none: the number of profiles for which *none* of the viable initial allocations turned out to be 'Walrasian';

all: the number of profiles for which *all* of the viable initial allocations turned out to be 'Walrasian'.

Several interesting observations can be made concerning Tables 4.4 and 4.5. First, the sample standard deviations in the 8th column (pertaining to samples of approximately 1000 estimates $\hat{d}(\Gamma, P)$ for fixed Γ and varying P) are high indeed, in comparison with the associated sample means in the fifth column.

The values in the 6th and 7th column (min and max, respectively denoting the smallest and largest density estimate in the sample) provide an explanation for this phenomenon, in combination with the values in the last two columns (none and all, respectively denoting for how many profiles the extremes min and max were reached). It appears that the high standard deviations occur in those cases in which the density estimates range over a large portion of the interval $[0, 100]$ with a strong predilection for the extremal points of the range. The lack of resemblance between the sample distributions and a normal distribution implies that the standard deviation estimates cannot be used to compute confidence intervals for the estimates of the mean.

In principle, there are various ways to overcome this problem. We could redo the entire computation, computing median instead of mean densities and use nonparametric methods to estimate the reliability of the medians. Or we could repeat the computation of mean density estimates N times to acquire a near-normally distributed sample of mean estimates. However, both of these procedures would be extremely costly,[17] and we do not really *need* greater precision to draw a meaningful conclusion. The observation that the *highest* estimates of local Walrasian density over 1000 different profiles for $n = 20$, $7 < m < 17$ are all below 10% is sufficient to conclude that Walrasian equilibria are often absent in typed assignment markets. Consequently, we have confined our supplementary experimentation to repetitive computation of mean density estimates, for a *small* number of problem dimensions.

[16] A percentage of 100 indicates that we simply generated 1000 viable allocations randomly, instead of *all* of the viable allocations for the profile.

[17] It already took about a week of processing on 5 (shared) workstations to generate the data for the plots in Figures 4.8–4.12.

The results are shown as histograms in Figures 4.13, 4.14, and 4.15. From the data, we can estimate $d(n, 5, \_)$ for $n = 10, 15, 20$ with greater accuracy, leading to estimates of 6.190%, 0.911%, and 0.243%, with (95%) confidence intervals[18] of $[6.14, 6.24]$, $[0.88, 0.94]$, and $[0.22, 0.26]$ respectively.

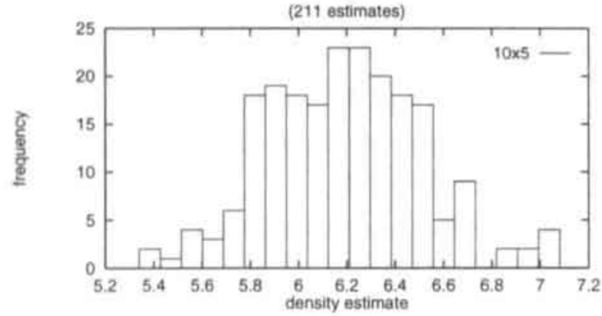


Figure 4.13: Histogram of 211 estimates of $d(10, 5, \_)$ in Fig. 4.8.

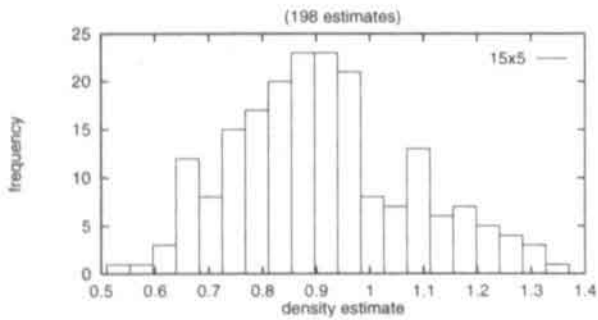


Figure 4.14: Histogram of 198 estimates of $d(15, 5, \_)$ in Fig. 4.8.

4.6.3 Walrasian density as a function of eccentricity

It is not apparent that the estimates of Walrasian densities in $\text{SPAM}^*(n, m, \_)$ are representative for the density in subspaces $\text{SPAM}^*(n, m, \xi)$ for multiplicity types ξ other than those of minimal eccentricity. Therefore, we investigate how the density estimates vary with the eccentricity (or inhomogeneity) of the markets.

Table 4.6 enumerates all of the 13 multiplicity types in $\text{SPAM}^*(12, 4)$, together with the respective values for inhomogeneity and eccentricity. In this case, the

[18] The confidence limits stem from a table of the Student t distribution with 200 degrees of freedom.

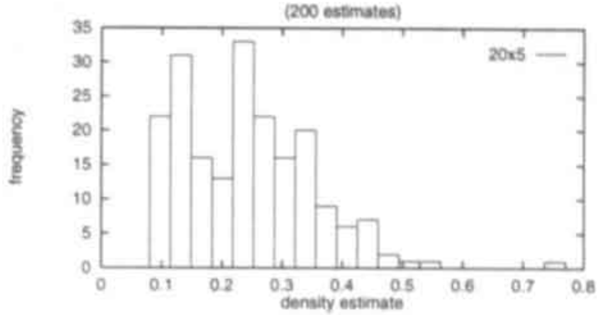


Figure 4.15: Histogram of 200 estimates of $d(20, 5, \_)$ in Fig. 4.8.

bag	card. type	inhomogeneity	eccentricity	avg. density
aaabbbcccddd	$\{3,3,3,3\}$	0.000	0.000	4.20
aaaabbbcccdd	$\{4,3,3,2\}$	0.014	0.250	2.09
aaaabbbbccdd	$\{4,4,2,2\}$	0.028	0.250	1.34
aaaabbbbcccd	$\{4,4,3,1\}$	0.042	0.375	0.86
aaaaabbbcccd	$\{5,3,3,1\}$	0.056	0.500	0.85
aaaaabbbbccd	$\{5,4,2,1\}$	0.069	0.500	0.06
aaaaaabbccdd	$\{6,2,2,2\}$	0.083	0.500	2.62
aaaaabbbbbcd	$\{5,5,1,1\}$	0.111	0.500	3.23
aaaaaabbbccd	$\{6,3,2,1\}$	0.097	0.625	0.38
aaaaaabbbbcd	$\{6,4,1,1\}$	0.125	0.625	2.45
aaaaaaabbbcd	$\{7,3,1,1\}$	0.167	0.750	1.47
aaaaaaaabbcd	$\{8,2,2,1\}$	0.236	0.875	2.00
aaaaaaaaabcd	$\{9,1,1,1\}$	0.333	1.000	6.27

Table 4.6: Numerical representation of imbalance and densities.

inhomogeneity is an injection, mapping different multiplicity types to different inhomogeneities. This is not the case for the eccentricity. There exist, for example, four different multiplicity types with an eccentricity of 0.500. To produce a plot of average density as a function of eccentricity, the density values in the above table are averaged, leading to an average density for the subspace of markets with $\mathrm{ecc}(\xi) = 0.500$ of $\frac{0.85+0.06+2.62+3.23}{4} = 1.69$.

The plots in Figures 4.16, 4.17, and 4.18 have been derived in this manner. The first two of these picture the average density in subspaces of SPAM$^*(12, 4)$ as a function of inhomogeneity and eccentricity, respectively. The curve in Fig. 4.16 is based on 13 values, one for each multiplicity type. The curve in Fig. 4.17 is only based on 8 values, some of which correspond with one, others with an average over several

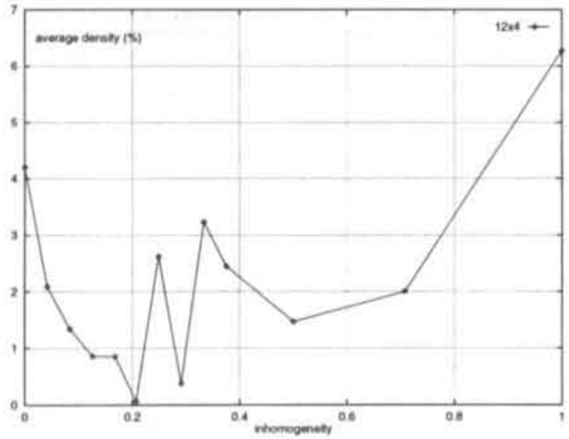


Figure 4.16: Walrasian density as a function of inhomogeneity.

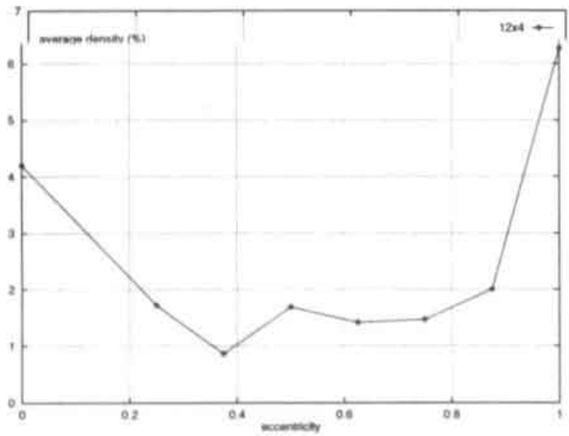


Figure 4.17: Walrasian density as a function of eccentricity.

multiplicity types. The same curve is also drawn in Fig. 4.18 (on page 143), next to similar curves for other population sizes.

Comparing Figures 4.16 and 4.17, we notice that the inhomogeneity curve is more erratic than the eccentricity curve, which can largely be attributed to the averaging that has taken place in computing the latter. However, both curves start out with a relatively high leftmost density value (at minimal eccentricity/inhomogeneity), somewhat lower values for the intermediate estimates, and reach their maximum density at maximal eccentricity.

As Fig. 4.18 shows, the same holds for curves associated with other problem dimen-

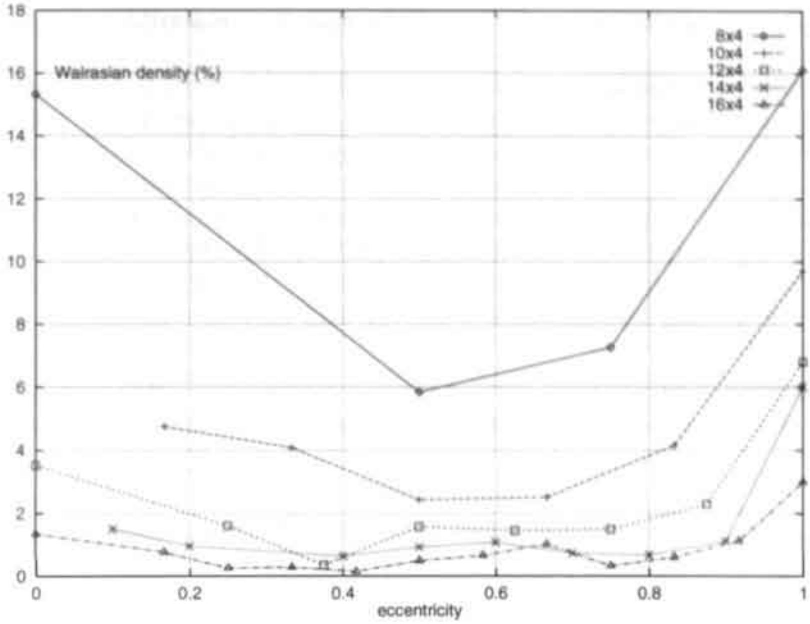


Figure 4.18: Density as a function of eccentricity
(for various problem dimensions).

sions (i.e., number of agents/tool types). There are two reasons to disregard the maxima at $\mathrm{ecc}(\xi) = 1$. First, as we contended in the beginning of this section, the associated Walrasian equilibria tend to be of low quality in terms of the number of agents that profit from Walrasian reallocation. Second, the associated subspaces $\mathrm{SPAM}^*(\xi)$ are very small in comparison with those associated with other eccentricity values (cf., Table 4.2 on page 123). Consequently, it seems legitimate to consider the density estimates $d(n, m, \_)$ (i.e., those plotted in Fig. 4.8) as *optimistic* estimates for $d(n, m)$.

4.6.4 The special case of markets with three tool types

The subspace $\mathrm{SPAM}^*(n, 3)$ of markets involving three tool types differs in various aspects from $\mathrm{SPAM}^*(n, m)$ with $m > 3$. The propositions below pertain to the following special characteristics of $\mathrm{SPAM}^*(n, 3)$.

1. Every Walrasian market in $\mathrm{SPAM}^*(n, 3)$ possesses exactly one stratification.

2. The Walrasian densities in subspaces $\mathrm{SPAM}^*(\Gamma, P)$ of $\mathrm{SPAM}^*(n, 3)$ are either 0% or 100%.

3. Theoretical derivation of a formula for $d(3k, 3, \_)$ is feasible.

With respect to the latter aspect, we remark that, *in principle*, it is possible to derive a formula for $d(n, m, \xi)$ for *any* n, m, and ξ. However, the combinatorial nature of such derivations effectively prohibits the derivation of formulas for large values of m. In fact, for any value of m that exceeds 3, derivation of the associated formula is very complicated. Hence, we only present (in Prop. 4.64 below) a formula for $d(n, m, \xi)$ with $n = 3k, m = 3$, and $\xi = \{k, k, k\}$. Towards the end of this section, we use this formula to compare the *estimates* for $d(3k, 3, \{k, k, k\})$ (produced by the density estimation software) for $k = 2, \ldots, 7$ with the *true* (theoretical) values.

Proposition 4.61
Every Walrasian market in SPAM*$(n, 3)$ *possesses exactly one stratification.*

Proof.
The top stratum of a *viable* stratifiable market must contain at least two tool types. Hence, a stratifiable market involving three tool types (i.e., $|\check{\Gamma}| = 3$) can possess the following stratifications.

$$\begin{aligned} S_1 &= \{a, b, c\}| \\ S_2 &= \{a, b\}|\{c\} \\ S_3 &= \{a, c\}|\{b\} \\ S_4 &= \{c, b\}|\{a\} \end{aligned}$$

We must prove that no market in SPAM*$(n, 3)$ can possess more than one of these stratifications. Let S_M denote the set of stratifications of the market $M \in$ SPAM*$(n, 3)$. Then (because of symmetry considerations with respect to the alphabetic variants S_1, S_2, and S_3) it suffices to show that

(i) $S_1 \in S_M \Rightarrow S_2 \notin S_M$

(ii) $S_2 \in S_M \Rightarrow S_1 \notin S_M$

(iii) $S_2 \in S_M \Rightarrow S_3 \notin S_M$

case (i): For $\{a, b, c\}|$ to be a stratification of M, {a,b,c} must be an equilibrial market (cf., Def. 4.24 and 4.29). For a *viable* market M, this implies that, among the agents endowed with a type-a, or a type-b tool, there must be $\Gamma(c) \geq 1$ agents with a first preference for tool type c. In the segmentation $\{a, b\}|\{c\}$ at least one agent in the top segment would thus have a first preference for a tool type of a lower segment. Hence, by Def. 4.24, the segmentation $S_2 = \{a, b\}|\{c\}$ is not a stratification of M.

case (ii): For $\{a,b\}|\{c\}$ to be a stratification of the *viable* market M, *each* of the agents endowed with a type-a tool must have a first preference for a type-b tool, and vice versa. Hence, in the market as a whole, there is no agent with tool type c as its first preference. This implies that M is not an equilibrial market, and, hence, $\{a,b,c\}|$ is not a stratification of M.

case (iii): Again, for $\{a,b\}|\{c\}$ to be a stratification of the *viable* market M, *each* of the agents endowed with a type-a tool must have a first preference for a type-b tool, and vice versa. Hence, in the segmentation $\{c,b\}|\{a\}$, each of the agents endowed with a type-b tool (i.e., a tool type in the top segment) has a first preference for a tool type (i.e., type a) in a lower segment. Hence, by Def. 4.24, the segmentation $S_3 = \{c,b\}|\{a\}$ is not a stratification of M.

■

Proposition 4.62
Let P, Γ be a preference profile and a tool bag which involve three tool types (i.e., $\text{SPAM}^*(\Gamma,P) \subset \text{SPAM}^*(n,3)$). Then $d(\Gamma,P) = 0\% \vee d(\Gamma,P) = 100\%$.

As in the proof of Prop. 4.61, we argue that there exist four possible stratifications for a Walrasian market in $\text{SPAM}^*(n,3)$.

$$\begin{aligned} S_1 &= \{a,b,c\}| \\ S_2 &= \{a,b\}|\{c\} \\ S_3 &= \{a,c\}|\{b\} \\ S_4 &= \{c,b\}|\{a\} \end{aligned}$$

To prove the proposition, its suffices to show that, in the context of each of these stratifications

$$\begin{aligned} &(\exists e)\ (\ \langle e,P\rangle \in \text{SPAM}^*(\Gamma,P)\ \wedge\ \langle e,P\rangle \text{ is Walrasian }) \Rightarrow \\ &(\forall e)\ (\ \langle e,P\rangle \in \text{SPAM}^*(\Gamma,P)\ \Rightarrow\ \langle e,P\rangle \text{ is Walrasian }) \end{aligned} \qquad (\dagger)$$

We prove this for S_1 and S_2. Proofs for the other cases (S_3 and S_4) are essentially the same as that for S_2.
In the following, π denotes the bag of first preferences in P.

S_1: The market $\langle e,P\rangle$ is equilibrial iff $(\forall r \in \{a,b,c\})\ \pi(r) = \Gamma(r)$. In other words, whether $\{a,b,c\}|$ is a stratification of $\langle e,P\rangle$ depends only on π and $\Gamma = \underline{\downarrow}\, \mathit{Im}\, e$; not on e itself. Hence, $(\dagger)$ holds.

S_2: The segmentation $\{a,b\}|\{c\}$ is a stratification for a *viable* market $\langle e,P\rangle$ iff all of the agents endowed with a type-a tool have a first preference for type b, and vice versa. $(\ddagger)$
Because of the viability of e, this implies that $\pi(c) = 0$. Let e_2 be an arbitrary P-viable allocation such that $\underline{\downarrow}\, \mathit{Im}\, e_2 = \Gamma = \underline{\downarrow}\, \mathit{Im}\, e$. Then it follows from

$\pi(c) = 0$ and the viability of e_2 that all of the agents endowed (under e_2) with a type-a tool have a first preference for type b, and vice versa. This implies (by (‡)), that $\{a,b\}|\{c\}$ is a stratification of $\langle e_2, P\rangle$. Hence, $\langle e_2, P\rangle$ is a Walrasian market. ■

Corollary 4.63
The probability distribution of $\hat{d}(n,3,\xi)$ is the binomial distribution with parameter $p = d(n,3,\xi)$.[19]

Proposition 4.64
The density in the subspace SPAM*$(3k, 3, \_)$ *equals*

$$d(3k, 3, \{k:3\}) = \frac{1}{2^{3k}} \sum_{l=0}^{k} \binom{k}{l}^3 + \frac{3}{2^{2k}}$$

Proof.
For a (viable) market with three tool types (say, $R = \{a,b,c\}$), the only possible stratifications are $\{a,b,c\}|$, $\{a,b\}|\{c\}$, $\{a,c\}|\{b\}$, and $\{b,c\}|\{a\}$. In other words, either the market is equilibrial, or it contains an equilibrial segment of two tool types. By Prop. 4.61, the four cases ($\{a,b,c\}|$, $\{a,b\}|\{c\}$, $\{a,c\}|\{b\}$ and $\{b,c\}|\{a\}$) are nonoverlapping, in the sense that there exist no viable market that complies with more than one of the stratifications. Furthermore, the latter two of these stratifications are alphabetic variants of $\{a,b\}|\{c\}$. Hence, $d(n,m,\_) = P(\{a,b,c\}|) + 3 \cdot P(\{a,b\}|\{c\})$. Here, $P(X)$ denotes the probability that X happens to be a stratification for a market that is chosen randomly from SPAM*$(3k, 3, \{k,k,k\})$. A market or market segment is equilibrial iff its demand bag (i.e., the bag π of first preferences of the associated agents) equals its supply bag (i.e., the tool bag of the market (segment)). In the current case, this leads to the following stratification constraints.

For $\{a,b,c\}|$:	$(\forall r \in \{a,b,c\})\ \pi(r) = \Gamma(r) = k$
For $\{a,b\}|\{c\}|$:	$(\forall r \in \{a,b\})\ \ \pi(r) = \Gamma(r) = k$

Consequently, we can compute the probabilities $P(X)$ as fractions of the number of viable profile heads that satisfy the stratification constraint associated with X divided by the total number of viable profile heads. Here, a profile head is a function $h: I \to R$, such that $h(i) = \max_{P_i}$. Hence, the demand bag π equals $\downarrow Im\, h$. For a profile head to be viable as well as compatible with the stratification $\{a,b\}|\{c\}$, the first preference of the a-possessors must be b, and that of the b-possessors must be a. Only for the c-possessors, there is more than one option for their first preference, namely either a or b. Since there are k c-possessors, this comes down to

[19] Here, the densities should be thought of as real numbers x with $0 \le x \le 1$, rather than percentages between 0 and 100.

2^k possibilities to choose a function h that is viable and stratification-compatible. To comply with the viability constraint only, it suffices that none of the agents have a first preference for their own tool type. This amounts to freedom of choice between two tool types for any of the $3k$ agents. Hence,

$$P(\{a,b\}|\{c\}) = \frac{2^k}{2^{3k}} = \frac{1}{2^{2k}} \tag{4.25}$$

The expression for $P(\{a,b,c\}|)$ obviously has the same denominator 2^{3k}, but the numerator is more complex, due to the fact that we have to comply with viability as well as stratifiability. Free choice is now restricted to:

1. Choose l a-preferences among the k b-owners ($0 \leq l \leq k$).
   (This determines the preferences of the other $k - l$ b-owners to be c, and the number of a-preferences among c-owners to equal $k - l$.)
2. Choose $k - l$ a-preferences among the k c-owners.
   (This determines the preferences of the other l c-owners to be b, implying that we must choose $k - l$ b-preferences among a-owners.)
3. Choose $k - l$ b-preferences among the k a-owners.
   (This determines preferences of the l other a-owners to be c.)

This leads to the formula

$$P(\{a,b,c\}|) = \frac{1}{2^{3k}} \sum_{l=0}^{k} \binom{k}{l} \binom{k}{k-l}^2$$

Since $\binom{k}{l} = \binom{k}{k-l}$,

$$P(\{a,b,c\}|) = \frac{1}{2^{3k}} \sum_{l=0}^{k} \binom{k}{l}^3$$

which, in combination with Eq. 4.25, renders

$$d(3k,3,\{k,k,k\}) = \frac{1}{2^{3k}} \sum_{l=0}^{k} \binom{k}{l}^3 + \frac{3}{2^{2k}}$$

■

Table 4.7 below features a comparison of the *density estimates* for $d(3k,3,\{k,k,k\})$ (produced by the density estimation software) for $k = 2,\ldots,7$ with the *true* (theoretical) values, acquired via the formula in Prop. 4.64. Recall that, while estimating densities, we approximate in three respects.

1. Estimates of $d(n,m,\_)$ are based on estimates $\hat{d}(\Gamma,P)$ for only 1000 profiles.

2. Estimates for $n \geq 15$ or $m \geq 5$ of $d(\Gamma, P)$ are based on only 1000 viable allocations.

3. In averaging over $\hat{d}(\Gamma, P)$, we employed *uniform* weighing instead of weights proportional to the size (in terms of number of viable allocations) of the subspaces SPAM$^*(\Gamma, P)$.

In view of these approximations and the rather large sample variances (cf., Table 4.4), we would expect considerable differences between the estimates and the true values. According to Table 4.7 below, this is not really so. The estimates turn out to be better than expected, especially for small values of n (which were associated with large variances in Table 4.4). In the context of a Walrasian (or other kind of) auction, an equilibrial SPAM constitutes a trivial reallocation problem, in the sense that an optimal allocation can be arrived at simply by having each agent release its tool into one big community bag, and subsequently allowing each agent to pick a tool of its favorite type from the bag. In the third column of Table 4.7, we added the percentage of Walrasian markets that are equilibrial, that is, possess a stratification $\{a, b, c\}|$. It turns out that this percentage rises rapidly toward 100% with n. In other words, for large k, nearly all of the reallocation problems (in the class SPAM$^*(3k, 3, _)$ which possess a Walrasian equilibrium are *trivial* problems. Of course, it is not necessarily so that the same pertains to markets in the other subclasses of SPAM*.

n	estimated density	true density	trivial markets
6	36.43%	34.37%	45.45%
9	15.05%	15.63%	70%
12	7.98%	9.62%	87.81%
15	5.65%	7.17%	95.91%
18	3.41%	5.78%	98.73%
21	3.51%	5.02%	99.63%

Table 4.7: Comparison between experimental and theoretical values for densities involving markets with 3 tool types.

However, experimental data suggest that something similar does hold for 'uniformly Walrasian' subspaces SPAM$^*(\Gamma, P)$ of SPAM$^*(\xi)$ for $m = |\breve{\xi}| > 3$ (i.e., subspaces that consist entirely of Walrasian markets). Checks performed during estimation of $d(10, 4, \xi)$ and $d(12, 4, \xi)$ provided us with estimates of the percentage of cases in which uniformly Walrasian subspaces SPAM$^*(\Gamma, P)$ consist entirely of trivial problems (i.e., equilibrial markets). These estimates are 95% for markets involving 10 agents and 4 tool types, and 94% for markets with 12 agents and 4 tool types.

4.7 Walrasian Density in AM\SPAM

In the previous section, we gave estimates for the density of Walrasian markets in assignment markets where all preferences are strict. This invokes the question 'And what if they're not...?'.

If some (or all) preferences are weak, the density of Walrasian markets is higher than described by the estimates for SPAMs. In the extreme case that each of the agents is totally indifferent with respect to the tool type that is allocated to it, the density equals one. This case, however, does not constitute a problem: Any initial endowment is a Walrasian allocation. For less extreme cases, the observations below give an impression of what is to be expected in AM\SPAM.

Definition 4.65 (SPAM enumeration)
Let $M = \langle I, R, e, P\rangle$ be a market in AM\SPAM, that is, $P = \{\succsim_i\}_i$ contains one or more weak preferences. Then the SPAM enumeration M^* of M is the set of all SPAMs $\langle I, R, e, P'\rangle$ such that

1. $(\forall i \in I)\, P'(i)$ *is a strict preference* $>_i$ *on* R
2. $(\forall i \in I)\, P'(i)$ *is compatible with* $P(i)$, *that is*
   $(\forall r_1, r_2 \in R)\, r_1 >_i r_2 \Rightarrow r_2 \not\succsim_i r_1$

△

Example 4.66 (SPAM enumeration)
The SPAM enumeration of $M = [\underline{b}.ac,\ b.\underline{a}.c,\ ab\underline{c},\ c.\underline{b}.a]$ *equals*

$$
\begin{aligned}
M^* = \{&[\underline{b}.a.c,\ b.\underline{a}.c,\ a.b.\underline{c},\ c.\underline{b}.a],\\
&[\underline{b}.c.a,\ b.\underline{a}.c,\ a.b.\underline{c},\ c.\underline{b}.a],\\
&[\underline{b}.a.c,\ b.\underline{a}.c,\ a.\underline{c}.b,\ c.\underline{b}.a],\\
&[\underline{b}.c.a,\ b.\underline{a}.c,\ a.\underline{c}.b,\ c.\underline{b}.a],\\
&[\underline{b}.a.c,\ b.\underline{a}.c,\ b.\underline{c}.a,\ c.\underline{b}.a],\\
&[\underline{b}.c.a,\ b.\underline{a}.c,\ b.\underline{c}.a,\ c.\underline{b}.a],\\
&[\underline{b}.a.c,\ b.\underline{a}.c,\ b.a.\underline{c},\ c.\underline{b}.a],\\
&[\underline{b}.c.a,\ b.\underline{a}.c,\ b.a.\underline{c},\ c.\underline{b}.a],\\
&[\underline{b}.a.c,\ b.\underline{a}.c,\ \underline{c}.a.b,\ c.\underline{b}.a],\\
&[\underline{b}.c.a,\ b.\underline{a}.c,\ \underline{c}.a.b,\ c.\underline{b}.a],\\
&[\underline{b}.a.c,\ b.\underline{a}.c,\ \underline{c}.b.a,\ c.\underline{b}.a],\\
&[\underline{b}.c.a,\ b.\underline{a}.c,\ \underline{c}.b.a,\ c.\underline{b}.a]\}
\end{aligned}
$$

△

Proposition 4.67
An assignment market with one or more weak preferences is Walrasian $\Leftrightarrow$ its SPAM enumeration contains a Walrasian market.

Proof.
$\Rightarrow$:
Let $M = \langle I, R, e, P \rangle \in$ AM\SPAM, with $M^* = \{\langle I, R, e, P^j \rangle\}_{j \in K}$. We prove the $\Rightarrow$-part by showing that if $\langle w, \succsim \rangle$ is a Walrasian equilibrium of M, then there exists a profile $P' \in \{P^j\}_{j \in K}$ such that $\langle w, \succsim \rangle$ is a Walrasian equilibrium of $\langle I, R, e, P' \rangle$.
Let, for $Q \subseteq R^n$, $\text{best}_{\succsim}(Q)$ denote the tuple $\langle \text{best}_{\succsim_1}(Q_1), \ldots, \text{best}_{\succsim_n}(Q_n) \rangle$. Define $\max_{\succ}(Q)$ analogously. Then, by definition of M^*,

$$(\forall Q \subseteq R^n) \ \underset{\succsim}{\text{best}}(Q) = \bigcup_{j \in K} \{\max_{P^j}(Q)\} \tag{4.26}$$

If $\langle w, \succsim \rangle$ is a Walrasian equilibrium of M, then

$$(\forall i \in I) \, w(i) \in \underset{\succsim_i}{\text{best}}(\{r \in R \mid r \sim e(i)\}$$

Hence, $w \in \text{best}_{\succsim}(\{r \in R^n \mid (\forall i \in I) \ r_i \sim e(i)\}$. Consequently, by (4.26),

$$(\exists j \in K) \, w = \max_{P^j}(\{r \in R^n \mid (\forall i \in I) \ r_i \sim e(i)\}$$

Let $j_0 \in K$ be such that $w = \max_{P^j}(\{r \in R^n \mid (\forall i \in I) \ r_i \sim e(i)\}$ Then $\langle w, \succsim \rangle$ is a Walrasian equilibrium of $\langle I, R, e, P^{j_0} \rangle$.

The converse ($\Leftarrow$) part is completely analogous, and therefore left to the reader.

■

The statements above may tempt the reader to believe that a Walrasian auction may be applicable to many assignment markets after all (viz. those in AM\SPAM). This is not so. Even though Walrasian markets generally occur more frequently in this class, two remarks deserve to be made.

First, some of the indifferences do not count. Since Walrasian allocations are individually rational (no agent prefers its initial endowment over its newly allocated tool), any indifference between tool types that are less desirable to an agent than its initial allocation is irrelevant.

As such, the SPAM enumeration in Ex. 4.66 can be pruned into

$M^* = \{[\underline{\text{b}}.\text{a.c}, \text{b}.\underline{\text{a}}.\text{c}, \text{a.b}.\underline{\text{c}}, \text{c}.\underline{\text{b}}.\text{a}],$
$[\underline{\text{b}}.\text{a.c}, \text{b}.\underline{\text{a}}.\text{c}, \text{a}.\underline{\text{c}}.\text{b}, \text{c}.\underline{\text{b}}.\text{a}],$
$[\underline{\text{b}}.\text{a.c}, \text{b}.\underline{\text{a}}.\text{c}, \text{b}.\underline{\text{c}}.\text{a}, \text{c}.\underline{\text{b}}.\text{a}],$
$[\underline{\text{b}}.\text{a.c}, \text{b}.\underline{\text{a}}.\text{c}, \text{b.a}.\underline{\text{c}}, \text{c}.\underline{\text{b}}.\text{a}],$
$[\underline{\text{b}}.\text{a.c}, \text{b}.\underline{\text{a}}.\text{c}, \underline{\text{c}}.\text{a.b}, \text{c}.\underline{\text{b}}.\text{a}],$
$[\underline{\text{b}}.\text{a.c}, \text{b}.\underline{\text{a}}.\text{c}, \underline{\text{c}}.\text{b.a}, \text{c}.\underline{\text{b}}.\text{a}]\}$

Second, even if a Walrasian equilibrium exists, it may be difficult to *reach* it in a Walrasian auction in case of weak preferences.[20] Especially in cases when many agents have weak preferences (leading to large SPAM enumerations) the agent community may fail to clear the market even if the auctioneer sets prices according to a Walrasian price assignment. The nondeterminism in the response of agents with an indifference between most preferred tools in their budget set can, on a community level, be described as "the community *randomly* chooses one element from the SPAM enumeration". As the following example illustrates, this turns the Walrasian auction into a process similar to a Bernouilli experiment. For the sake of simplicity, the market in the example is an untyped assignment market, but this is not essential.

Example 4.68 (The Bernouilli effect of indifference)
Let M be a market in UAM\SPAM *in which every agent is dissatisfied with its initial endowment but indifferent between any of the other tool types. An example of such a market is*

$$M = [\text{bcdef}.\underline{a}, \text{acdef}.\underline{b}, \text{abdef}.\underline{c}, \text{abcef}.\underline{d}, \text{abcdf}.\underline{e}, \text{abcde}.\underline{f}] \tag{4.27}$$

Since $M \in \text{UAM}$, *M is a Walrasian market. Moreover, the indifference between any two tool types in* $R\backslash\{e(i)\}$ *implies that it is an equilibrial market: Any allocation* $w \in W_e \triangleq \{w \in \text{PERM}(R) \mid w(i) \neq e(i)\}$ *is Walrasian under the price preorder* $\succsim$ *defined by* $r_1 \sim r_2 \sim \cdots \sim r_n$. *In fact,* $W_M(\succsim) = W_e$.
However, despite the abundance of Walrasian allocations under this price ordering, the probability p_0 *of arriving at a Walrasian allocation when the auctioneer broadcasts all prices to be equal is low: In response to the price preorder* $\succsim$, *each of the agents will randomly select a bid* x_i *from* $\text{best}_{\succsim_i}(R) = R\backslash\{e(i)\}$. *Hence, the number of possible community responses (i.e., proposed allocations)* $\langle x_1, \ldots, x_n\rangle$ *equals* $(n-1)^n$, *where* $n = |I| = |R|$. *The number of responses that lead to market clearance, that is, a Walrasian equilibrium state equals* $|W_e| = (n-1)!$. *Consequently,*

$$p_0 = \frac{(n-1)!}{(n-1)^n}$$

For the example market M in (4.27), this amounts to $\frac{5!}{5^6} = 0.008$. △

The example describes a worst-case scenario, in the sense that other price preorders compare favorably with $\succsim$. If, for instance, the auctioneer happens to set prices as $r_1 \sim r_2 > r_3 \sim r_4 > \cdots > r_{n-1} \sim r_n$ (i.e., pairwise price equality), then $p_0 = 1$. The point is, however, that the auctioneer has no way of knowing that such a price preorder will work. It could, of course, apply the Top-Trading-Cycles algorithm in the above case, but this is not guaranteed to work for a market outside of UAM.

[20] Such procedural considerations are often absent in economic literature on Walrasian equilibria, but they are vital in a computational context like that of MAT.

Moreover, it is questionable whether the term 'Walrasian auction' would still be appropriate. The whole idea of a Walrasian auction is to let the auctioneer set tool prices according to a *simple* schema, preferably adapting prices purely on the basis of tool scarcities (or other *aggregate* information). Under this constraint, arriving at a price setting like the pairwise equality preorder is a matter of *luck* indeed.

4.8 Walrasian Density in TR\AM

In the previous sections, we have studied the existence of Walrasian equilibria in assignment markets, concluding that, in many cases, a Walrasian equilibrium does not exist. It would be interesting to know to what extent this conclusion can also be drawn for tool reallocation problems outside of AM. In terms of Fig. 4.19, we would like to conclude something about the existence of Walrasian equilibria in the shaded area TR\AM.

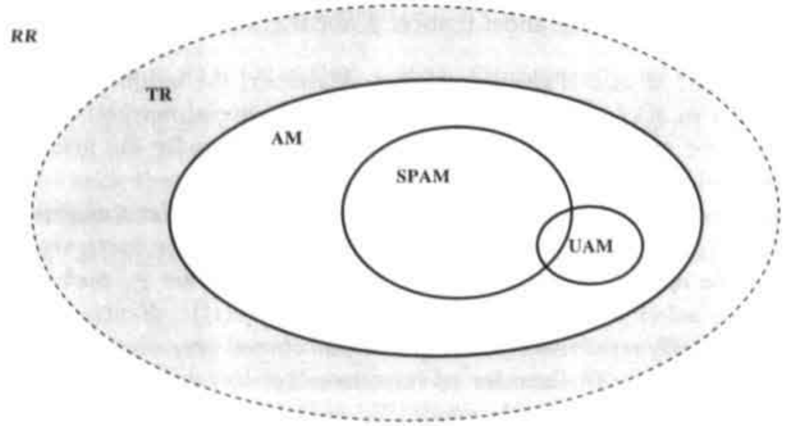


Figure 4.19: Venn diagram of reallocation problem classes.

Recalling our remarks on page 85 in Sect. 4.1, it is not possible to draw steadfast conclusions for the problems in this area, because of the fuzziness of its outer border (i.e., the boundary between TR and RR\TR).

Also, the line of reasoning which we used to arrive at a tractable algorithm to test market stratifiability in AM is of little use for problem instances in TR\AM. For one thing, Prop. 4.15,[21] which is vital for most of the derived properties of Walrasian equilibria in assignment markets, no longer holds if (some) agents possess more than one tool. We can, however, provide *some* plausible arguments as to why it is

[21] This proposition states that the tools involved in an exchange that is part of a Walrasian reassignment always have the same price.

unlikely that the Walrasian auction is a satisfactory mechanism for tool reallocation in TR\AM. The argumentation involves a comparison of Walrasian auctioneering in AM, TR\AM, and RR\TR, viewed as a *static control problem*, in which the auctioneer attempts to control the scarcities of goods by means of prices. Because a bag-based representation cannot be used in RR\TR, we employ a vectorial notation in the sequel. As in Sect. 3.3.3, we use bold font for vectors to distinguish them from scalars.

Abstracting from utilities and preferences, the (aggregate) desires of agents can be described in terms of the *excess demand function*. This function describes the scarcity of each good, as a function of the prices of the goods. It is common in economic literature (Aliprantis *et al.*, 1989).

In the context of Walrasian exchange, a price is an exchange ratio. Hence, a price vector $\mathbf{p}$ can be normalized to a unit length vector without loss of generality. In the following, we therefore assume that

$$\mathbf{p} \in \mathbb{B}^m = \{\mathbf{x} \in \mathbb{R}^m \mid \|\mathbf{x}\| = 1\}$$

Thus, the excess demand function is a function $\zeta : \mathbb{B}^m \to \mathbb{R}^m$, and the goal of a Walrasian auction is to find a $\mathbf{p} \in \mathbb{B}^m$ such that $\zeta(\mathbf{p}) = \mathbf{0}$. As such, the auctioneer attempts to control $\zeta(\mathbf{p})$ through $\mathbf{p}$.

In economic literature (Hildenbrand & Kirman, 1976; Hildenbrand & Kirman, 1988; Aliprantis *et al.*, 1989; Mukherji, 1990), the preferences of agents over the set $\mathbb{R}^m$ of (individual) endowments are usually (e.g., in neoclassical exchange economies) presumed to be strictly monotone. 'More is better', so to speak. Under this assumption, an agent will always spend its entire budget in a Walrasian auction. In other words, the commodity bundle $\mathbf{x} \in \mathbb{R}^m$ that it prefers to obtain in exchange of its current endowment $\mathbf{e} \in \mathbb{R}^m$ is always *just* affordable under the current prices:

$$\sum_{j=1}^{m} p_j \cdot x_j = \sum_{j=1}^{m} p_j \cdot e_j \tag{4.28}$$

In terms of an agent's *individual* demand $\mathbf{d} \triangleq \mathbf{x} - \mathbf{e}$, this implies that

$$\sum_{j=1}^{m} p_j \cdot d_j = 0 \tag{4.29}$$

Since Eq. 4.29 holds for all agents, the excess demand $\zeta \triangleq \sum_{i \in I} \mathbf{d}_i$ of the entire community I has the same property:

$$\sum_{j=1}^{m} p_j \cdot \zeta(\mathbf{p})_j = 0 \tag{4.30}$$

Equation 4.30 is known as Walras' Law. It implies that, *whenever* we change the price vector $\mathbf{p}$ to another value in $\mathbb{B}^m$, this will cause the excess demand for (at least some) goods to change too. We use the term *control space* to denote a minimal subspace of values for the controlling variables (in this case, the price vector $\mathbf{p}$) that are effective in changing the variable to be controlled (in this case, the excess demand ζ). Hence, if we define the relation $\sim$ between price settings as

$$\mathbf{p} \sim \mathbf{p}' \Leftrightarrow \zeta(\mathbf{p}) = \zeta(\mathbf{p}')$$

then the control space of a Walrasian auction equals $\mathbb{B}^m/_\sim = \mathbb{B}^m$ if the goods are infinitely divisible.

In the face of *indivisible* goods, the control space is much smaller. The observations of low Walrasian densities in subspaces of AM in the previous sections can be explained in terms of these smaller control spaces. In an assignment market involving m tool types, there are at most $|\text{PREORD}[m]|$ effectively different price orders. In other words, the size of the control space is bounded from above by $|\text{PREORD}[m]|$, *irrespective* of the number of agents participating in the market. In contrast, the size of the problem space increases with n, and does not have an upperbound that is independent from n. The fact that the number of control decisions has an upper bound that depends on the number of tool types, but *not* on the number of agents, implies that, in assignment markets where the numbers of agents is much greater than the number of tool types, a Walrasian auctioneer will be unable to control individual agents. In fact, the discrepancy between Walrasian density in UAM and AM\UAM stems from the fact that, in UAM, changing the price of one tool amounts to changing the budget set of exactly one agent, whereas in AM\UAM, such a price change generally influences the behavior of many agents that currently possess the tool type involved, and some others (for which the price change removes or adds the tool type to the agent's budget set). The impact of the inability to control the behavior of individual agents grows, of course, with the courseness of the control. This explains why the Walrasian density decreases if we keep m fixed, while increasing n.

A similar observation can be made for (at least some) problems in TR\AM. Consider, for example, the subclass of tool reallocation problems involving m tool types, where each agent is endowed with *one or two tools*, and tool-bag preferences are such that every agent prefers its initial endowment over any endowment that contains *less* tools.[22] In this case, tool-bag preference is essentially constrained to bags of the same size as the initial endowment, and a similar property holds for the budget sets of the agents (i.e., any differences between budget sets that pertain only to bags containing more than two tools are irrelevant in the sense that, in equilibrium, all budget sets *must* be limited to bags containing at most two tools.) In TR problems with the above characteristics, endowments must be of the type $\{x, y\}$, or $\{x, x\}$ or

[22] This implies that differences in tool utility for different tool types are small in comparison with the utility of any single tool.

$\{x\}$. Hence, the *maximum* number of different endowments in such a community is

$$M_m = \binom{m}{2} + m + m \tag{4.31}$$

Any mapping from endowments to budget sets can be defined by specifying the relation "has at least as much value as" between any two *conceivable* endowments. Consequently, the size of the control space is, in this case, bounded by[23]

$$|\text{PREORD}[M_m]| = \left|\text{PREORD}\left[\binom{m}{2} + 2m\right]\right| \tag{4.32}$$

Again, we have specified an upperbound on the control space that does not depend on the number of agents. Hence, low Walrasian densities are to be expected in markets with many more agents than tool types.

4.9 Chapter Summary and Conclusions

The main theoretical results we derived are the following.

- We have characterized the (typed) assignment markets in which a Walrasian equilibrium exists in terms of their preference profile and initial endowment, showing that such an equilibrium exists iff the assignment market is stratifiable.

- We have shown that, in a stratifiable assignment market with strict preferences (a SPAM), there exists *only one* Walrasian allocation.

- We have shown that, within the class SPAM* of *viable* SPAMs, large subclasses exist that consist entirely of non-Walrasian markets, and that most of the markets in the (rare) subclasses which consist entirely of Walrasian markets contain only *equilibrial* markets, which can be regarded as *trivial* reallocation problems.

- An algorithm to test SPAMs (i.e., assignment markets in which the preferences of all agents are strict) for stratifiability was specified, based on derived properties of market stratifications. It was indicated how this test can – in principle – be applied to assignment markets with weak preferences also, via the SPAM enumeration of such markets.

[23] The bound is not a sharp one, since it neglects that the price preorder must be consistent with the subbag-of relationship.

The algorithm was used to compute estimates for the density of Walrasian markets in various subspaces involving strict-preference assignment markets. From such density estimation, it appeared that Walrasian densities decrease rapidly as the number of agents (or the average number of tools per tool type) increases. Except for some trivial cases, these densities seem too low for a Walrasian auction to be useful.

On the basis of the theoretical and experimental results, we draw the following conclusions.

1. Except for the special case of untyped reassignment, the application of Walrasian auctions to tool reassignment problems is highly problematic, especially for problems with many agents and relatively few tool types.

2. Our analysis of the existence of Walrasian equilibria in TR\AM suggests that the Walrasian density in the space of general tool reallocation problems is subject to the same pattern (of decreasing density with increasing population size). Hence, the Walrasian auction does not seem suited for general tool reallocation either.

Chapter 5

The Framework of Informed Gambling

5.1 Chapter Overview

In this chapter, we describe Informed-Gambling (IG) mechanisms for Tool Reallocation (TR), in terms of the bag-based representation introduced in Chapter 2. In Sect. 5.2, we recapitulate the conclusions drawn in Chapter 3 on candidate distributed approaches to TR problems. In Sect. 5.3, we present the general notion of iterative mechanism. This abstract notion provides the context for the formal definition of Informed-Gambling mechanisms. It is, however, sufficiently general to accommodate most game-theoretic and economic mechanisms, such as the Walrasian auction, as well. The general idea of mechanism design is that the outcome mapping serves as an *incentive* for rational agents to relax their demands if these conflict with those of other agents.

In Sect. 5.4, we explain what is meant with 'rational agents'. This entails a short discussion of four key notions: perfect rationality, bounded rationality, the principle of minimal rationality, and the descriptive level of rationality.

In Sect. 5.5, we illustrate the effectiveness of the heuristics employed in the outcome mapping of IG, by describing the mechanism that has been its precursor. This *delegated-negotiation* (DN) mechanism (Lenting & Braspenning, 1993) features the same outcome mapping as IG, but it is a simpler: DN is single-shot mechanism, while IG is iterative. We show by means of an example how DN's outcome mapping resolves clashes of interest between agents, by creating a situation in which agents must make commitments in the face of uncertainty to satisfy their desires. In addition, we demonstrate how the mapping can be implemented in a fully decentralized manner.

In Sect. 5.6, we formally define the class of IG mechanisms. A key component of IG mechanisms is a form of agent rationality which we refer to as fictitious rationality. It is similar to the rationality exhibited by agents in a Walrasian auction, but it is Bayesian in nature, due to the fact that we choose to keep the agents less well informed. Its core element is the estimation of proposal success probabilities. Explicit formulas are provided for such estimation. In the general case of full-fledged reallocation problems, these formulas do not constitute a complete definition of the agent response, but they can serve as a basis to define a bounded-rational response suited for a particular problem domain. For the tool *reassignment* domain, we provide a complete, implementable specification of fictitiously rational agent response, and show that the associated mechanism is capable of avoiding the trust formation that can lead to low-quality outcomes with the Walrasian exchange auction, or other mechanisms that render solutions in the core of the associated assignment games (cf. Section 3.3.8).

In Sect. 5.7 we show that the termination of the relaxation process in IG is a problematic issue, and propose the incorporation of *negotiation weariness* as a remedy. This entails that agents gradually become more indifferent with respect to proposal adaptation. Initially, the prospect of a minute utility gain is sufficient reason for an agent to adapt its proposal, but as the negotiation process proceeds, agents gradually become less fussy. Whereas such behavior is anthropomorphically plausible — it plays an important role in political negotiations between human agents — it does constitute a breach with the game-theoretic dogma that one should assume agents to behave rationally, that is, to grab *every* opportunity to increase their utility.

In Sect. 5.8, we attempt to characterize IG mechanisms in game-theoretic terms. The stationary states of the relaxation process in IG can be viewed as correlated equilibria. Apart from this, there appears to be little kinship with game theory or economics: IG's outcomes are neither always individually rational, nor always Pareto optimal. The kind of rationality exhibited by IG agents in the context of simple (tool-reassignment) problems is neither perfect nor bounded; to characterize it, we introduce the new notion of near-perfect rationality. And even the characterization of IG's final states as correlated equilibria has little importance, since it only applies to IG mechanisms without negotiation weariness, for which we cannot guarantee that a correlated equilibrium will ever be reached.

In Sect. 5.9, we address the adequacy of IG as an instrument to solve real-life reallocation problems. For CMAT, this involves a discussion of the kind of bounded rationality that is called for in IG mechanisms for real-life reallocation and constrained optimization. For OMAT, we reflect on the adequacy of IG as a negotiation framework for unknown, external agents. Since it is commonly considered permissable to assume that unknown (computational) agents behave rationally, our reflection focuses on the question whether the 'irrational' aspects of IG agent behavior (viz. negotiation weariness, and the neglect of part of the available information by fictitiously rational agents) qualify as acceptable assumptions in the context of open systems.

The chapter is concluded in Sect. 5.10 with a summary of concepts and findings.

5.2 The Design Considerations for IG

An important goal behind the development of IG has been to conceive a framework that is principally suited for CMAT as well as OMAT. These two fields induce different requirements on the design. Computational efficiency and solution quality are relevant in both cases, but in OMAT, we face the additional requirement that an agent-behavior specification should constitute a plausible model for the — principally unpredictable — behavior of unknown, external agents. This additional requirement complicates the design task considerably. For the sake of simplicity, we choose to postpone the discussion of such complications, and present the IG framework as if it were designed for CMAT only. This entails that, for the time being, we merely *mention* aspects of the design which are at odds with OMAT agent properties like autonomy or self-centeredness. The *motivation* of our design choices in this respect is postponed until Sect. 5.9.2, at the end of the current chapter.

In Chapters 3 and 4, we have drawn a number of conclusions that are relevant for the design of tool-reallocation (TR) mechanisms, such as IG and its predecessor, DN. These conclusions can be summarized as follows.

1. To solve TR problems, coordination via a central module is preferable to fully decentralized coordination (Sect. 3.1.1).
2. If one aims at operational decentralization, one should not assign a complex and demanding task to the central coordination module of a TR mechanism (Sect. 3.4).
3. A mechanism that features communication of *composite* reallocation proposals is liable of incurring excessive computational and design complexity in the coordination module (Sect. 3.4).
4. The productivity of an iterative mechanism should not depend on the attainment of equilibria which exist only under certain (restrictive) conditions (Sect. 4.9).
5. It is worthwhile to develop a mechanism that does not make use of money (Sect. 1.3.2 and 3.1.2).
6. Trust formation can lead to low-quality outcomes in markets without money (Sect. 3.3.8).

In Constraint-Directed Negotiation (CDN),[1] the coordination is taken care of by a mediator who proposes partial solutions in an ongoing discussion with individual agents. In view of the above conclusions, we have decided to let the coordination module in IG mechanisms play a more modest role. It acts as an *auctioneer*, whose main task is to aggregate data received from the agents into relevant global problem characteristics, which are then communicated — as coordination messages — to the agents. The agents respond by revising their previous proposals, and this dialectic process continues until a stable state is reached. A decision protocol maps the associated tuple of agent proposals to a final outcome.

The functionality of the coordination messages in DN and IG is similar to that of the problem textures communicated among agents in the Cortes architecture (Sycara *et al.*, 1991): in conjunction with the rules of the decision protocol (which is assumed to be known to the agents), the coordination messages enable the agents to take collective interests into account without being aware of another agent's individual goals, or even its existence. This implies that we need not concern ourselves with second-order (or higher-order) beliefs of agents about beliefs of other agents (Cohen & Levesque, 1987; Shoham, 1993), which are liable of greatly complicating the negotiation process, with little or no gain in terms of the average agent satisfaction. In this respect, the information in an IG market profile is akin to the price information conveyed to agents in a Walrasian auction. The main difference between the two kinds of coordination messages is twofold: In IG, the coordination messages are less informative, and easier to compose than in the Walrasian auction. Hence, in terms of operational decentralization, IG surpasses not only CDN, but also the Walrasian auction.

IG mechanisms for tool reallocation are based on the metaphor of the *all-pay auction* (Weber, 1985; Milgrom, 1985). Contrary to most auction types, bidders engaged in an all-pay auction are required to pay the amount mentioned in their (last) bid, irrespective of whether they acquire the item they bid on. As such, the protocol of an all-pay auction resembles that of lotteries and option markets. The all-pay auction has been used in the past as a model for bribing and lobbying (Weber, 1985; Baye & Kovenock, 1993). We use it as a metaphor for commitment in the face of uncertainty. This plays a key role in IG.

Unlike the all-pay auction, IG features barter trade. Agents pay for tools with other tools instead of money. As such, Informed Gambling can be characterized as a mixture of an all-pay auction and Walrasian exchange.

[1] cf. Sect. 3.1.2 and 3.2.1

5.3 The Notion of Mechanism

The conceptual framework of self-centered agents acting autonomously within some kind of *mechanism* has proven useful to solve problems in a decentralized fashion.

From a software-engineering perspective, a mechanism is a software specification that involves an agent model and a description of a coordination module. The agent model specifies how agents behave in response to the messages of the coordination module. It is inherent to the notion of agent that this response is *rational*, that is, understandable in view of the agent's personal interests and its current (knowledge of the) situation. Rational agent behavior is commonly regarded as vital in the context of open systems, where the designer of a mechanism designs only part of the system (e.g., the coordinations module), and, consequently, has no absolute control over the behavior of all agents. For the design of closed multi-agent systems, agent rationality is merely convenient, because of the design modularity it incurs. Hence, in CMAT, the agent model does not specify how external agents are *assumed* to behave, but how they are *designed* to behave. Obviously, this leaves the mechanism designer more slack to promote the efficiency of the mechanism.

To describe agent behavior within a heterogeneous agent population, multiple agent models are required. In our IG framework, the agent population is assumed to be homogeneous, that is, all agents use the same procedure to determine their responses to external stimuli. This does not imply that all agents exhibit the same response, because an agent's endowment and its tool bag utilities are variables in the agent model. Furthermore, IG agent behavior is generally not completely deterministic, so even if two agents have exactly the same endowments and tool-bag utilities, they may behave differently.

The coordination module of a mechanism comprises a set of rules that determine a mapping of the ensemble of agent messages to a final outcome. An agent that knows these rules will take them into consideration when it chooses a strategy. Thus, the rules in the coordination module can be used to constrain the agents' behavior in some desired manner. In general, the coordination rules map a tuple of agent messages to a *set* of possible final states. As in economics literature, we refer to such a nondeterministic mapping as a *correspondence*.

It is useful to distinguish between single-shot and iterative mechanisms. In a single-shot mechanism, the agents get *one* opportunity to revise their original message in response to a message of the coordination module. In iterative mechanisms, there is room for multiple proposal revisions.

5.3.1 Iterative mechanisms

An iterative mechanism features a dialectic process of agents responding to information provided by the coordination module and vice versa, until some termination criterion is met. Reaching a stationary state is the most common termination criterion. The final outcome then depends on the messages communicated in the last round of agent response. An IG mechanism, the target concept of this chapter, is an iterative mechanism. So is a Walrasian auction.

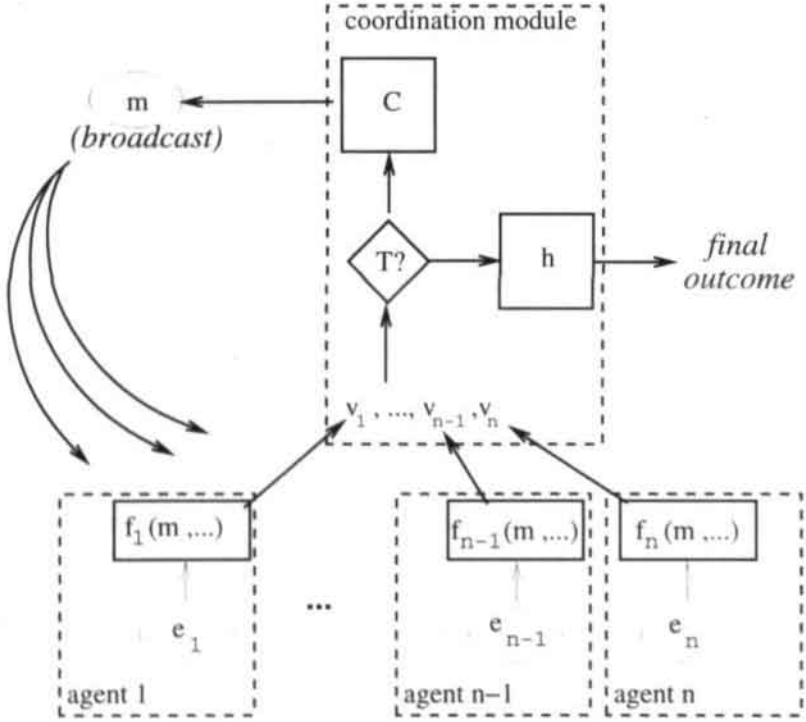


Figure 5.1: Our general model of iterative mechanisms: agent messages v_i and coordination messages m are communicated until some termination criterion T is satisfied; a final outcome is then computed from the last-sent agent messages.

Our general model of iterative mechanisms, of which IG mechanisms constitute a special case, is shown in Fig. 5.1. Iterative mechanisms are similar to *message mechanisms*, as presented in (Hurwicz, 1986). The main difference is that, in an iterative mechanism, an agent does not perceive the messages sent by other agents, but only an aggregate thereof. Furthermore, some of the mappings that are functions in Hurwicz' model are correspondences (i.e., involve nondeterminism) in ours.

The input to the mechanism is the distributed problem specification $e = \langle e_1, \ldots, e_n \rangle$ in the lower part of Fig. 5.1. For tool reallocation problems, e_i comprises the endowment and the tool bag utilities of agent i. The information e_i is *private*, that is, only accessible to agent i.

The *agent response correspondence* f_i describes how agent i responds to the messages of the coordination module. v_i^t denotes the response of agent i in round t to the coordination message m^{t-1} transmitted after round $t-1$. The agent response messages v_i^t are elements of some message space $\mathcal{M}_A$. The response correspondence f_i describes how v_i^t depends on the agent's previous response v_i^{t-1} (if any), its private information e_i, the last coordination message m^{t-1} (if any), and the time t (i.e., the number of message rounds that have already taken place).

The coordination message $m^t \in \mathcal{M}_C$ is a function of the agents' responses $v_1^t, \ldots, v_n^t$.[2] We refer to the dialectic process of agents responding to coordination messages and vice versa as the *adjustment process* of the mechanism. Formally, this process is described by a set of nondeterministic difference equations of the following kind.

$$\begin{cases} v_1^t & \in & f_1(e_1, m^{t-1}, v_1^{t-1}, t) \\ \vdots & & \\ v_n^t & \in & f_n(e_n, m^{t-1}, v_n^{t-1}, t) \\ m^t & = & C(v_1^t, \ldots, v_n^t) \end{cases} \tag{5.1}$$

where the initial agent response v_i^0 of agent i in round 0 is determined solely by its private information e_i. The termination of this process is governed by a termination criterion T that depends on v^t and v^{t-1} (where $v^t \triangleq \langle v_1^t, \ldots, v_n^t \rangle$), and sometimes on t. The default criterion is $v^t = v^{t-1}$, but we will also consider others. If the adjustment process terminates at step t, the outcome correspondence $h : \mathcal{M}_A^n \rightarrow \mathcal{P}(Z)$ renders a final outcome z,[3] according to

$$z \in h(v_1^t, \ldots, v_n^t) \tag{5.2}$$

Thus, the coordination module of an iterative mechanism is represented by the triplet $\langle C, T, h \rangle$, and the behavior of agent i by f_i, with e_i as a parameter. Formally, we define an iterative mechanism as a tuple

$$M = \langle \mathcal{M}_A, \mathcal{M}_C, E, Z, C, T, h, f \rangle \tag{5.3}$$

where $\mathcal{M}_A$ and $\mathcal{M}_C$ are the message spaces for the agents and the coordination module respectively, E is the problem space, Z the outcome space, C, T, and h are defined as above, and $f : E \times \mathcal{M}_C \times \mathcal{M}_A^n \times \mathbb{N} \rightarrow \mathcal{P}(\mathcal{M}_A^n)$ describes the collective

[2] To accommodate for the Walrasian auction, one would have to add the previous coordination message m^{t-1} to the list.

[3] Here, $\mathcal{P}(Z)$ denotes the power set of Z.

behavior of the agent community. Here, $\mathcal{M}_A^n$ is the usual shorthand notation for $\mathcal{M}_A \times \cdots \times \mathcal{M}_A$, the n-fold Cartesian product of $\mathcal{M}_A$ with itself. As apparent from the above formulas, C is a deterministic function, but f and h are nondeterministic functions, represented as correspondences. The agents' knowledge of C and h is supposed to play a role in f, but we come to this later.

The above dynamic model of iterative mechanisms is associated with a static model, that abstracts from the dynamic interplay between agents and coordination module, described by C, T and f. The *static adjustment correspondence* is the mapping

$$\mu : E \to \mathcal{P}(\mathcal{M}_A^n)$$

that specifies the set of possible final agent response messages in the adjustment process as a function of the private information $e \in E$.
Formally, μ is defined as

$$\begin{aligned} \mu(e) = \{v^t \in \mathcal{M}_A^n \quad | \quad & (\exists v^0, v^1, \ldots, v^{t-1}) \ (\\ & v^0 \in f(e, \bot, \bot, 0) \\ \wedge \quad & v^j \in f(e,\ C(v^{j-1}),\ v^{j-1},\ j-1), \quad j \in \{1, \ldots, t\} \\ \wedge \quad & T(v^t, v^{t-1}) \quad) \ \} \end{aligned} \tag{5.4}$$

Here, the symbol $\bot$ signifies "undefined". In words, Eq. 5.4 expresses that a final agent response tuple v^t is an element of $\mu(e)$ if it is the agent response component of the end point $x^t = \langle e, m^t, v^t, t \rangle$ of a possible trajectory $\langle x^0, \ldots, x^{t-1}, x^t \rangle$ through the state space $E \times \mathcal{M}_C \times \mathcal{M}_A^n \times T$, starting at the point $x^0 = \langle e, \bot, \bot, 0 \rangle$.

This definition of μ is the intermediate step that helps us to define the overall effect (in terms of possible final outcomes $z \in Z$) of applying the mechanism M to a problem instance $e \in E$. We refer to the overall mapping $F_M : E \to \mathcal{P}(Z)$ from the problem space $E = E_1 \times \ldots \times E_n$ to the (power set of the) outcome space Z as the *performance correspondence* of $M = \langle \mathcal{M}_A, \mathcal{M}_C, E, Z, C, T, h, f \rangle$.
Formally,

$$F_M(e) = \{z \in Z \mid \ z \in h(v) \wedge v \in \mu(e)\} \tag{5.5}$$

A single-shot mechanism is a special case of the above model for iterative mechanisms, where the termination criterion T is "$t = 1$". Note that the first round is round 0. Hence, the invocation of a single-shot mechanism involves the communication of one coordination message and two agent messages per agent.

5.3.2 Mechanism design

In literature on game theory, social choice theory, and micro-economics, the design of mechanisms is a prominent topic.

The goal of mechanism design in social choice theory is to guarantee some desirable property, such as Pareto optimality of the final outcome, or truthfulness of the agents' response. In social choice theory, mechanisms often do not involve any coordination message. Such mechanisms do not really fit into our definition of iterative mechanisms, but we will act as if they do (i.e., regard them as degenerate cases of our general model), because the key idea behind their design is also relevant for the design of iterative mechanisms within MAT. This key idea concerns the conception for an outcome correspondence that can guarantee the desired property by providing suitable *incentives* to the (self-centered and self-ruling) agents (Maskin, 1985; Postlewaite, 1985; Myerson, 1985). In the sequel, we will use the term *incentive engineering* for this style of mechanism design. In terms of our general model of iterative mechanisms, the core activity of incentive engineering is the design of an outcome correspondence h which, given a rational[4] agent response correspondence f, guarantees the desired property for the performance correspondence F_M.

The idea of incentive engineering also plays a role in the design of mechanisms that serve as market models in micro-economics, such as the Walrasian auction, but because these mechanism are iterative, the focus of attention lies elsewhere: Most of the effort goes into the conception of a coordination message function C (i.e., a tâtonnement algorithm) that can guarantee the convergence of the dynamic adjustment process (and is economically plausible) (Scarf, 1973; Joosten, 1996).

Another aspect in which iterative mechanisms differ essentially from single-shot mechanisms or social choice mechanisms is the definition of rational agent response. The fact that iterative mechanisms involve a *sequence* of agent responses complicates the conception of a sound formal definition of rationality considerably.

In economics, one typically employs a convenient a-priori assumption to circumvent these complications. In a Walrasian auction, for example, the assumption of perfect competition ensures that it is rational for agents to take prices for granted, instead of attempting to influence their evolution. Thus, within a Walrasian auction, tropistic agent response is rational; neither past nor future prices are relevant for the agents' present decisions.

In Informed Gambling, we do not assume perfect competition, because we regard this assumption as too restrictive for MAT. As a consequence, conceiving an adequate specification of agent response is more difficult for IG mechanisms than in mechanism design in economics or social choice theory. This is particularly so in the context of open systems, since the rationality of the agents' behavior is vital here. Another complicating factor in this respect is the uncertainty, which is incorporated in IG to prevent trust formation, and to enable cardinal utility comparison without

[4]For now, *rational* should be interpreted as "motivated solely by the desire to maximize ones own satisfaction"; more precise definitions will be given later.

communicating utilities.[5]

In comparison with f, neither C nor h involves much design effort in IG. As in economic mechanism design, the outcome mapping h aims to provide incentives to the agents to behave in a certain manner. However, unlike economic mechanisms, IG mechanisms are not designed for their performance correspondence to have a specific *property*. Our aim has been to conceive mechanisms that render outcomes of high quality in a (relative-)utilitarian sense. Since this design goal is not susceptible to rigorous mathematical proof, the design of the outcome mapping h involves less effort than in social choice theory: The mapping is heuristically motivated, and its adequacy will be evaluated (in Chapter 7) by means of experimentation.

5.4 Rational Agent Behavior

In everyday speech, we label somebody's decision or action as *rational* if we can ascribe a viable purpose to it. In view of its inherent subjectivity, this definition is rather vague.

5.4.1 Perfect rationality

Definitions of rational (agent) behavior in artificial intelligence (and related disciplines such as game theory and micro-economics) tend to be more specific. A common element in most of these definitions is *utility maximization*. Behavior is considered rational iff it can be explained completely as stemming from the strife for an outcome of maximal utility.

In the classical model of rational decision making, each agent picks a strategy from its strategy space which it believes will lead to an outcome of maximal utility. Here, outcome utilities are quantitative, a strategy space is a *finite* set of strategies, and a strategy is either a single deliberated action, or a course of action that extends beyond the present. The beliefs of an agent on the causal connection between their strategy choice and the outcome are based on the agent's knowledge of the strategy spaces and outcome utilities of the other agents. This classical conception of rational decision-making is known as *perfect rationality*.

In our IG framework, we use the above terminology, except that the term *strategy* always refers to a single agent message, since IG agents are not supposed to plan ahead.

The above conception of perfect rationality is not suited at all for computational agents operating within an iterative mechanism like IG. The most obvious discrep-

[5] In this respect, the uncertainty incorporated in the IG framework can be regarded as a substitute for the price information communicated in a monetary Walrasian auction.

ancy is that perfect rationality is grounded in *knowledge* (or at least definite beliefs) concerning the behavior of other agents, and the outcome is fully determined by the strategy choices of all agents. An agent operating within an iterative mechanism faces considerably more uncertainty. For such an agent, it is very difficult to form definite hypotheses about the behavior of specific other agents, since it only perceives the coordination message $m = C(v)$ *composed* from the agent messages $v = \{v_1, \ldots, v_n\}$. And even if an agent succeeds in forming definite hypotheses on the agent messages behind the coordination message, it is still not certain of the outcome: It only knows that this will be some value in the *set* $h(v_1, \ldots, v_n)$.

As such, the above description of perfect rationality pays too little attention to uncertainty to be useful in our framework. The following formal definition of perfect (Bayesian) rationality aims to mend this deficiency.

Definition 5.1 (perfect (Bayesian) rationality)
An agent that is uncertain of the consequences of its decisions exhibits perfect rationality iff it always chooses a strategy such that the conditional expectation of the utility of the outcome, given the available information, is maximal. △

Note that Def. 5.1 differs from perfect rationality as described above, not only in that it is a formal, probabilistic definition, but also in that it is essentially behavioral: The criteria which determine whether or not an agent exhibits perfect Bayesian rationality pertain solely to the decision the agent takes; how it arrives at this decision is not taken into account. The definition can, however, be applied to a decision procedure if 'always chooses' is replaced by 'is guaranteed to choose'.

5.4.2 Bounded rationality

The classical conception of perfect rationality was criticized in (Simon, 1955), because it neglects the facts that both the knowledge and the reasoning capacity of a human being is *bounded*. Hence, it may not be feasible for a human agent to select a strategy that is optimal in terms of outcome utility, due to insufficient computational capacity or incomplete knowledge. Furthermore, even if perfect rationality is feasible, it need not be sensible, if it incurs a high computational cost. Simon proposes the term *bounded rationality* for a model of human decision making which takes the above considerations into account.

Obviously, Simon's criticism is also relevant for computational agents, in OMAT as well as CMAT. In Def. 5.1, we have — at least in principle — dealt with one of the two objections against the classical conception of perfect rationality: its lack of realism with respect to uncertain or missing information. To cope with the other objection as well, one needs to relax the requirement that the chosen strategy *maximizes* the expected utility. This can be done in various ways. Although different domains tend to call for different implementations of boundedly-rational agent response, one can distinguish a few domain-independent *categories* of bounded rationality. We mention four such categories.

1. strategy-space filtering
2. strategy-space limitation
3. strategy-space expansion
4. cost-calculative rationality

Strategy-space filtering entails that, to cut down on their computational expenses, agents merely filter out those strategies that are *obviously inferior* to some other strategy, and choose randomly from the remaining ones. This kind of bounded rationality is adequate if the strategy space is small, some strategies can be compared very easily, while other comparisons incur a high computational cost. We employ strategy-space filtering in our specification of delegated negotiation, the forerunner of IG.

Although strategy-space filtering avoids costly comparison procedures, it does require that comparison is at least considered for each pair of strategies in the strategy space. If strategy spaces are too large for this to be feasible, or if it is not immediately apparent for an agent what its strategies are, agents can employ strategy-space limitation, or strategy-space expansion. Limitation simply means that agents only investigate a small — possibly rather arbitrary — subspace of strategies. Expansion is a refinement thereof, in which agents deliberate the investigation of other regions of the strategy space, after, and on the basis of their previous investigation(s). We will propose strategy-space expansion as a suitable basis for strategy selection by agents in IG mechanisms for constrained optimization.

The last category, cost-calculative rationality, entails that agents estimate the computational costs associated with deliberated calculations to decide which calculations should be performed. This category is not disjoint from the others. It can be combined with each of them, and may also be used in relation to other computational tasks (e.g. to conditionalize the computation of outcome utilities that are not readily available). We do not use such an approach.

Perfect and bounded rationality are commonly viewed as mutually exclusive characterizations of decision making. However, with our formal definition of perfect rationality (in Def. 5.1 above), these concepts are no longer disjoint: It is quite possible that bounded rationality is perfect, because the label "bounded rationality" pertains to the procedure of decision making, while perfect rationality — in the sense of Def. 5.1 — pertains to the outcome of the decision process. Even a procedure that does not *aim* for perfection can sometimes render a perfect result. In Sect. 5.8.2, we will use this observation to characterize the agent response in IG mechanisms for tool reassignment.

5.4.3 Other aspects of rationality

Next to the notion of perfect (Bayesian) rationality, and the above categories of boundedly-rational strategy selection procedures, there are two aspects of agent rationality that are relevant for the sequel, but not specifically related to the computational cost of decision making. They pertain to the way in which an agent which faces uncertainties forms hypotheses about its situation. These aspects are

- the principle of minimal rationality;
- the descriptive level of agent rationality.

The principle of minimal rationality entails that an agent which has to hypothesize about the situation it is in, should opt for the simplest possible hypothesis if there are no apparent objections against this. If, for example, the agent knows that the value of some variable x is in some set X, but has no clue with respect to the probability that $x = y$ for any y in X, it should either hypothesize that x has some arbitrary value y in X, or assume that all values in X are equally likely. In the absence of reasons to prefer the latter assumption, it should opt for the simpler $x = y$. Similarly, if the agent does have some information, for example from past observations of x, and knows that x tends to change gradually with time, it may assume that the current value of x equals its last-observed value y, or it may hypothesize a probability distribution centered around y. Again, it should employ the definite hypothesis $x = y$, if there is no compelling reason to opt for the more complex, probabilistic hypothesis instead. In view of the vagueness of the phrase "no compelling reason", the minimal-rationality principle does not constitute any *definite* specification of rational behavior, or even a category of such specifications. It is merely a guideline, similar to Ockham's razor in theory development, and the concept of null hypotheses in statistics. To arrive at a formal definition for the rationality exhibited by IG agents, we will use the minimal rationality principle in multiple respects.

The other aspect, the descriptive level of agent rationality, constitutes a classification of rational behavior specifications according to the subject of the agents' hypotheses. We distinguish between hypotheses on

1. the hypotheses which other individual agents will make;
2. the behavior which other individual agents will exhibit;
3. the collective behavior which the other agents will exhibit.

The bulk of agent rationality in AI research belongs in category 1. In most cases, one pays little attention to the depth to which the recursion — inherent in this type of hypothesizing — should be allowed to proceed. A notable exception is the research by Durfee and collaborators on the Recursive Modeling Method (RMM) (Gmytrasiewicz & Durfee, 1995; Durfee, 1995; Vidal & Durfee, 1996).

With the exception of Bayesian equilibria (Harsanyani, 1968; Mertens & Zamir, 1985; Myerson, 1985), the rationality conceptions implied by game-theoretic equilibrium notions fall in category 2.

MAS research featuring rationality definitions in category 3 is largely confined to economically inspired approaches, such as MOP (Wellman, 1994b). IG's fictitious rationality is also in this category. Recently, category-3 rationality has been advocated in multi-agent learning (Schmidhuber, 1996) and theoretical MAS research in the spirit of classical, symbolic AI (Singh, 1998).

5.5 Delegated Negotiation: A Single-Shot Mechanism

5.5.1 *Delegated negotiation as a stepping stone toward IG*

We envision different Informed-Gambling (IG) mechanisms for different kinds of reallocation problems. The common elements in these variants are

- They fit in the framework of iterative mechanisms.
- The messages communicated by the agents to the coordination module are *elementary* reallocation proposals.
- A submitted reallocation proposal constitutes a *commitment* by the associated agent, rather than a constraint that must be obeyed by the coordination module.
- The functionality of the coordination module is similar to that of an *auctioneer*: it aggregates the reallocation proposals into a market profile, which comprises the supply and demand of the tool types in the proposals, and communicates this market profile to the agents.
- The outcome correspondence embodies a heuristic that favors proposals which offer scarce tool types.

One of the key elements of the IG framework is the scarcity-based heuristic employed in its outcome correspondence. We explain how this heuristic works by discussing its use in IG's precursor, delegated negotiation (DN) (Lenting & Braspenning, 1993). DN has the same outcome correspondence as IG, but is simpler in other respects. IG is an iterative mechanism, in which the composition of tool exchange proposals is preceded by multiple rounds of relaxation. In contrast, DN is a single-shot mechanism that attempts to solve reallocation problems by means of composition alone (cf. Sect 3.2).

5.5.2 Delegated negotiation for proposal composition

Delegated negotiation (DN) comprises only the easiest subtask in tool reallocation: the composition of individual proposals into completely satisfiable cascades. In fact, it is more appropriate to speak of *pseudo*-composition. The pseudo-composition that takes place in delegated negotiation is similar to composition as defined in Sect. 3.2.3, except that the conditionality in proposals is not always respected.

In mediated negotiation (Sathi & Fox, 1989), the proposal $\{x \rightleftharpoons y\}$ is interpreted as a *conditional constraint*, expressing "I am prepared to relinquish tool x, *provided* that I acquire tool y". This conditionality of $\{x \rightleftharpoons y\}$ is respected by the mediator: Though the mediator may propose a relaxation $\{x \rightleftharpoons z\}$, it is the *agent* that decides whether this relaxation is acceptable.

In delegated negotiation, the proposal $\{x \rightleftharpoons y\}$ is not interpreted as a constraint, but as a *commitment*. An agent proposing $\{x \rightleftharpoons y\}$ expresses that it would like to acquire y, and, in order to get y, it *commits* itself to relinquish x. The all-pay characteristic of delegated negotiation entails that it must keep this commitment, even if it ultimately obtains z instead of y. The motivation for this irreverence with respect to agent desires is to prevent inflexibility of a single agent from obstructing a trade cascade that would be proficient for many.

In CDN, the mediator negotiates with the agents to overcome such an obstruction, by proposing relaxations of the agents' initial demands. Because this tends to be a cumbersome process, the mediator is a computation and design bottleneck in CDN. In contrast, the use of agent commitments in DN leads to a straightforward solution procedure, which — apart from some trivial pre-processing by the auctioneer — allows for fully decentralized implementation.

As shown in Fig. 5.2, the activity in delegated negotiation comprises four stages, each with distinct communication and activity patterns.

In the first stage, both the auctioneer and the agents are active. Agents communicate their reallocation desires to an auctioneer in the form of a single, *composite* reallocation proposal (e.g., $\{a,a,b \rightleftharpoons c,d\}$), which specifies their most-preferred tool exchange.

The second stage features only auctioneer activity. The individual proposals are aggregated by the auctioneer into a *market profile*. This profile comprises the community's demand and supply of the tool types in the proposals. In addition to this market profile the auctioneer composes tool-seller lists. These specify, for each tool

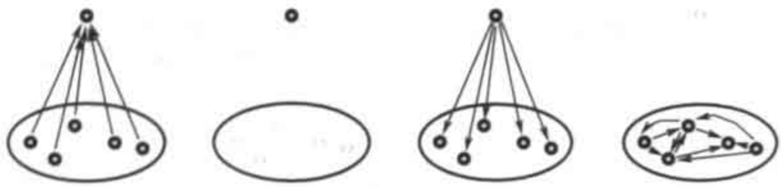


Figure 5.2: Communication and activity in delegated negotiation (boldface circles denote agent and auctioneer activity).

type, which agents have offered the tool type in their proposal.

In the third stage, all participants are active again. The auctioneer communicates seller lists to the buyers, and market profile data to sellers and buyers of scarce tools. This information determines the behavior of agents in the fourth stage, in which pseudo-composition takes place in a decentralized fashion, without participation of the auctioneer.

5.5.3 The pseudo-composition protocol

Initially, we will describe the protocol for pseudo-composition as if it were performed by the auctioneer, since this is easier to explain.

The protocol must resolve the question who should get a tool of type x if the demand for x exceeds its supply. It does this by defining an eligibility ordering on proposals, which is based on the scarcity (demand minus supply) of the tool type that is offered in the proposal (if any).[6] The heuristic rule embodied in the protocol entails that an agent is more *eligible* to receive a scarce tool, if it can offer another scarce tool in return. This creates an incentive for the agents to release tools that are valuable for other agents.

In formal terms, the eligibility of the proposal $\{x \rightleftharpoons y\}$ is a nondecreasing function of the scarcity of x. This leaves some room for variation. The eligibility can be equated with the scarcity, or the sign of the scarcity, or yet another nondecreasing function of scarcity. In Chapter 7, we experiment with various eligibility definitions to see to what extent the various possibilities influence the quality of solutions. For now, we define the eligibility of proposal $\{x \rightleftharpoons y\}$ to be -1 if x is oversupplied, and equal to the scarcity of x otherwise. The eligibility of an unconditional request for x (i.e., a proposal $\{\rightleftharpoons x\}$) is -1. Proposals offering a tool type of scarcity zero are more

[6] The protocol allows only elementary proposals (i.e., proposals of the form $\{x \rightleftharpoons y\}$, $\{x \rightleftharpoons\}$, or $\{\rightleftharpoons y\}$) to be submitted.

eligible than those offering an oversupplied tool type, because the withdrawal of an offer to relinquish a tool of zero scarcity would reduce the possibilities of cascade formation.

Commitment and eligibility are combined into the following protocol rules for pseudo-composition, where $S(y)$ denotes the total supply of tool type y.

1. Grant all proposals that involve the acquisition of a tool which is not undersupplied.

2. For each undersupplied tool type y, grant $S(y)$ proposals asking for that tool type in accordance with proposal eligibility, that is, in such a manner that no granted proposal is less eligible than any proposal that is not granted.

3. Modify each proposal $\{x \rightleftharpoons y\}$ that is not granted by substituting z for y, where z is chosen *randomly* from the bag of (oversupplied) tools.

Since the rules of the pseudo-composition protocol allow only elementary exchanges, an agent with a current endowment $\{a, a\}$ and a preference for the endowment $\{b, c, d\}$, can express its desire by communicating $\{aa \rightleftharpoons bcd\}$ in stage 1, but it must decompose this composite proposal into a bag of elementary ones in stage 4. This can be done in different ways. The proposal bags $\{\{a \rightleftharpoons b\}, \{a \rightleftharpoons c\}, \{\rightleftharpoons d\}\}$ and $\{\{a \rightleftharpoons\}, \{a \rightleftharpoons d\}, \{\rightleftharpoons b\}, \{\rightleftharpoons c\}\}$ are two of the many possible decompositions of $\{aa \rightleftharpoons bcd\}$. Because the probability that the entire proposal bag is accepted depends on the decomposition chosen, some decompositions are more sensible than others. Hence, an agent can profit from intelligent decomposition.

To prevent our illustration of the protocol's effectiveness from becoming too complicated, we let our DN agents behave intelligently only in a limited sense. They exhibit a form of bounded rationality that combines strategy-space limitation with strategy-space filtering. The limitation entails that they search for a good strategy to obtain their preferred reallocation *without* considering the possibility of aiming for anything else than this first preference. The filtering entails that they search for a good strategy (i.e., a decomposition of their composite reallocation proposal into elementary ones) by rejecting the decompositions whose inferiority can be determined without computing success probabilities, or inspecting their tool-bag utilities. They then choose randomly from the remaining strategies.

5.5.4 A pseudo-composition example

Tables 5.1a, and b provide an example of delegated negotiation on a simple composition problem. We do not specify the tool-bag utilities of the agents. The goals listed in the second column of Table 5.1a represent the agents' preferred exchanges. The agents exhibit the aforementioned form of bounded rationality, filtering out the obviously inferior proposal decompositions, and choosing randomly from the remaining ones.

agent	goal	strategy	outcome
1	$\{ac \rightleftharpoons bb\}$	$\{\{a \rightleftharpoons b\}, \{c \rightleftharpoons b\}\}$	$\{ac \rightleftharpoons bb\}$
2	$\{d \rightleftharpoons a\}$	$\{\{d \rightleftharpoons a\}\}$	$\{d \rightleftharpoons d\}$ or ...
3	$\{be \rightleftharpoons ac\}$	$\{\{b \rightleftharpoons a\}, \{e \rightleftharpoons c\}\}$	$\{be \rightleftharpoons ac\}$
4	$\{fhh \rightleftharpoons abe\}$	$\{\{f \rightleftharpoons a\}, \{h \rightleftharpoons b\}, \{h \rightleftharpoons e\}\}$	$\{fh \rightleftharpoons eg\}$ or ...
5	$\{bg \rightleftharpoons fh\}$	$\{\{b \rightleftharpoons f\}, \{g \rightleftharpoons h\}\}$	$\{bg \rightleftharpoons fh\}$

(a) Outcome of delegated negotiation, with given agent goals and strategies

tool	a	b	c	d	e	f	g	h
demand	3	3	1	0	1	1	0	1
supply	1	2	1	1	1	1	1	2
scarcity	2	1	0	-1	0	0	-1	-1

(b) The associated market profile

Table 5.1: Delegated negotiation on an simple composition problem.

The example problem is such that this bounded rationality is perfect (in the sense of Def. 5.1 on page 167) for all agents, except for agent 4. This agent must choose whether it will use its most valuable[7] asset (f) to increase its chances of acquiring a or b. Neither of these options is obviously inferior to the other, and we will assume that random selection results in agent 4 opting for a.

In view of the fact that the eligibility of $\{x \rightleftharpoons y\}$ equals that of $\{\rightleftharpoons y\}$ if x is oversupplied, and exceeds it otherwise, the strategy $\{\{x \rightleftharpoons y\}\}$ is never inferior to $\{\{x \rightleftharpoons\}, \{\rightleftharpoons y\}\}$. Consequently, agents 1 and 2 do not face much of a decision problem. Since the scarcity of c is zero, the protocol guarantees that agent 3 will obtain c. It has to worry only about a. Hence, the strategy $\{\{e \rightleftharpoons a\}, \{b \rightleftharpoons c\}\}$ is inferior to the strategy shown in Table 5.1a (since $\{b \rightleftharpoons a\}$ is more eligible than $\{e \rightleftharpoons a\}$). Agent 5 does not need to worry at all about its strategy. It is certain of acquiring f and h, since neither of these tool types is undersupplied.

The optimal solution of the general composition problem constituted by the agents' *goals* (the second column of Table 5.1) is the empty bag. In other words, *none* of the composite proposals (i.e., the agents' goals) is satisfied if one applies proposal

[7]Here, valuable means that the tool type is scarce, such that its use renders a highly eligible proposal.

composition. The elementary composition problem induced by the agents' *strategies* has an optimal solution consisting of the three proposals $\{a = b\}$ (from agent 1), $\{f = a\}$ (from agent 4), and $\{b = f\}$ (from agent 5). In contrast, the solution rendered by delegated negotiation satisfies 7 out of 10 elementary proposals and 3 of the 5 goals. As such, the example illustrates that the adaptation of proposals brought about by DN's pseudo-composition is an effective means of promoting trade.

The delegated-negotiation solution shown in Table 5.1 is not unique, but the *number* of satisfied proposals is the same for all possible outcomes. However, this is not a general property of DN. It is the case in this particular example, due to the fact that the proposals competing for a and b have different eligibilities. In general, the number of satisfied proposals can vary if DN is repeatedly applied to the same problem.

We cannot conclude from this example whether the outcome shown in Table 5.1 qualifies as a good solution, because we would have to know the agents' tool-bag utilities to weigh the utility gains obtained by agents 1, 3, and 5 against a possible utility loss suffered by agent 4. Such evaluation will be performed later, in Chapter 7.

5.5.5 Distributed pseudo-composition

We have described the pseudo-composition protocol as a decision process performed by the auctioneer. However, the involvement of the auctioneer is not required, once the market profile has been computed. In the following, we describe how the protocol can be effectuated by the agents themselves in a distributed fashion.

At first sight, one might expect that, once the market profile information is made available to the agents, the remaining problem can be solved in a Contract-Net manner (Smith, 1980). This is not entirely true. Distributed pseudocomposition amounts to the execution of several *related* contracting processes in parallel. This requires more message traffic than the execution of a comparable number of *independent* contracting processes in a Contract Net.

The distributed pseudo-composition in stage 4 of delegated negotiation involves five types of messages, three of which (the buy bids) are sent from buyers to sellers, the other two (sell bids) in the opposite direction. The contents and purpose of message types are summarized in Table 5.2.

Below, we provide some explanation of the table by sketching the message traffic that is required to sell the tools of an *undersupplied* tool type. A complete and detailed account of the entire distributed pseudo-composition process is presented in Appendix A.

The prospective buyers of the undersupplied tool type initiate the negotiation process by sending option messages to *all* agents that possess the tool type. These option messages contain — among other data — information on the eligibility of the buyer, to enable a seller (who possesses one or more tools of the desired type) to select the most eligible buyers. Election messages are then sent by the seller to these selected buyers. Because message traffic takes place in parallel, and a prospective buyer sends option messages to *all* sellers of the desired tool type, the number of tool offers (i.e., election messages) received by highly eligible buyers will generally exceed the number of tools they require, and the converse will happen with buyers of low eligibility. Hence, additional message traffic is required to tie all sellers to buyers. A buyer that has received election messages from sellers sends *one* commitment message to some seller for each required tool, and it sends retraction messages to all other sellers. A seller that receives a retraction message removes the associated buyer from its list of interested buyers, and sends an election message to the next most eligible buyer. And so forth. Once a seller has sold all of its tools (i.e., once it has received one commitment message for each tool), it sends rejection messages to any buyers that are still waiting for a response to their option messages.

type	**class**	**contents**	**semantics**
option	buy bid	sender, type, option-id, tool type, eligibility	offer to buy a tool
election	sell bid	sender, type, option-id	election of an option
rejection	sell bid	sender, type, option-id	rejection of an option
commitment	buy bid	sender, type, option-id	commitment to an option
retraction	buy bid	sender, type, option-id	retraction of an option

Table 5.2: The five message types of distributed pseudo-composition.

Apart from the information an agent has received from the auctioneer (scarcity and demand of tool types they offer, and seller lists of tool types they require), selling agents keep track of the number of options received for tool types with positive scarcity, the number of tools not yet committed (for each tool type they offer), as well as the number of unanswered election messages sent out. Buying agents keep track of the number of tools they still have to acquire for each tool type.

The above description presumes that the agents are truthful in their communication. If necessary, their truthfulness can be verified afterwards by letting the sellers communicate an account of their deliveries to the auctioneer, who can then perform a checksum-like audit.

Since the auctioneer does not participate in stage 4 of delegated negotiation (cf. Fig. 5.2), it can perform other tasks, while the agents are engaged in pseudo-composition. In the context of a large number of agents, one can make use of this to decompose the overall problem in the following manner. Dividing the agents into groups, each group is assigned a separate auctioneer. While the group members solve their composition problem, the auctioneer engages in a similar negotiation with other auctioneers, coordinated by a *higher-level* auctioneer. A base-level auctioneer can provide the necessary information to *its* auctioneer, because the tool scarcities computed in stage 3 (see Fig. 5.2) determine completely which tools will be left over after solution of the local composition process by the agents it manages. It can thus try to trade these tools within the community of base-level auctioneers. The agents who saw (some of) their proposals rejected can profit from the outcome of the higher-level pseudo-composition if they engage in a second attempt to reallocate.

Hence, the use of a distributed protocol facilitates the decomposition of large-scale problems. However, there is a price tag attached to such decentralization: The distributed protocol requires considerably more communication than the centralized one.

message type:	from:	to:	number of messages
option	1,2,3,4,5	6,7,8	15
election	6,7,8	1	3
commitment	1	6	1
retraction	1	7,8	2
rejection	6	2,3,4,5	4
election	7,8	2	2
commitment	2	7	1
retraction	2	8	1
rejection	7	3,4,5	3
election	8	3	1
commitment	3	8	1
rejection	8	4,5	2
Total number of messages:			36

Table 5.3: Required message traffic to assign 3 tools among 5 agents.

This is illustrated in Table 5.3. The table lists the message traffic involved in the distributed assignment of three tools of some undersupplied type x, currently possessed by agents 6, 7, and 8 among five prospective buyers, say, agents 1, 2, 3, 4, and 5, with respective eligibilities $e_1 > e_2 > e_3 = e_4 = e_5$. If the protocol were centralized, the auctioneer would be able to define an assignment by telling agents 6, 7, and 8 to which agent they should transfer their x, that is, by sending 3 messages. In contrast, assignment via the distributed protocol entails the transmission of 36 messages, as shown in Table 5.3.

The table is block-chronological. Messages that are in the same block (i.e., not separated by horizontal lines) do not depend on each other, and can therefore occur in an arbitrary temporal order. If two messages are in different blocks, the message in the uppermost block is sent and received before the other message is sent. The message flow depicted in the table is an arbitrary possible outcome of the nondeterministic negotiation process. Senders or receivers printed in italics result from nondeterministic selections. In this particular case, this involves only two receivers: agent 6 in the third block, and agent 7 in the fifth.

5.6 Informed Gambling: A Class of Iterative Mechanisms

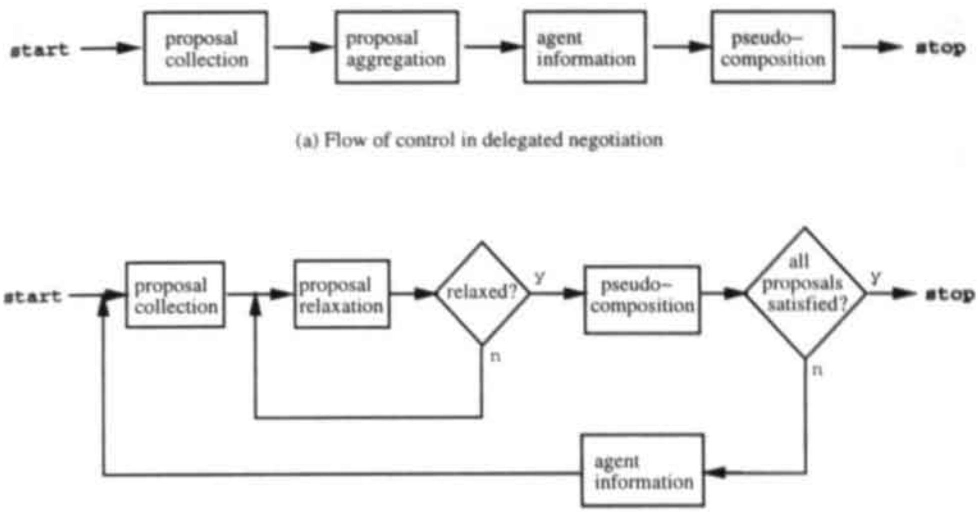
5.6.1 From DN to IG: the incorporation of relaxation

The main difference between DN and IG is that, in IG, pseudo-composition is preceded by proposal relaxation. This relaxation amounts to repeated execution of the DN stages of proposal collection, proposal aggregation, and agent information (stages 1, 2 and 3 in Fig. 5.2). In the first iteration of this cycle, IG does not differ from DN. Each agent communicates its preferred exchange in terms of a composite proposal (stage 1). The auctioneer aggregates these proposals into a market profile of gross demand and supply (stage 2), and communicates this profile to the agents (stage 3).

The relaxation in IG takes place in subsequent iterations of this cycle. The agents can now use the information in the market profile, and their knowledge of the pseudo-composition protocol to perform some analysis with respect to the bag of elementary proposals they should submit. This analysis may lead to the submission of a proposal bag that is a decomposition of their initial, composite proposal, but it can also involve an adaptation of this proposal. The auctioneer aggregates the newly communicated proposal bags into a new market profile, and transmits this profile to the agents again. This three-stage process is repeated until all agents stick to their proposals. The pseudo-composition protocol is then invoked to implement a reallocation.

Experiments with a prototypical IG mechanism pointed out that it is often worthwhile to allow agents which are dissatisfied with the outcome to engage in a renewed reallocation attempt. The associated flow of control is depicted in Fig. 5.3(b). Obviously, it is more complex than the flow of control in delegated negotiation. Solving a reallocation problem by means of DN comprises a single application of a single-shot mechanism, while solving it by means of IG entails the (generally repetitive) application of an iterative mechanism.

We refer to a single application of the iterative mechanism in IG as a *phase*. Each



(a) Flow of control in delegated negotiation

(b) Flow of control in Informed Gambling

Figure 5.3: Flow of control in DN and in IG.

phase consists of a number of relaxation rounds, followed by pseudo-composition. Hence, in terms of Fig. 5.3b, a phase comprises the traversal of the action blocks between start and stop, without entering the *outer* loop. The relaxation block corresponds with a sequence of three blocks in Fig. 5.3a, namely, proposal aggregation, agent information, and proposal collection (in that order). The relaxation in the inner loop constitutes the adjustment process of the iterative mechanism, while the pseudo-composition protocol constitutes its outcome correspondence.

If some of the proposals are rejected in the pseudo-composition of the first phase, the process is repeated with the agents associated with these proposals. This is indicated by the outer loop in Fig. 5.3. Even though such a renewed attempt typically involves only a small fraction of the original agent population, it tends to enhance the quality of the ultimate outcome — in terms of average agent satisfaction — considerably.

Proposal collection in subsequent phases differs from that in the first phase in that the agents are guided by an additional market profile from the auctioneer, based on the last submitted proposals of the remaining agents. This explains the agent-information block in the outer loop of Fig. 5.3. The combination of agent information and proposal collection amounts to proposal *revision*, as defined in Sect. 3.2.2. The entire IG auction ends if a phase terminates with the satisfaction of all last-submitted proposals.[8]

[8]This includes the case in which all the remaining agents pass (by submitting empty proposal bags).

5.6.2 Informed-Gambling mechanisms

Informed Gambling constitutes a *class* of mechanisms for Tool Reallocation (TR). In view of the broadness of the TR problem domain, it is not realistic to expect any single mechanism to be adequate for all kinds of TR problems. There are, however, certain common elements in IG mechanisms. Apart from the above flow-of-control schema, this involves the following characteristics.

An IG mechanism can be viewed as an auction, in which the coordination module plays the role of auctioneer. However, an IG auctioneer should be thought of as a *device*, rather than an agent. Its main task is to aggregate incoming reallocation proposals into a market profile, and reveal this to the agents. In addition, the auctioneer executes the rules of the pseudo-composition protocol described in Sect. 5.5. As such, its response behavior is straightforward in pseudo-composition, and extremely simple in relaxation, where it does little more than adding numbers. In any case, an it is much less sophisticated than the auctioneers responsible for the — often complex — tâtonnement schemes in computational implementations of Walrasian-like auctions (Scarf, 1973; Joosten, 1996).

The converse is true for the traders in IG. More so than the traders in a Walrasian auction, whose response to price information is straightforward, because they do not face any uncertainties, the traders in an IG auction resemble human decision makers: They search for a good response by comparing alternative strategies in a manner that — even in simple domains such as tool reassignment — involves nontrivial situation assessment.

In view of the above, it is appropriate to state that the relaxation in IG is performed by the *agents*, who respond to the market profiles revealed by the auctioneer in a manner that is rational, in view of their tool-bag utilities and the decision rules employed in pseudo-composition.

Below, we present the formal, general definition of "IG mechanism". Due to its generality, the definition is of an abstract nature. It does not entail a precise formulation of all that goes on within an IG mechanism. Components like the agent response correspondence, the termination criterion for the relaxation process, and the definition of eligibility are discussed, but not specified in full detail. Such specification is possible, but the details depend on the nature of the application domain. In Sect. 5.6.5, we provide a detailed specification of the agent response correspondence for the tool-reassignment domain.

The formal definition of IG uses the terminology and notation introduced in Chapter 2. For the reader's convenience, a survey of the semantics of the symbols occurring in Def. 5.2 is presented in Table 5.4.

Concerning Table 5.4, we make the following remarks.

symbol	semantics	definition
I	agent community	a finite set
R	set of tool types	a finite set
$\mathfrak{B}(R)$	universe of tool bags	the set of finite bags over R
λ	tool allocation	$\lambda : I \to \mathfrak{B}(R)$
Γ	community tool bag	$\Gamma \triangleq \underline{\downarrow} \mathit{Im}\, \lambda$
$\mathcal{P}(\Gamma)$	the power bag of Γ	$\{\gamma \in \mathfrak{B}(R) : \gamma \subseteq \Gamma\}$
u	agent level utility (for tool bags)	$u : \mathcal{P}(\Gamma) \to [0, 1]$
U	universe of agent level utilities	...
ρ^+	demand bag of the gbag ρ	$\rho^+(r) \triangleq \max(0, \rho(r))$
ρ^-	supply bag of the gbag ρ	$\rho^-(r) \triangleq \max(0, -\rho(r))$
$\mathcal{G}(R)$	universe of reallocation proposals	the set of finite gbags over R
$\mathcal{G}_1(R)$	universe of elementary proposals	$\mathcal{G}_1(R) \triangleq \{\rho \in \mathcal{G}(R) : \|\rho^+\| \leq 1 \wedge \|\rho^-\| \leq 1\}$
ψ	proposal(-bag) profile	$\psi : I \to \mathfrak{B}(\mathcal{G}_1(R))$
Ψ	universe of proposal profiles	...
μ	market profile (a pair of bags)	$\mu \triangleq \langle \mu^-, \mu^+ \rangle \triangleq \langle \biguplus_{i \in I} \psi(i)^-, \biguplus_{i \in I} \psi(i)^+ \rangle$
σ	scarcity profile (a gbag)	$\sigma \in \mathcal{G}(R)$, $\sigma \triangleq \mu^+ - \mu^-$, $\sigma \triangleq \underline{\downarrow} \mathit{Im}\, \psi$

Table 5.4: Overview of symbols used in the formal definition of IG.

- The symbol u denotes the normalized (i.e., relative-utilitarian) agent-level utility as defined in Def. 2.32 on page 39.
- The bags μ^- and μ^+ denote *gross* supply and demand, while σ^- and σ^+ denote *net* supply and demand (i.e., oversupply and undersupply). Note that, while $\mu = \langle \mu^-, \mu^+ \rangle$ denotes a pair of bags. σ is a gbag, in which the negative multiplicities denote excess tool supply, and the positive ones denote excess tool demand.
- In the sequel, we sometimes add an index to μ, μ^- or μ^+ to indicate the round number. Thus, μ_i^- denotes the gross supply associated with the proposal profile ψ_i submitted in round i.
- It is left to the reader to check that the two alternative definitions $\sigma \triangleq \mu^+ - \mu^-$ and $\sigma \triangleq \underline{\downarrow} \mathit{Im}\, \psi$ are consistent.

An IG mechanism is an iterative mechanism (see page 163)

$$\langle \mathcal{M}_A, \mathcal{M}_C, E, Z, C, T, h, f \rangle, \text{ where}$$

- The agent message space $\mathcal{M}_A$ is $\mathfrak{B}(\mathcal{G}_1(R))$, the set of bags of elementary proposals involving tool types of some set R.
- The coordination message space is $\mathfrak{B}(R) \times \mathfrak{B}(R)$. A coordination message is a market profile, a pair of bags of tool types in R, which describe the current supply of and demand for tools in the agent community.

- The problem space E is a set of tool reallocation problems.
- The outcome space Z is a set of proposal-bag profiles. An outcome is a proposal-bag profile, that is, a function $z : I \to \mathfrak{B}(\mathcal{G}_1(R))$, which specifies the tool exchanges to be executed on the endowments of the agents.
- The message-aggregation function C computes a market profile μ (consisting of a demand bag μ^- and a supply bag μ^+) by aggregating the proposals of a proposal profile ψ_i, in the manner indicated by the definition of μ in Table 5.4.
- The termination criterion T varies between IG mechanisms, and is represented by a parameter $\mathfrak{T}$ in a mechanism representation.
- The outcome correspondence h is defined by the pseudo-composition protocol of delegated negotiation with the eligibility specification $\mathfrak{E}$ as a parameter.
- The agent response correspondence f is based on fictitious rationality, a behavior characteristic that we will discuss shortly. However, the precise definition of agent response varies between IG mechanisms, and is represented by the parameter $\mathfrak{R}$ in a mechanism representation.

Although IG mechanisms apparently fit into our general model of iterative mechanisms, we do not use the general model to distinguish different IG mechanisms. This would be unnecessarily verbose, since most of these constituents of the iterative mechanism model are the same for all IG mechanisms. Hence, we represent specific IG mechanisms as a triplet of constituents that do vary.

Definition 5.2 (IG mechanism)
An IG mechanism is a triplet $\langle \mathfrak{R}, \mathfrak{E}, \mathfrak{T} \rangle$, where

- $\mathfrak{R} : \mathfrak{B}(R) \times U \times \mathfrak{B}(R)^2 \times \mathfrak{B}(\mathcal{G}_1(R)) \times \mathbb{N} \to \mathcal{P}(\mathfrak{B}(\mathcal{G}_1(R)))$ *is an agent response correspondence, that maps a tuple consisting of an endowment, an agent-level utility function, a market profile, a previously submitted proposal bag, and a relaxation round number to a set of proposal bags. The idea is that every* $\beta_i \in \mathfrak{R}(\varepsilon, u, \mu_{i-1}, \beta_{i-1}, i)$ *denotes a proposal bag that a rational agent with endowment* ε*, and tool bag utilities* u *may communicate in round* i*, if* μ_{i-1} *is the market profile aggregated from the proposals communicated in the previous round, in which the agent itself has communicated* β_{i-1}*.*
- $\mathfrak{E} : \mathcal{G}(R) \to \text{PREORD}(R)$ *is an eligibility specification.*[9] $\mathfrak{E}$ *maps a scarcity profile* σ *to an eligibility preorder* $\succsim$ *on tool types.* $\mathfrak{E}(\sigma)$ *denotes the eligibility preorder* $\succsim$ *that is defined on the set* R *of tool types if* σ *is the current scarcity profile. Scarcer tool types tend to render higher eligibility, that is,* $\sigma(r_1) > \sigma(r_2) \Rightarrow r_1 \succsim r_2$.

[9] PREORD(R) denotes the set of preorders definable on the set R, allowing element x of R to be either more eligible, or less eligible, or equally eligible than another element y of R.

- $\mathfrak{T} : \Psi \times \Psi \times \mathbb{N} \rightarrow$ {False, True} *is the termination condition for the relaxation stage.* $\mathfrak{T}(\psi_i, \psi_{i-1}, i)$ *denotes the truth value of termination after round* i, *if* ψ_i *is the proposal profile of round* i, *and* ψ_{i-1} *that of the previous round.*

△

Def. 5.2 harbors a number of constraints for IG mechanisms that deserve to be mentioned explicitly.

- The definition of $\mathfrak{R}$ constrains the information that agents use to determine an appropriate response. It tells us, for example, that they do not use the information embodied in earlier market profiles than the last communicated one. Metaphorically speaking, an IG agent forgets a market profile as soon as it receives a new one. As a consequence, IG agents cannot compare two subsequent market profiles, and are therefore generally not able to assert whether the current round will be the last one, even if they know that the termination criterion is $\psi_i = \psi_{i-1}$.

  The definition of $\mathfrak{R}$ also tells us that IG agents do remember their previous proposal while they deliberate their current one. Hence, if stationarity of the adjustment process is the sole criterion for its termination (i.e., if $\mathfrak{T} \triangleq (\psi_i = \psi_{i-1})$), IG agents *can* conclude that the current round will certainly not be the last one, if they submit a proposal bag that differs from their previous submission. In principle, a self-motivated agent can use this fact to attempt the conception of a plan (i.e., a sequence of proposal-bag submissions) to lead the adjustment process to a state in which the agent's optimal proposal(s) are more eligible than they presently are. Again, the definition of $\mathfrak{R}$ tells us that IG agents do not exhibit such strategic behavior; their response is always tropistic: Except for the agent's last-submitted proposal bag, neither the past, nor the future of the adjustment process play a role in the determination of the agent's present response.

- If $\succsim$ denotes the eligibility preorder $\mathfrak{E}(\sigma)$, then $\succsim$ is defined between any two tool types in D_σ (i.e., any tool types currently allocated to some agent within the community), but it is generally not an order. Obviously, any tool types with equal scarcities are equally eligible, but even tool types with different scarcities can be equally eligible. The variation among eligibility specifications in IG mechanisms lies in the extent to which the latter is the case. At one extreme, the eligibility specification $\succsim_0$, defined by $r_1 \succ_0 r_2 \Leftrightarrow sign(\sigma(r_1)) > sign(\sigma(r_2))$, only distinguishes between oversupplied, undersupplied, and zero-scarcity tool types. At the other extreme, the eligibility specification $\succsim_1$, defined by $r_1 \succ_1 r_2 \Leftrightarrow \sigma(r_1) > \sigma(r_2)$ distinguishes between any two tool types with different scarcities. An example of an eligibility specification between these two extremes is $\succsim_\bullet$, defined by $r_1 \sim_\bullet r_2$ if both

r_1 and r_2 are oversupplied, and $r_1 \succsim_* r_2 \Leftrightarrow r_1 \succsim_1 r_2$ otherwise. This eligibility specification distinguishes between oversupplied tool types, zero-scarcity tool types, tool types of scarcity 1, tool types of scarcity 2, ...

- The fact that an eligibility specification $\mathfrak{E}$ maps a scarcity profile to a preorder on R rather than on $\mathfrak{B}(R)$ implies that IG is only able to deal with *elementary* exchange proposals. This is a vital restriction. The incentives embodied in the definition of proposal eligibility cannot be generalized from elementary proposals (elements of $\mathcal{G}_1(R)$) to general ones (elements of $\mathcal{G}(R)$), without hampering the agents severely in their estimation of the probability that a proposal will be executed.
- The motivation behind the definition of the termination criterion $\mathfrak{T}$ as a function of ψ_i, ψ_{i-1}, and i lies in the following three valid[10] reasons to terminate the relaxation process.
    1. The relaxation process reaches a stationary state, that is, $\psi_i = \psi_{i-1}$.
    2. The relaxation has led the agents to a proposal profile ψ_i that is completely satisfiable, that is, $\sigma_i \triangleq \underline{\downarrow} \mathit{Im}\, \psi_i = \emptyset$.
    3. The relaxation has been going on for too long. The round number i has reached a previously imposed deadline value, and the relaxation is broken off.

5.6.3 Fictitiously-rational agent response

To implement relaxation in IG, we need to specify how the relaxation incentives embodied in IG's pseudo-composition protocol affect the agents' responses.

Complex TR problems call for another kind of agent response than simple problems. Whereas an (almost) perfectly-rational response may be feasible in the context of a simple TR domain such as tool reassignment, it is most likely too demanding for TR problems associated with complex, real-life optimization, such as transportation and rostering problems. A boundedly-rational agent response will usually be called for in such cases. However, as apparent from our discussion of bounded rationality in Sect. 5.4.2, some forms of bounded rationality (e.g., strategy-space limitation/expansion) come down to applying (almost) perfect rationality to a suitably confined portion of the agent's strategy space. We stipulate that such a scheme is a feasible option for IG mechanisms designed to tackle complex TR problems.[11] Hence, in the sequel, we present an almost-perfect form of Bayesian rationality that can serve as a definition of agent response in IG mechanisms for tool reassignment,

[10] In Chapter 7, we experiment with these termination conditions to assess their validity.

[11] We will provide an example in Sect. 5.9.1.

and as a basis for the definition of a boundedly-rational response for (at least some) complex TR domains.

The response correspondence we propose to attain relaxation in IG mechanisms for tool reassignment is a Bayesian variant of Brown-Robinson fictitious play, a dynamic process that can be used to approximate mixed-strategy Nash equilibria (Brown, 1951; Robinson, 1951; Jordan, 1993). In fictitious play processes, an agent's strategy in round k is a weighted average of the currently optimal strategy and the strategies chosen in the previous m rounds, for some $m > 0$. In our Bayesian variant, $m = 0$. We refer to the agent rationality that is associated with such a response correspondence as *fictitious rationality*.

The general idea of fictitious rationality is similar to that of Nash equilibrium. It is reflected by the following 'agent instruction'.

> "Select, in each round, a strategy that maximizes the conditional expectation of your outcome utility, under the condition that each of the other agents sticks to its last-chosen strategy."

Agents in an iterative mechanism do not know the actual strategies (i.e., the responses) of other agents. They only know the coordination message composed from these responses. As a consequence, agents who exhibit fictitious rationality within an iterative mechanism actually employ a weaker assumption. Their behavior is consistent with the following instruction.

> "Submit, in each proposal round t, a proposal v_i^t that maximizes the conditional expectation of your outcome utility, under the condition that the responses of the the other agents are such that they, in combination with your previous proposal, would lead to the most recently received coordination message being sent again."

If we denote the utility of the outcome z for agent i by $u_i(z)$, and the conditional expectation of x, given y, by $E\{x|y\}$, the above description of fictitious rationality can be expressed formally, in terms of the symbols used in our definition of iterative mechanisms as

$$f_i(e_i, m^t, v_i^t, t) \in \{s \in S_i \mid s \text{ maximizes } E\{u_i(\bar{h}(s, v_{\bar{i}}^t)) \mid m^t = C(s, v_{\bar{i}}^t)\}\ \} \quad (5.6)$$

Eq. 5.6 features $\bar{h}$ instead of h, the symbol for the outcome correspondence in our model of iterative mechanisms, because the expression $E\{u_i(h(s, v_{\bar{i}}^t))| \ldots\}$ is not well-defined. Since h represents the outcome *correspondence*, the expression $h(s, v_{\bar{i}}^t)$ denotes a *set* of outcomes. Hence, the only reasonable interpretation of $u_i(h(s, v_{\bar{i}}^t)$ is that of the set of utilities associated with these outcomes. However, for $E\{x| \ldots\}$ to make sense, x must be a stochastic variable. This implies that, to define fictitious rationality properly, we must turn the nondeterministic mapping $h : S \to \mathcal{P}(Z)$ into a *stochastic* mapping $\bar{h} : S \to Z$ (i.e., a function $\bar{h}$ that maps each $s \in S$ to a *stochastic* variable $\bar{h}(s)$). The method to arrive at such a stochastic mapping depends on the kind of mechanism. In the next section, we explain how one can obtain "the natural stochastic equivalent" of a nondeterministic mapping for IG mechanisms, by applying the minimal-rationality principle.

5.6.4 The estimation of expected utilities

IG agents are supposed to use their knowledge of the protocol, together with the market-profile information, to estimate the expected utilities of proposals. However, the protocol is a nondeterministic, procedural description of pseudo-composition. To enable the agents to compute expected utilities of alternative strategies, a more informative description is required. Such a description is provided below by Prop. 5.4, which transforms the pseudo-composition protocol into a probabilistic input-outcome mapping.

For the reader's convenience, we recapitulate the pseudo-composition protocol of delegated negotiation (and IG) from Sect. 5.5.3.

Definition 5.3 (Pseudo-composition protocol)
Pseudo-composition comprises the following steps, to be executed by the auctioneer in the order listed.

1. *Execute all proposals that involve acquisition of a tool which is not undersupplied.*
2. *For each undersupplied tool type y, execute $\mu^-(y)$ proposals asking for that tool type in accordance with proposal eligibility, that is, in such a manner that none of the executed proposals are less eligible than any proposal that is not executed.*
3. *Modify any proposal that is not executed in the previous steps, by substituting z for y, where z is drawn randomly from the (oversupplied) tools that are still available.*
4. *Execute the modified proposals.*

△

Steps 2 and 3 of the protocol involve nondeterminism. Step 2 does not specify which of two competing proposals will be executed in case of equal eligibility. Step 3 encompasses a random selection from the oversupplied tools. To turn the nondeterministic procedure into a probabilistic mapping, we employ — putting ourselves in the position of an IG agent — the principle of minimal rationality. In this case, this leads to two assumptions. First, we assume that the selection of proposals to be executed is *impartial*, that is, the probability that a proposal is executed depends *only* on its eligibility. This implies that proposals of equal eligibility have equal success probabilities.

Second, we assume that the probability of receiving z instead of y (in the case that y is undersupplied) is *proportional* to the multiplicity of z in the bag of excess oversupplied tools.

These two assumptions lead to a description — presented in Prop. 5.4 below — of the input-outcome mapping in terms of a probability distribution over the elementary exchanges that can take place. We denote the event that some agent proposes $\{x \rightleftharpoons y\}$ by $\text{prop}(x \rightleftharpoons y)$, and the event that this proposal is executed on the endowment of the agent by $\text{exec}(x \rightleftharpoons y)$.

Proposition 5.4 (objective proposal success probability)
Let P_y denote the conditional probability that the proposal $\{x \rightleftharpoons y\}$ is executed, given the proposal profile ψ and the eligibility definition $\mathfrak{E}$.[12] *Formally,*

$$P_y \triangleq P\{\text{exec}(x \rightleftharpoons y) \mid \text{prop}(x \rightleftharpoons y), \psi, \mathfrak{E}\}$$

Let P_z denote the probability that the proposal is rejected, and $\text{exec}(x \rightleftharpoons z)$ takes place instead of $\text{exec}(x \rightleftharpoons y)$. Formally,

$$P_z \triangleq P\{\text{exec}(x \rightleftharpoons z) \mid \text{prop}(x \rightleftharpoons y), \psi, \mathfrak{E}\}$$

Let γ_y denote the bag of tools offered by agents in exchange for y, that is, $\gamma_y \triangleq \{r \in R \mid |\{r \rightleftharpoons y\} \in Im\,\psi\}$. Let γ_-, γ_0, and γ_+ denote the subbags of γ_y containing the tools that are less eligible, equally eligible, and more eligible than x, that is

$$\begin{aligned} \gamma_- &\triangleq \{r \in \gamma_y | r \prec x\} \\ \gamma_0 &\triangleq \{r \in \gamma_y | r \sim x\} \\ \gamma_+ &\triangleq \{r \in \gamma_y | r \succ x\} \end{aligned}$$

Let q denote what is left of the supply $\mu^-(y)$ of y after all proposals $\{X \rightleftharpoons y\}$ that are more eligible than $\{x \rightleftharpoons y\}$ have been executed. Formally,

$$q \triangleq \mu^-(y) - |\gamma_+|$$

Then

[12] The proposal basically aims to cover only the case in which the multiplicity of $\{x \rightleftharpoons y\}$ in $Im\,\psi$ is one. If there are multiple proposals $\{x \rightleftharpoons y\}$ in $Im\,\psi$, P_y denotes the probability that a *specific* one of these proposals is executed, not the probability that at least one of them is executed.

(i) If $\sigma(y) \leq 0$ then $P_y = 1$ and hence, $P_z = 0$ for all $z \neq y$.

(ii) If $\sigma(y) > 0$ then

$$\begin{array}{lll} P_y = & 0 & \text{if } q \leq 0 \\ P_y = & 1 & \text{if } q \geq |\gamma_0| \\ P_y = & \frac{q}{|\gamma_0|} & \text{if } 0 < q < |\gamma_0| \end{array}$$

(iii) If $\sigma(y) > 0$ then

$$P_z = (1 - P_y) \cdot \frac{\sigma^-(z)}{|\sigma^-|} \text{ forall } z \neq y$$

Proof.
The proposition is not more than a transcription of the procedural description of the nondeterministic protocol in terms of outcome probabilities, under the assumptions of impartiality and proportionality. Hence, we merely indicate which steps of the protocol are used to arrive at the above statements.

Statement (i) in the proposition follows immediately from step 1 of the protocol. Statement (ii) follows from steps 1 and 2, together with the assumption of impartiality. Statement (iii) is a consequence of steps 1, 2, and 3, together with the assumption of proportionality. ■

The outcome probabilities P_y and P_z in Prop. 5.4 represent the probabilities as they would be perceived by an *omniscient* observer. The agents, however, are not omniscient. Their knowledge of other agents' proposals is confined to what they can derive from the market profile and their knowledge of their own proposals in the last round. Since they cannot determine γ_y (or its derivatives γ_-, γ_0 and γ_+) from μ, they are unable to compute P_y or P_z as in Prop. 5.4. However, they can compute a subjective estimate $\widehat{P}_y$ of P_y based on their knowledge of the market profile μ. This estimate, in turn, can be used to compute $\widehat{P}_z = (1 - \widehat{P}_y) \cdot \frac{\sigma^-(z)}{|\sigma^-|}$.

Let us look at the estimation of $\widehat{P}_y$ by an agent that has submitted $\{x = y\}$ in the previous round, and is deliberating to submit the same proposal again. In the following, we refer to this agent as "the estimator". The estimator exhibits minimal rationality in two respects.

1. Since it has no indication as to what the other agents may propose in the current round, except for the market profile computed from their proposals in the previous round, it assumes that their response in the current round will be such that the same market profile will be the same, if the estimator itself re-submits its previous proposal.

2. To assess the (subjective) success probability of its proposal, the estimator hypothesizes which tool types are offered by its (unknown) competitors (i.e.,

the agents interested in the same tool type). Again, the market profile is its only clue in this respect, so it assumes a probability distribution over the possible bags of tool types offered by their competitors based on the relative frequencies of tool types in the supply part of the market profile.

As a first step toward calculation of $\widehat{P}_y$, the estimator can conclude that the tools which *may* be offered in exchange for y in the competing proposals are the tools in the bag

$$\hat{\gamma} \triangleq \{r \in \mu^- \mid r \neq y\} \setminus \{x\} \tag{5.7}$$

We refer to this bag as the bag of potential adversaries (to x). To arrive at a concise formula for $\widehat{P}_y$, we attach some symbols to relevant quantities that can be derived from $\hat{\gamma}$ and μ.

l: the supply of y
$(l \triangleq \mu^-(y))$

k: the number of actual adversaries, that is, the number of competing proposals
$(k \triangleq \mu^+(y) - 1)$

s: the number of potential adversaries
$(s \triangleq |\hat{\gamma}|)$

t: the number of potential adversaries that are less eligible than x
$(t \triangleq |\{r \in \hat{\gamma} \mid r \prec x\}|)$

v: the number of potential adversaries that are equally eligible as x
$(v \triangleq |\{r \in \hat{\gamma} \mid r \sim x\}|)$

w: the number of potential adversaries that are more eligible than x
$(w \triangleq |\{r \in \hat{\gamma} \mid r \succ x\}|)$

A formula for $\widehat{P}_y$ in terms of these symbols is specified in Prop. 5.5 below. Here, $\widehat{P}_y$ denotes the subjective success probability of $\{x = y\}$ where this is the *only* proposal to be submitted by the estimator, and the same proposal that was submitted in the previous round. The general case, in which the estimator deliberates the submission of multiple proposals, not necessarily the same as those submitted in the previous round, will be discussed later.

Proposition 5.5 (subjective success probability of an isolated proposal) *Consider an agent that has submitted $\{x = y\}$ in the previous round, deliberates repeating this proposal in the current round, and calculates a subjective estimate $\widehat{P}_y$*

of the success probability of the proposal on the basis of the market profile. Let k, l, s, v, and w be defined as above. Define

$$\binom{n}{m} \triangleq 0 \text{ if } n < m. \tag{5.8}$$

Then the subjective success probability of proposal $\{x \rightleftharpoons y\}$ equals

$$\widehat{P}_y = \frac{\sum_{i=0}^{l-1} \binom{w}{i} \cdot \sum_{j=0}^{k-i} \binom{v}{j} \binom{s-w-v}{k-i-j} \cdot \min\left(1, \frac{l-i}{j+1}\right)}{\binom{s}{k}} \tag{5.9}$$

Proof.
Let W, V, and T denote the actual values associated with w, v and t. In view of the definitions of γ_+, γ_0, and γ_- in Prop. 5.4, and the difference between γ_y in Prop. 5.4 and $\hat{\gamma}$ in Eq. 5.7,

$$\begin{aligned} W &= |\gamma_+| \\ V &= |\gamma_0| - 1 \\ T &= |\gamma_-| \end{aligned}$$

To the estimator, who has no knowledge of these actual values, W, V, and T are stochastic variables. Let $P_{i,j}$ reflect the probability distribution which the estimator attributes to the triplet $\langle W, V, T\rangle$. Formally,

$$P_{i,j} \triangleq P\{W = i \wedge V = j \wedge T = k - i - j \mid w, v, t\} \tag{5.10}$$

For the estimator, it is as if the relative eligibilities of its adversaries are determined by a stochastic experiment. The estimator has just drawn a violet ball from a vase. Its k adversaries stand next to the vase which now contains w white balls, v violet balls, and t terra-cotta balls, and each adversary draws a ball from the vase (without replacement). In other words, in view of the information available to the estimator, it will attribute a hypergeometrical distribution to $\langle W, V, T\rangle$:

$$P_{i,j} = \frac{\binom{w}{i}\binom{v}{j}\binom{s-v-w}{k-i-j}}{\binom{s}{k}} \tag{5.11}$$

Note that, grace to (5.8), Eq. 5.11 defines $P_{i,j}$ for all nonnegative values of i and j such that $i + j \leq k$ (that is, even if $i > w$ or $j > v$). Note also, that the same holds if we replace k by l: since y is undersupplied, its demand $k + 1$ exceeds its supply l, so $l \leq k$.

With regard to statement (ii) in Prop. 5.4, we note that, if $i = |\gamma_+|$ and $j = |\gamma_0| - 1$, then $q = l - |\gamma_+| = l - i$, and hence, the objective success probability P_y can be defined in terms of i and j as

$$P_y^{i,j} = \begin{array}{ll} 0 & \text{if } i \geq l \\ 1 & \text{if } i < l \wedge l - i \geq j + 1 \\ \frac{l-i}{j+1} & \text{if } i < l \wedge l - i < j + 1 \end{array}$$

This can be expressed more concisely as

$$P_y^{i,j} = \begin{array}{ll} 0 & \text{if } i > l \\ \min(1, \frac{l-i}{j+1}) & \text{if } 0 \leq i < l \end{array}$$

Thus, we have obtained an expression for $P_y^{i,j}$ that is valid for any nonnegative i and j. The expression for $P_{i,j}$ in Eq. 5.11 is valid for any nonnegative i and j such that $i + j \leq k$, and we have observed that $l \leq k$. Consequently, we can combinatorialize $\widehat{P}_y$ over $P_{i,j}$ to obtain

$$\widehat{P}_y = \sum_{i=0}^{l-1} \sum_{j=0}^{k-i} P_{i,j} \cdot P_y^{i,j} \tag{5.12}$$

Substitution of the expressions for $P_{i,j}$ and $P_y^{i,j}$ in this equation renders Eq. 5.9.

■

For the derivation of Eq. 5.9, we have assumed that $\{x \doteq y\}$ is the only proposal submitted by the estimator, and that it submitted the same proposal in the previous round. If the estimator submits a single proposal that *differs* from the proposal(s) submitted in the previous round, the quantities l, k, s, v and w in Eq. 5.9 can no longer be computed directly from the market profile of the previous round. First, we must modify the profile to account for the difference between the estimator's current proposal and its proposal bag in the previous round. Thus, we arrive at the profile that will result from the current round if the estimator submits its new proposal as planned and the other agents re-submit their previous proposals.

If the estimator's (new) proposal bag contains other proposals next to $\{x \doteq y\}$, these other proposals constitute a-priori knowledge of ψ_i, the proposal profile of the current round. To account for such knowledge, modification of Eq. 5.9 may be required. Prop. 5.6 below describes the computations that must be performed to estimate the success probability of $\{x \doteq y\}$ in this case.

Proposition 5.6 (subjective proposal success probability)
Suppose that agent m, to be referred to as "the estimator", has submitted the proposal

bag β_{i-1} in round $i-1$, and deliberates the submission of the proposal bag β_i in round i. Formally,

$$\psi_{i-1}(m) = \beta_{i-1} \;\wedge\; \psi_i(m) = \beta_i \tag{5.13}$$

Suppose that agent m wants to estimate the success probability of $\{x \rightleftharpoons y\} \in \beta_i$, under the assumption that the other agents will re-submit their last proposals, that is,

$$(\forall j \in I)\; j \neq m \Rightarrow \psi_i(j) = \psi_{i-1}(j) \tag{5.14}$$

Let μ_{i-1} denote the market profile computed from ψ_{i-1}, and let μ_i denote the market profile associated with ψ_i defined by (5.13) and (5.14), that is, the profile that the estimator expects to result from the current round, if it submits β_i. Formally,

$$\begin{cases} \mu_i^- \triangleq (\mu_{i-1}^- \backslash \beta_{i-1}^-) \uplus \beta_i^- \\ \mu_i^+ \triangleq (\mu_{i+1}^+ \backslash \beta_{i+1}^+) \uplus \beta_i^+ \end{cases} \tag{5.15}$$

Define the proposal bags θ and ζ, the tool bag $\hat{\gamma}$, and the quantities s^*, w^*, v^*, s, w, v, l and k as below, where X denotes an arbitrary tool type in R.

$$\begin{aligned}
\theta &\triangleq \{\{X \rightleftharpoons y\} \in \beta_i\} \\
\zeta &\triangleq \beta_i \backslash \theta \\
\hat{\gamma} &\triangleq \{r \in \mu_i^- \mid r \neq y\} \backslash (\zeta^- \uplus \{x\}) \\
s^* &\triangleq |\theta| \\
w^* &\triangleq |\{\{X \rightleftharpoons y\} \in \theta \mid X \succ x\}| \\
v^* &\triangleq |\{\{X \rightleftharpoons y\} \in \theta \mid X \sim x\}| \\
s &\triangleq |\hat{\gamma}| - s^* \\
w &\triangleq |\{r \in \hat{\gamma} | \, r \succ x\}| - w^* \\
v &\triangleq |\{r \in \hat{\gamma} | \, r \sim x\}| - v^* \\
l &\triangleq \mu_i^-(y) - w^* \\
k &\triangleq \mu_i^+(y) - s^*
\end{aligned}$$

Then the subjective success probability of $\{x \rightleftharpoons y\}$ equals

$$\widehat{P}_y = \frac{\displaystyle\sum_{i=0}^{l-1} \binom{w}{i} \cdot \sum_{j=0}^{k-i} \binom{v}{j} \binom{s-w-v}{k-i-j} \cdot \min\left(1, \frac{l-i}{v^*+j+1}\right)}{\displaystyle\binom{s}{k}} \tag{5.16}$$

Proof.
We prove the proposition by comparing it with Prop. 5.5 on page 189, and showing that all a-priori knowledge of the estimator is indeed incorporated in (5.16).

The a-priori knowledge that the estimator has of ψ_i involves its own proposal bag β_i. The relevant information in this knowledge is expressed in the subbags θ and ζ of β_i. θ contains the proposals of which the estimator knows that they are competitors to $\{x \rightleftharpoons y\}$, and ζ contains the known noncompetitors.

The definition of the bag $\hat{\gamma}$ (of potential adversaries to x) differs from the corresponding definition in Prop. 5.5, in that the tools in the supply bag ζ^- of ζ are removed. These are the tools that the estimator uses to obtain other tools than y. Hence, the estimator knows that these tools are *not* potential adversaries to x.

The semantics of the symbols s^*, w^*, and v^* is analogous to that of s, w, and v in Prop. 5.5, except that they pertain exclusively to the *known* adversaries. Thus, w^* denotes the number of adversaries *known* to be more eligible than x, v^* the number of adversaries *known* to be of equal eligibility, and s^* denotes the total number of *known* adversaries.

Since s^*, w^*, and v^* are subtracted in the corresponding definitions of s, w, and v above, the latter symbols pertain exclusively to *unknown* adversaries. Similarly, l now denotes the number of available y-tools that is left, after any known competitors more eligible than $\{x \rightleftharpoons y\}$ have taken their share, and k denotes the actual number of *unknown* competitors.

Hence, next to the presence of v^* in the formula, (5.16) differs from (5.9) on page 190, in that it features combinatorialization over the number (i) of *unknown* competitors that turn out to be more eligible than $\{x \rightleftharpoons y\}$, and the number ($j$) of *unknown* competitors that happen to be of equal eligibility.

This explains the appearance of v^* in "$\min(1, \frac{l-i}{v^*+j+1})$": The latter expression represents the objective success probability of $\{x \rightleftharpoons y\}$ if the supply of y is $l + w^*$, and there are $i + w^*$ competitors more eligible than $\{x \rightleftharpoons y\}$, and $j + v^*$ competitors of equal eligibility (cf. Prop 5.4).

■

By providing formulas for subjective proposal execution probabilities, we have suggested that the estimation of success probabilities is a vital part of the agents' rationality in any IG mechanism. However, it may not always be *feasible* to investigate the strategy spaces of agents exhaustively to ensure that the expected utility of the selected strategy is maximal. Such exhaustive investigation is not a problem when solving reassignment problems, but strategy spaces can be quite large in general reallocation problems, where agents are endowed with multiple tools. To *ensure* that a bag of elementary proposals constitutes an optimal strategy, a separate calculation is required for each proposal bag with a higher utility than the agent's current endowment.

Moreover, the exact computation of the subjective success probability of a single proposal *bag* is much more cumbersome than that of a single proposal. Prop. 5.6 specifies a formula for the success probability of a single proposal in a bag, not for the probability that *all* of the proposals in the bag are accepted. If the success probabilities of different proposals in a proposal bag were independent, we could simply multiply the single-proposal probabilities to arrive at the success probability of the entire bag. Unfortunately, single-proposal probabilities are almost always *dependent*. Consequently, to compute the success probability of a proposal bag, we would have to perform a calculation similar to that of Eq. 5.16, except that we must now combinatorialize over *tuples* of competitors for the different proposals in the bag. This turns the computation of expected utilities of proposals into a very cumbersome task.

Hence, to apply IG to the (typically full-fledged) reallocation problems associated with real-life optimization problems (cf. Sect. 2.1.3), bounded rationality is called for. How the agents' rationality should be bounded depends on the underlying optimization problem. We pay more attention to this issue in Sect. 5.9.1. For now, we turn to a subclass of reallocation problems that poses none of the above problems: the class of reassignment problems.

5.6.5 An IG mechanism for tool reassignment

In a reassignment problem, each agent is endowed with one tool, which it would like to exchange against a tool of some other type. Because the endowment of an agent is a tool, rather than a bag of tools, we denote endowments by Latin, rather than Greek letters.[13] An IG reassignment mechanism features the same elementary exchange proposals as a reallocation mechanism. However, the agent now only submits one proposal $\{x \rightleftharpoons y\}$ in each round, and it cannot vary x. In view of this, it is more economical to let the auctioneer keep track of the tool types offered by agents in their first proposal, and abbreviate subsequent proposals $\{x \rightleftharpoons y\}$ as 'y'. If an agent endowed with a type-a tool submits the proposal 'a', we will interpret this as an empty proposal (signifying "I pass") instead of an offer to exchange a type-a tool for another type-a tool. The latter, literal interpretation would make little sense, since the utility of a tool is determined completely by its type. Furthermore, if type-a tools are undersupplied, submission of $\{a \rightleftharpoons a\}$ would amount to the altruistic offer "Please, take my a and give me back any junk that you would like to get rid of". For clarity, we *denote* the empty proposal by '$-$' in the text that follows. Note that '$-$' is not a *definitive* pass bid. An agent that passes in the current round may take part again in the bidding of the next round.

[13] Greek letters are reserved for bags and bag constructs (cf. Sect. 2.2.2).

Because a proposal is now identified by a tool type instead of a bag of elementary gbags, a proposal profile is no longer a bag construct, but simply a function that maps each agent to a tool type. We therefore represent proposal profiles in reassignment mechanisms by the Latin letter p instead of the Greek letter ψ.

For the formal definition of an IG mechanism as a tuple $\langle \mathfrak{R}, \mathfrak{E}, \mathfrak{T} \rangle$, the above implies that the agent rationality specification $\mathfrak{R}$ is of the form

$$\mathfrak{R}(e, u, \mu_{i-1}, b_{i-1}, i) \in B_i \quad \text{where } B_i \subseteq R$$

instead of (see page 182)

$$\mathfrak{R}(\varepsilon, u, \mu_{i-1}, \beta_{i-1}, i) \in \Xi_i \quad \text{where } \Xi_i \subseteq \mathfrak{B}(\mathcal{G}_1(R))$$

The particular IG reassignment mechanism that we propose has the following features.

$\mathfrak{R}$: Agents are risk-neutral expected-utility maximizers.

$\mathfrak{E}$: Tool eligibility is equated with scarcity.

$\mathfrak{T}$: A relaxation process terminates when all agents re-submit their last proposals.

Attributing the same semantics to σ and $\widehat{P}_y$ as in Prop. 5.6, this can be expressed formally in terms of $\langle \mathfrak{R}, \mathfrak{E}, \mathfrak{T} \rangle$ as

$$\mathfrak{R}((e, u, \mu_{i-1}, b_{i-1}, i) \in \{b \in R \mid (\forall r \in R)\ \hat{u}(b) \geq \hat{u}(r)\} \tag{5.17}$$

$$\begin{aligned} \text{where} \quad & \hat{u}(e) \triangleq u(e) \\ & \hat{u}(y) \triangleq \widehat{P}_y \cdot u(y) + (1 - \widehat{P}_y) \sum_{z \in \hat{\sigma}_i} \tfrac{u(z)}{|\hat{\sigma}_i|} \qquad \text{if } y \neq e \\ & \hat{\sigma}_i \triangleq \sigma_{i-1} \uplus \{y\} \setminus \{b_{i-1}\} \end{aligned}$$

$$\mathfrak{E}(\sigma) \triangleq \succsim \quad \text{where } (\forall r_1, r_2 \in R)\ r_1 \succ r_2 \Leftrightarrow \sigma(r_1) > \sigma(r_2) \tag{5.18}$$

$$\mathfrak{T}(p_{i-1}, p_i, i) \triangleq p_i = p_{i-1} \tag{5.19}$$

To see what the behavior of agents is like in such a reassignment mechanism, we recall a reassignment example (Example 3.18 on page 77) which we used to illustrate the computation of a Walrasian allocation via the Top-Trading-Cycles algorithm. Table 5.5 recapitulates the relative agent-level utilities of this example, and Table 5.6 shows the profile p_0 of initial proposals and the initial market profile μ_1 computed from p_0.

From the market profile, agents 1 and 5 conclude that their most preferred tool is undersupplied. Agent 1 has two viable options for its proposal in the next round. It can propose '$-$', satisfying itself with the little utility its currently allocated

U	1	2	3	4	5	6
a	10%	.	.	.	.	100%
b	100%	10%	.	.	100%	.
c	.	80%	10%	.	.	.
d	.	.	100%	10%	.	.
e	.	100%	.	100%	10%	.
f	.	.	.	.	90%	10%

Table 5.5: Agent-level utilities (dots denote negligible utilities).

agent	1	2	3	4	5	6
endowment	a	b	c	d	e	f
proposal	b	e	d	e	b	a

(a) initial proposal profile

tool	a	b	c	d	e	f
demand	1	2	0	1	2	0
supply	1	1	1	1	1	1
scarcity	0	1	-1	0	1	-1

(b) market profile and scarcities

Table 5.6: Proposal and market profiles after the initial bidding round.

tool provides, or repeat its bid on b. Due to the considerable increment of utility associated with acquiring b, a risk-neutral agent will opt for the latter. From the market profile, it can deduce that its single competitor can possess any one tool of type c, d, e or f with equal probability. In two of these four cases, agent 1 is more eligible. In one case, the eligibilities are equal, rendering a toss. Thus, the subjective success probability $\widehat{P}_b$ of acquiring b equals $\frac{2+0.5}{4} = \frac{5}{8}$, rendering an estimated utility $\hat{u}(b)$ of at least[14] $\frac{5}{8} \cdot 100\% = 62.5\%$. This exceeds $\hat{u}(-) = 1 \cdot 10\% = 10\%$. The same line of reasoning leads to agent 4 proposing e. Agent 5 must choose between b or f. In view of the small difference between the utilities of b and f, a risk-neutral agent would generally be inclined to opt for f. In this particular case, however, agent 5 can deduce from the market profile that the only other agent that is equally eligible as itself possesses b, and can therefore not be a competitor for b. In other words, a bid on b is *certain* to succeed. A similar conclusion can be drawn by agent 2 with respect to a bid on e. Hence, each of the agents 1, 2, 4 and 5 sticks to its original bid. Since the others have no reason to change theirs, the same proposal profile occurs again, and the relaxation is terminated.

In the pseudo-composition that follows, agents 2, 3, 5 and 6 will acquire their most preferred tool, and agents 1 and 4 will receive one of the oversupplied tool types c and f. Hence, the outcome will be one of the allocations [*cedfba*] or [*fedcba*], with a community utility of approximately $\frac{400}{6} \approx 67\%$. In terms of community utility, IG performs significantly better than a Walrasian auction could. As we derived in

[14]We neglect the second term in the definition of $\hat{u}(y)$ in Eq. 5.17.

Sect. 3.3.8 (at page 78), the only Walrasian allocation is [*aecdbf*], with a community utility of approximately $\frac{240}{6} = 40\%$.

In this particular example, the *commitment* required by IG from agents bidding on a tool is instrumental to its outperforming a Walrasian auction. In other cases, however, the *uncertainty* about the endowment and preferences of the other agents is more important. Situations such as the one described above, in which an agent can be *certain* of success when bidding on a scarce tool are relatively rare. In many cases, such uncertainty is capable of preventing a *small group* of agents (in the above example: agents 2 and 5) from frustrating community interest.

Suppose, for example, that we face a problem similar to the one described above, but with two additional agents possessing f, that have a preference for tool type d. The corresponding market profile after the first round differs from that of Table 5.6b in that the supply of f and the demand of d now equal 3. This implies that d is now the scarcest tool. As a consequence, agents 2 and 5, no longer certain of success, will opt for their alternatives c and f, rendering the *optimal* allocation [*bcdefaff*] with a community utility of 73%.

In contrast, application of the Top-Trading-Cycles algorithm to this problem instance would render the (unique) Walrasian allocation [*aecdbfff*] with a community utility of only 32%.

5.7 Termination of the Relaxation Process in IG

In Def. 5.2 on page 182, we mentioned three reasons to terminate the relaxation process.

1. The sequence of proposal profiles reaches a stationary point.
2. The current profile features zero excess demand.
3. The number of rounds reaches a previously imposed deadline.

The presence of the third condition suggests that the other two, in isolation or in combination, are insufficient to ensure timely termination of the process. This is indeed so. The termination of the relaxation process is a problematic issue.

Initially, we envisioned stationarity (i.e., the event that the proposal profile ψ_i equals the profile ψ_{i-1} of the previous round) as the sole condition for termination, because it amounts to attaining the goal of relaxation: to arrive at a state where each of the agents willingly commits to its proposal and the risk it may encompass. In any case, it is pointless to continue if the collective agent response to the associated market profile μ_i happens to be deterministic: in this case, the stationary state will persist.

Hence, the termination criterion should at least include this stationarity condition as a sufficient condition for termination.

The validity of the second condition, market clearance, is less obvious. In a Walrasian auction, market clearance is always a (persistent) stationary state, but this is not the case in IG. Upon reaching a state of zero excess demand, an IG agent may well change its proposal if given the opportunity to do so. Hence, the incorporation of market-clearance as a sufficient condition for termination incurs a violation of the design principle of agent autonomy: Agents may be forced to accept a commitment that they did not willingly engage in.

In CMAT, where the proper attitude in design is to be pragmatic rather than principled, the market-clearance condition may still be viable, if it incurs a considerable improvement of solution quality. There is at least some reason to expect that this is the case: Pseudo-composition on a proposal profile with zero excess demand amounts to executing *all* of the agents' proposals. This will often render a relatively high average agent satisfaction. Thus, while the market clearance condition can lead to a violation of agent autonomy, it may enhance mechanism performance. To weigh these two conflicting design goals, we have performed experiments, whose outcomes will be discussed in Chapter 7. For now, we assume that an IG relaxation process can terminate for two reasons only: due to reaching a stationary state or due to reaching the deadline.

The deadline condition is, of course, the least attractive of all. Like the market-clearance condition it violates the agents' autonomy. Moreover, it is more likely to have an adverse effect on mechanism performance than to enhance it. Indeed, our only motivation to incorporate the deadline condition is to have an emergency measure in case things go wrong. That things can go wrong, in the sense that the relaxation process may never reach a stationary state, is demonstrated in the following example.

Example 5.7 (Non-termination due to cyclic behavior)
Consider the reassignment problem described by the utility matrix in Table 5.7. The problem comprises 7 agents and 4 tool types. The initial endowments of the agents are underlined in the matrix. Thus, initially, agents 1 and 2 possess a type-a tool, agents 3 and 4 a type-b tool, etc. We investigate the relaxation process that unfolds if the example problem is fed to the IG reassignment mechanism defined in Sect. 5.6.5.

The course that a relaxation process takes is described by a sequence of proposal profiles ψ_t. *We denote a proposal profile by a string of characters, where the i-th character in the string indicates the tool type requested in the proposal of agent i. To elucidate our argumentation, we also present the associated market profiles* (μ_t) *and scarcity profiles* (σ_t). *The first round, in which the agents communicate their*

u	*1*	*2*	*3*	*4*	*5*	*6*	*7*
a	$\underline{80}$	$\underline{80}$	*100*	*0*	*0*	*0*	*0*
b	*100*	*100*	$\underline{10}$	$\underline{0}$	*0*	*100*	*0*
c	*0*	*0*	*0*	*0*	*100*	$\underline{0}$	*100*
d	*0*	*0*	*0*	*100*	$\underline{0}$	*0*	$\underline{0}$

Table 5.7: A termination problem.

first preferences to the auctioneer, leads to the following state.

$$\psi_1 = \text{`}bbadcbc\text{'},\ \ \mu_1 = \langle\{aabbcdd\},\ \{abbbccd\}\rangle,\ \ \sigma_1 = \{ad \rightleftharpoons bc\}$$

In view of their tool utilities (see Table 5.7), agents 4, 5, 6, and 7 will stick to their initial proposals no matter what the current market profile is like. Consequently, we need only consider the reasoning employed by agents 1, 2 and 3.

Agent 1 concludes from σ_1 that b is scarce. The subjective success probability $\widehat{P}_b$ in the context of μ_1 can be computed from Eq. 5.9 on page 190. It equals $0.58\not{3}$.[15]*. This leads to an expected utility for $\{a \rightleftharpoons b\}$ of*

$$\hat{u}(b) = \widehat{P}_b \cdot u_1(b) + (1 - \widehat{P}_b) \cdot \frac{u_1(a) + u_1(d)}{2} = 0.58\not{3} \cdot 100 + 0.41\not{6} \cdot 40 = 75\%$$

Since this is less than the utility of its current endowment (80%), agent 1 will submit '−' in the next round. Agent 2, who faces exactly the same situation, will do the same.

Agent 3 will stick to its previous proposal $\{b \rightleftharpoons a\}$, since this proposal is guaranteed to succeed under σ_1. Hence, the situation after the second round of bidding is

$$\psi_2 = \text{`}--adcbc\text{'},\ \ \mu_2 = \langle\{bbcdd\}, \{abccd\}\rangle,\ \ \sigma_2 = \{bd \rightleftharpoons ac\}$$

In this situation, agents 1 and 2 are — or rather, feel — certain of the success of $\{a \rightleftharpoons b\}$, and will bid accordingly. In contrast, agent 3 is no longer certain of the success of $\{b \rightleftharpoons a\}$. In fact, this proposal is certain to fail under μ_2, which features zero supply of a. Hence, the expected utility of $\{b \rightleftharpoons a\}$ equals $\hat{u}(a) = 0\cdot 100 + 1 \cdot \frac{10+0}{2} =$ 5%, less than the utility of its current endowment. So agent 3 will submit '−'. This leads to

$$\psi_3 = \text{`}aa-dcbc\text{'},\ \ \mu_3 = \langle\{aabcdd\}, \{bbbccd\},\ \ \sigma_3 = \{aa \rightleftharpoons bb\}$$

In this situation, a proposal $\{a \rightleftharpoons b\}$ by agent 1 or 2 is doomed to fail again, while $\{b \rightleftharpoons a\}$ by agent 3 is certain of success. Hence, the situation after this round is

$$\psi_4 = \text{`}--adcbc\text{'},\ \ \mu_4 = \langle\{bbcdd\}, \{abccd\}\rangle,\ \ \sigma_4 = \{bd \rightleftharpoons ac\}$$

[15] $0.58\not{3}$ is a shorthand for the number 0.5833333..., that is, $0.58 + \frac{1}{3}$; likewise $0.4\not{2}1\not{8}$ denotes the number 0.4218218218....

We observe that $\psi_4 = \psi_2$. The relaxation process has entered a cycle, and will go on indefinitely, if left to its own dynamics. △

The relaxation process in the above example never reaches a stationary state, even though such states do exist. If the process would reach the state

$$\psi^* = \text{`}-abdcbc\text{'},\ \ \mu^* = \langle\{abbcdd\},\{abbccd\}\rangle,\ \ \sigma^* = \{d \rightleftharpoons c\}$$

then agents 2 and 3 would stick to their proposals, because the zero scarcities for a and b guarantee the success of their proposals. However, agent 1 would also stick to '−'. Its subjective success probability for $\{a \rightleftharpoons b\}$ under μ^* equals 0.583, leading to $\hat{u}(b) = 75\%$ (as we saw earlier). This is lower than $\hat{u}(-) = 80\%$.

That the above stationary state is never reached is due to lack of coordination between agents 1 and 2, which, in turn, is a consequence of the fact that they respond to the last market profile in parallel. If the agents would submit their proposals sequentially, on the basis of a market profile that incorporates the latest submitted proposal, the stationary state would be reached. Of course, sequential bidding is not really attractive for a distributed mechanism. We would prefer to retain at least some parallelism and still prevent the overcompensation by agents from stalling the termination of the relaxation process indefinitely. This can be obtained if we employ *asynchronous* parallelism.

Termination under asynchronous parallelism

Asynchronously parallel proposal updating entails that the agents submit proposals in a random and varying order, and the auctioneer computes a new market profile as soon as it has received the responses of a certain percentage (say, $k\%$) of the agent population on the previous profile. The agents must communicate a round number together with their proposal, to indicate on which market profile their proposal is based. This enables the auctioneer to discard proposals that are based on outdated market profiles. In turn, the communication from the auctioneer to an agent involves, next to a market profile, a flag indicating whether an update of the agent's proposal has been incorporated in the profile.

This procedure can be applied in all rounds, with one exception. If the auctioneer observes that the new proposal profile[16] equals the previous one, it waits until *all* agents have submitted their responses, and uses the complete profile to decide whether the relaxation process can be terminated. If it would not do this, agent autonomy could be violated by binding agents to commitments that they did not willingly engage in.

[16]The new profile comprises the new proposals of the $k\%$ agents that have managed to submit these, and the last submitted proposals of the remaining agents

In the above example with 7 agents, asynchronously parallel updating guarantees that the relaxation process reaches a stationary state within a finite number of rounds with probability 1, provided that the parallelism is sufficiently asynchronous (which is the case if $k/100 \leq 6/7$). The question is, of course, whether such a sufficient condition for termination exists for *every* assignment problem.

Unfortunately, this is not the case, as the example below shows. The example is derived from a randomly generated problem, which I ran a number of times on the Informed-Gambling Reassignment Testbed. In each trial run, the relaxation terminated by reaching the deadline (which was set to 2000 rounds). The trials involved asynchronous bidding using various values of k, as well as sequential bidding in a randomly varying order. This prompted for a closer examination, which is recorded below.

Example 5.8 (Cyclic behavior with asynchronous parallelism)
Consider the tool reassignment problem defined by the utility matrix in Table 5.8. The initial endowments of the agents are underlined in the matrix. We employ the same notational conventions for ψ, μ, and σ as in Example 5.7. We do not spell out the reasoning processes of all agents, but confine ourselves to those agents that change their proposals at some point in the relaxation process. In this particular example, this pertains only to agents 1 and 2. It turns out that all others stick to their initial proposals in each of the four states reached by the relaxation process. Also, we describe the reasoning by agents 1 and 2 in terms of their expected utilities for proposals, rather than the underlying success probabilities.

u	1	2	3	4	5	6	7	8	9	10
a	0	0	20	0	100	10	30	0	60	100
b	10	50	40	0	90	30	40	100	20	0
c	50	100	0	0	70	100	0	0	0	0
d	30	60	60	30	0	0	0	60	100	0
e	0	0	0	40	0	0	30	60	0	20
f	100	90	60	100	70	50	100	70	50	40
g	80	0	0	30	0	0	50	60	0	90
h	0	30	100	30	90	20	0	0	90	60

Table 5.8: A troublesome termination problem.

The initial state of the relaxation process is

$$\psi_1 = \text{'}fchfacfbda\text{'},\quad \mu_1 = \langle\{aabbcdefgh\},\{aabccdfffh\}\rangle,\quad \sigma_1 = \{beg \rightleftharpoons cff\}$$

In this situation, agent 2 is the only agent that is motivated to deviate from its initial proposal. The new state that arises from this update prompts agent 1 to deviate, *which prompts agent 2 to change back to its initial proposal, etc. This results in a relaxation process of the form $\psi_1, \psi_2, \psi_3, \psi_4, \psi_1, \ldots$ The reasoning processes of agents 1 and 2 in the four different states that occur in the relaxation process are described in Table 5.9, in terms of the expected utilities which the agents compute for their options. Here, an option is any tool type with a greater utility than the agent's current endowment. Inspection of Table 5.8 tells us that agents 1 has only two options (f and g), whereas agent 2 has five (b, c, d, f, and h).*

$\hat{u}$	*options for agent 1:*		*options for agent 2:*				
state:	*f*	*g*	*b*	*c*	*d*	*f*	*h*
$\psi_1 =$ '*fc...*'	*10.0*	*8.0*	*5.0*	*3.2*	*2.5*	*1.7*	*1.9*
$\psi_2 =$ '*fb...*'	*6.6*	*8.0*	*5.0*	*3.2*	*2.5*	*1.7*	*1.9*
$\psi_3 =$ '*gb...*'	*6.6*	*8.0*	*5.0*	*7.2*	*4.7*	*5.7*	*2.8*
$\psi_4 =$ '*gc...*'	*10.0*	*8.0*	*5.0*	*7.2*	*4.7*	*5.7*	*2.8*

Table 5.9: Expected utilities of agent options in 4 states.

From Table 5.9, we conclude that, in the initial state ψ_1, agent 1 will stick to 'f', while agent 2 changes to 'b'. Hence, the new state is ψ_2, in which agent 1 changes to 'g', and agent 2 sticks to 'b'. The process reaches state ψ_3, in which agent 1 sticks to 'g', and agent 2 changes to 'c'. This brings about state ψ_4, which prompts agent 1 to change to 'f', while agent 2 sticks to 'c'. The result is that we are back in state ψ_1 again. △

In the context of asynchronously parallel updating, the relaxation process is a Markovian process, which generally takes different paths through the state space in different trial runs. In the case of this particular reassignment problem however, the relaxation process is fully deterministic, due to the fact that, in each state that can be reached from the initial one, only one agent is motivated to deviate from its previous proposal. This explains why asynchrony is, in this case, not able to "break the cycle", as it did in Example 5.7. No matter how we choose the asynchrony parameter k, either the single agent that wants to deviate is among the first $k\%$ that manage to submit a proposal, or it is not, and — so as not to violate agent autonomy — the auctioneer waits for it to submit its proposal.

Hence, the relaxation process will never reach a stationary state, no matter whether the mechanism features synchronously parallel, asynchronously parallel or sequential proposal updating. Note that this still does not mean that there is no stationary state, only that the process will never reach one if the initial state is ψ_1. So perhaps it should start somewhere else. But since we have no clue as to what would be "good initial states", this does not help us much. Restarting the relaxation process from a different initial state does not seem a very attractive idea, since we cannot preclude that this amounts to investigating the entire state space.

The inevitable conclusion is that, even with asynchronously parallel updating, we still need to incorporate the deadline condition in the termination criterion to accommodate the cases in which the relaxation process ends up in a persistent cycle. However, as remarked earlier, incorporating a deadline is a rather unsatisfactory measure. If the deadline is reached too frequently, we may just as well refrain from relaxation altogether. An alternative way to guarantee termination within an acceptable number of rounds would be welcome.

The alternative that we have thought of for IG involves a modification of the agents' rationality (i.e., of the response correspondence $\mathfrak{R}$). We presume that agents will grow weary of prolonged relaxation. This *negotiation weariness* characteristic is implemented in $\mathfrak{R}$ as follows.

Instead of simply bidding on a tool type y of maximal expected utility $\hat{u}(y)$, the agent compares $\hat{u}(y)$ with the expected utility of its previous bid x (under the current market profile), and bids on y only if the difference $\hat{u}(y) - \hat{u}(x)$ exceeds a certain treshold level L. Otherwise, it repeats its bid on x. The treshold value L starts at zero, and increases linearly with the round number i, according to

$$L = \frac{i}{D} \tag{5.20}$$

Here, D is the value of the deadline parameter of the mechanism. Since the range of $\hat{u}(\cdot)$ is the interval $[0, 1]$, the incorporation of weariness guarantees that the relaxation process becomes stationary in or before round D. In practice, however, the number of relaxation rounds rarely exceeds $D/4$.

Although the incorporation of negotiation weariness is a less extreme measure than the incorporation of a deadline, it can theoretically lead to severely suboptimal responses. If this happens frequently, negotiation weariness would constitute a rather irrational aspect of the agents' behavior. Furthermore, it can cause a significant degradation of mechanism performance. Hence, it is important to know how often weariness plays a role in the relaxation process, and how large the concessions are that it brings about. In Chapter 7, we will discuss our observations in this respect.

5.8 Characterization of IG

We characterize IG in two respects. First, we compare its key elements, the stationary states of its adjustment process, and the notion of rationality employed in its agent response correspondence to common conceptions of equilibria and rationality in game theory and AI. Second, we investigate whether the performance correspondence of the specific IG mechanism proposed for tool reassignment has any game-theoretic properties (such as Pareto optimality).

5.8.1 Characterization of IG equilibria

Equilibria and rationality in game theory

In game theory, equilibrium concepts and rationality notions are intimately connected. In fact, solution concepts like the well-known Nash equilibrium notion *define* a conception of rationality. In the following, we discuss three equilibrium notions in competitive game theory in this respect: pure Nash equilibria, mixed-strategy Nash equilibria, and correlated equilibria. The latter equilibrium notion is the game-theoretic concept that is most akin to equilibria attained in IG mechanisms. Our discussion of the other two notions merely serves as a leg up for the definition of correlated equilibria. Hence, readers already familiar with the notion of Nash equilibrium and its purpose as a characterization of rational agent behavior *should feel free to jump to our description of correlated equilibria on page 209.*

In game theory, the relative desirability of possible outcomes is expressed in terms of (cardinal or ordinal) utilities, called payoffs. In classical, competitive game theory (Neumann & Morgenstern, 1947), a player in a game is supposed to pursue only one goal: to maximize its payoff. A *game in normal form* (Wang & Parlar, 1989) is an abstraction of real-life multi-agent decision problems that comprises the following entities:

- a finite set of players, identified by the natural numbers $1, \ldots, n$;
- a finite[17] set S_i of strategies for each player i;
- a cardinal payoff function u_i for each player, which associates a real-valued payoff with each strategy tuple $\langle s_1, \ldots, s_n \rangle$, where $s_i \in S_i$ denotes the strategy chosen by agent i.

The usage of the terms player and payoff is largely confined to game theory. In the sequel, we therefore speak of *agents* instead of players, and *utilities* instead of payoffs. We will, however, use the game-theoretic term '*strategy*', instead of 'action' or 'decision', because this term has become common in MAT, as well as in economics, social choice theory, and game theory.

The idea of a game in normal form is that the agent's decision problem is to select a strategy that will lead to an outcome of maximal utility. Here, the outcome of a game is the joint strategy[18] itself, not a mapping thereof, as in our model of

[17] In game theory, the strategy space can also be infinite, but we do not discuss such games.

[18] The joint strategy is the tuple of strategies selected by the agents.

iterative mechanisms. One distinguishes between games with complete and games with incomplete information. In the former, the agents all have *perfect knowledge*, that is, each agent knows the strategy spaces and the outcome utilities of the other agents. An agent with perfect knowledge can determine, for each of the possible joint moves of its opponents, which of its strategies would be best. It can also form hypotheses about the deliberations of other agents in this respect. The combination of these two kinds of inferences comprises a straightforward strategy-selection procedure only if there happens to be a strategy in the agent's strategy space that is best in all cases, that is, no matter which strategies the other agents choose. Such a *dominant strategy* is not likely to exist very frequently. In the complementary case, the agent may get entangled in reciprocal reasoning of the form

> She 'll probably (...), so I'll (...).
>
> But then, she may *expect* that I 'll (...), because I expect her to (...), and will therefore (...) instead of (...), so I 'll (...).
>
> But then, if she *expects* that I expect that she expects that I 'll (...), she will (...) instead, so perhaps I 'd better (...).
>
> ⋮

It is clear that reciprocal reasoning poses a serious obstacle to the definition of (multi-agent) rational behavior: The outcome generally depends on the depth to which the recursion is performed, and there is no *general* a-priori reason to opt for any specific depth.

The Nash equilibrium notion provides a way out of this dilemma by looking at the stationary states of the reciprocal-reasoning process. As such, it focuses on likely outcomes, rather than on how and when these might be arrived at. In words, a Nash equilibrium is a strategy tuple s^* such that no agent i can profit from unilateral deviation from s_i^*. A formal definition is given below.

Definition 5.9 (Nash equilibrium)
Let $\langle S, u\rangle$ denote a game in normal form, where $S = \langle S_1, \ldots, S_n\rangle$ and $u = \langle u_1, \ldots, u_n\rangle$, with S_i and u_i defined as above. Let $s^*|_{s_i^* \leftarrow s}$ denote the substitution of s for s_i^* in $s^* = \langle s_1^*, \ldots, s_n^*\rangle$, that is, $\langle s_1^*, \ldots, s_{i-1}^*, s, s_{i+1}^*, \ldots, s_n^*\rangle$. Then the outcome $s^* = \langle s_1^*, \ldots, s_n^*\rangle$ is a (pure-strategy) Nash equilibrium for $\langle S, u\rangle$ iff

$$(\forall i \in \{1, \ldots, n\})\ (\forall s \in S_i)\ u_i(s^*) \geq u_i(s^*|_{s_i^* \leftarrow s})$$

△

The standard context of the Nash equilibrium notion is that of agents with perfect knowledge, who select a strategy *simultaneously*. Its usual interpretation as a definition of agent rationality (Damme, 1991) entails that the pure strategies s_i^* do not merely specify the choices which the agents (intend to) make, but also the choices that each agent expects or assumes the other agents to make. As such, a Nash equilibrium can be viewed as a tuple of consistent beliefs and intentions with respect to agent behavior that constitutes a stationary state in a reciprocal-reasoning process. Here, consistency entails that the strategy s_i^*, which all agents other than i expect agent i to choose, is optimal for agent i, if every other agent j picks the strategy s_j^* which agent i expects it to pick.

Example 5.10 (Nash Equilibrium)
Consider the 2-agent game depicted in Table 5.10. The elements of row x, column y in the table denote the utilities of the outcome $\langle x, y \rangle$ for the two agents, where the number in the lower left corner denotes the utility for agent 1, and the number in the upper right corner denotes that of agent 2. Thus, according to the table, the utility of the outcome $\langle s_1, s_2 \rangle$ is 3 for agent 1, and 0 for agent 2.

agent 1's strategy:	*agent 2's strategy:* s_1	s_2
s_1	2 2	0 3
s_2	1 0	3 1

Table 5.10: A 2-agent game with Nash equilibrium $\langle s_1, s_2 \rangle$. Numbers in upper-right corners represent outcome utilities of agent 2; those in lower-left corners pertain to agent 1.

The joint strategy $\langle s_1, s_1 \rangle$ is a Nash equilibrium, since unilateral deviation by agent 1 would lead to $\langle s_2, s_1 \rangle$ with a utility of 0 instead of 2 (for agent 1), while unilateral deviation by agent 2 would lead to $\langle s_1, s_2 \rangle$ with a utility of 0 instead of 2 (for agent 2). In this case, the equilibrium is unique. The outcome $\langle s_2, s_2 \rangle$ is not a Nash equilibrium, since unilateral deviation would be profitable for agent 1. △

A Nash equilibrium does not always exist, and if it exists, it need not be unique. In Table 5.11a, there are two Nash equilibria ($\langle s_1, s_2 \rangle$ and $\langle s_2, s_1 \rangle$). In Table 5.11b, there is no Nash equilibrium. The scissors-paper-stone game is another example of a game without Nash equilibria.[19]

[19] This game involves two players, who simultaneously choose one of the strategies scissors, paper, or stone. The winner is determined by the rules: scissors cut paper, paper wraps stone, and stone smashes scissors.

	s_1	s_2
s_1	0 / 0	2 / 3
s_2	4 / 2	1 / 1

(a)

	s_1	s_2
s_1	0 / 2	1 / 0
s_2	1 / 0	0 / 1

(b)

Table 5.11: Games may have multiple (a) or no (b) Nash equilibria.

A common basic demand for any notion of rationality is that there should be at least one rational approach to any problem, and preferable only one. Since game-theoretic equilibrium concepts are supposed to constitute definitions of rational behavior, it is unacceptable for equilibria not to exist in certain games. It is also undesirable to have many equilibria in a single game. The agents in the game of Table 5.11a, for example, still face a reciprocal-reasoning dilemma. The presence of *two* Nash equilibria implies that they must choose between the following two alternative hypotheses.

1. My opponent will aim for the equilibrium with the highest payoff (i.e. $\langle s_2, s_1\rangle$ for agent 2, and $\langle s_1, s_2\rangle$ for agent 1).
2. My opponent will expect *me* to aim for the equilibrium with the highest payoff, and will select its strategy accordingly.

In the absence of any prearranged conventions, or some other means of coordination, neither of the above hypotheses is more attractive than the other. As a consequence, all of the four possible outcomes are equally likely, and the probability that the outcome is one of the two Nash equilibria is only 50%.

The strife for existence and uniqueness of equilibria led to the conception of several refinements of the above notion of Nash equilibrium. We discuss two of these concepts. The first one, the concept of mixed-strategy Nash equilibria, is based on the notion of randomized strategy.

Strategy randomization entails that agents define a probability distribution on their strategy space, and select a strategy randomly, according to this distribution. The original (nonrandomized) strategies are referred to as pure strategies; the term mixed strategy is used to denote a randomized strategy, including the extreme cases in which a specific pure strategy is selected with probability 1. The term mixed-strategy *equilibrium* is commonly reserved for the case that none of the probabilities p_i equal 1. Otherwise, one speaks of a pure-strategy equilibrium.

A mixed strategy for an agent with a (pure-)strategy space $s_1, \ldots, s_m$ can be represented as an m-tuple of probabilities ($\langle p_1, \ldots, p_m\rangle$), where p_i denotes the probability that the application of the mixed strategy leads to selection of strategy s_i. Obviously, $\sum_{i=1}^{m} p_i = 1$, so a mixed strategy in a game with a strategy space of only two (pure) strategies for each agent (as in the games of Table 5.11) can be represented by the single probability p that the agent will select s_1. If the game involves two agents — as in Table 5.11 — a joint mixed strategy can then be represented as $\langle p_1, p_2\rangle$, where p_i denotes the probability that agent i will select s_1.

The joint mixed strategy $\langle \frac{1}{2}, \frac{1}{2}\rangle$ (i.e., a coin-toss by both agents) is not a Nash equilibrium in Table 5.11a, because $\langle \frac{1}{2}, 1\rangle$ has a higher expected utility for agent 2. To see whether a mixed-strategy Nash equilibrium exists for the game in Table 5.11a, we write down the expected utilities $\hat{u}_i$ of the joint mixed strategy $\langle p_1, p_2\rangle$ as a function of p_1 and p_2.

$$\begin{aligned} \hat{u}_1(p_1,p_2) &= 3p_1 \cdot (1-p_2) + 2\cdot(1-p_1\cdot)p_2 + (1-p_1)\cdot(1-p_2) = \\ &= 2p_1 - 4p_1p_2 + p_2 + 1 \end{aligned} \quad (5.21)$$

$$\begin{aligned} \hat{u}_2(p_1,p_2) &= 2p_1 \cdot (1-p_2) + 4\cdot(1-p_1\cdot)p_2 + (1-p_1)\cdot(1-p_2) = \\ &= p_1 - 5p_1p_2 + 3p_2 + 1 \end{aligned} \quad (5.22)$$

To determine for which mixed strategy p_i the expected utility $\hat{u}_i$ for agent i is maximal, we differentiate $\hat{u}_i$ with respect to p_i

$$\frac{d\hat{u}_1}{dp_1}(p_1,p_2) = 2 - 4p_2 \quad (5.23)$$

$$\frac{d\hat{u}_2}{dp_2}(p_1,p_2) = 3 - 5p_2 \quad (5.24)$$

From (5.23) and (5.21), it follows that $\hat{u}_1$ has a local maximum of $\frac{3}{2}$, independent from p_1, if $p_2 = \frac{1}{2}$. Likewise, we conclude from (5.24) and (5.22) that $\hat{u}_2$ has a local maximum of $\frac{8}{5}$, independent from p_2, if $p_1 = \frac{3}{5}$. From the independence of $\hat{u}_1(p_1, \frac{1}{2})$ from p_1, and that of $\hat{u}_2(\frac{3}{5}, p_2)$ from p_2, it follows that $\langle \frac{3}{5}, \frac{1}{2}\rangle$ is a mixed-strategy Nash equilibrium in Table 5.11a.

Because there are only two (pure) strategies in each strategy space in this case, any border extreme of $\hat{u}_i(p_1, p_2)$ is a pure strategy. Hence, $\langle \frac{3}{5}, \frac{1}{2}\rangle$ is the only mixed-strategy Nash equilibrium.

A similar computation leads to the conclusion that the joint mixed strategy $\langle \frac{1}{2}, \frac{1}{3}\rangle$ is the unique Nash equilibrium in the game of Table 5.11b, with expected utilities of $\frac{3}{2}$ and $\frac{1}{2}$ for agents 1 and 2 respectively.

The notion of mixed-strategy Nash equilibrium has two major advantages over the pure-strategy version. First, a mixed-strategy equilibrium always exists. Second, it can be arrived at without any communication between the agents. There is also a disadvantage, however. If we compare the mixed-strategy equilibrium with the two pure strategy equilibria $\langle s_1, s_2 \rangle$ and $\langle s_2, s_1 \rangle$, we observe that the total expected utility $(\hat{u}_1 + \hat{u}_2)$ of the mixed-strategy equilibrium is significantly below that of the pure-strategy equilibria. In this respect, the mixed-strategy Nash equilibrium is unsatisfactory from a utilitarian point of view. However, a more profitable solution concept is not available, as long as the agents are unable to coordinate their behavior.

If coordination is possible through indirect[20] communication via a mediator, the profitable joint strategies $\langle s_1, s_2 \rangle$ and $\langle s_2, s_1 \rangle$ are attainable as *correlated equilibria* (Aumann, 1974; Aumann, 1987).

Correlated equilibria

Aumann's (1987) definition of correlated equilibrium is a generalization of mixed-strategy (Nash) equilibrium. Whereas, in mixed-strategy equilibria, the probability distributions that constitute the mixed strategies of agents are independent, this is not necessarily so in correlated equilibria. Metaphorically speaking, in a mixed equilibrium each agent tosses its own coins, while in a correlated equilibrium, some or all of the agents share them.

Myerson (1985, p. 252) describes correlated equilibria in the context of a mechanism, where a fully informed mediator (i.e., one that knows the strategy space and the outcome utilities of each agent) suggests a strategy to each agent. The mediator chooses an outcome (i.e., a strategy tuple) by applying a random procedure similar to that of a mixed strategy, informs the agents of the procedure (i.e., the probability distribution) that it has used to arrive at the outcome, and advises each of the agents which strategy it should select to bring about this outcome. It does not reveal the outcome itself. From the employed procedure and the mediator's advice, each agent can compute the expected utilities of its strategies under the assumption that the other agents will follow the mediator's advice. The probability distribution on the joint strategy space that is used by the mediator to pick an outcome is called a *correlated strategy*. A correlated strategy is a correlated equilibrium iff no agent can expect to gain utility by deviating from the mediator's advice, as long as the other agents follow the advice which the mediator gave them. A formal definition is given below. If f is a function with domain $S = S_1 \times \cdots \times S_n$, and $s \in S$ then $s_{\bar{i}}$ is short for $\langle s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n \rangle$, and $f(x, s_{\bar{i}})$ is short for $f(s_1, \ldots, s_{i-1}, x, s_{i+1}, \ldots, s_n)$.

Definition 5.11 (correlated equilibrium)
Let $g = \langle S, u \rangle$ be game in normal form with joint strategy space $S = \prod_{1..n} S_i$ and outcome utilities $u_i : S \to \mathbb{R}$. Let $p : S \to [0,1]$ be a correlated strategy. If the mediator advises agent i to use strategy s_i^*, and informs it of p, then the assumption (of agent i) that the other agents will follow the mediator's advice induces a subjective

[20] If we would consider coordination by *direct* communication between the agents, we would enter the field of cooperative game theory.

probability distribution $\hat{p}_i(s_i^*)$ on $S_{\bar{i}}$, the joint strategy space of the other agents. Let $\hat{p}_i(s_i^*)(s_{\bar{i}}^\sharp)$ denote the probability that agent i attaches to the event that the joint response of the other agents will be $s_{\bar{i}}^\sharp$. This probability is related to p and s_i^* in the following manner.

$$\hat{p}_i(s_i^*)(s_{\bar{i}}^\sharp) = \frac{p(s_i^*, s_{\bar{i}}^\sharp)}{\sum_{s_{\bar{i}} \in S_{\bar{i}}} p(s_i^*, s_{\bar{i}})} \tag{5.25}$$

Let $\hat{u}_i(s_i)$ be the stochastic variable that denotes the utility that agent i will obtain if it selects strategy s_i. Then p is a correlated equilibrium iff

$$(\forall s^* \in S) \qquad (\forall i \in \{1,..n\}) \ (\forall s_i \in S_i)$$
$$E\{\hat{u}_i(s_i^*)|\hat{p}_i(s_i^*)\} \geq E\{\hat{u}_i(s_i)|\hat{p}_i(s_i^*)\} \tag{5.26}$$

△

Example 5.12 (Correlated equilibrium)
Consider the game in Table 5.11a. If we denote the mediator's correlated strategy by the tuple $\langle p_{11}, p_{12}, p_{21}, p_{22}\rangle$ where p_{ij} denotes the probability that the outcome $\langle s_i, s_j\rangle$ will be selected, then any correlated strategy p^* of the form $\langle 0, p, 1-p, 0\rangle$ with $0 \leq p \leq 1$ (i.e., a "convex combination" of the pure-strategy Nash equilibria) is a correlated equilibrium: If agent 1 receives the advice to select s_1, it can deduce (from $p_{11} = 0$) that, with probability 1, the mediator has advised agent 2 to select s_2. Under the assumption that agent 2 will follow the mediator's advice, the expected utility of following the mediator's advice is $0 \cdot 0 + 1 \cdot 3 = 3$, whereas that of deviating from it by selecting s_2 equals $0 \cdot 2 + 1 \cdot 1 = 1$. Hence, it is not advantageous for agent 1 to deviate from s_1. Similarly, if agent 2 receives the advice to use strategy s_2, it deduces from $p_{22} = 0$ that agent 1 will use strategy s_1 with probability 1. Hence, also for agent 2, it is best to follow the mediator's advice. Note that, even in the unlikely event that the outcome of applying p^* is one of the zero-probability tuples $\langle s_1, s_1\rangle$, or $\langle s_2, s_2\rangle$, the agents will still follow the mediator's advice. △

In the above example, the total expected utility obtained at the correlated equilibrium $\langle 0, p, 1-p, 0\rangle$ equals $(3+2)\cdot p + (2+4)\cdot(1-p) = 6-p$, which is considerably more (for any $p \in [0,1]$) than the total expected utility of $\frac{3}{2} + \frac{8}{5} = 3.1$ that is obtained from the mixed-strategy equilibrium $\langle \frac{3}{5}, \frac{1}{2}\rangle$. However, the expected total utility associated with a "truly correlated" equilibrium is not always higher than that of a mixed equilibrium, and sometimes the mixed equilibrium is the only correlated equilibrium. In the game of Table 5.11b, for example, we can derive from Eq. (5.26) that a necessary condition for $p^* = \langle p_{11}, p_{12}, p_{21}, p_{22}\rangle$ to be a correlated equilibrium is $p_{12} \leq 2p_{11} \leq 2p_{21} \leq p_{22} \leq p_{12}$. This implies that $\langle \frac{1}{6}, \frac{1}{3}, \frac{1}{6}, \frac{1}{3}\rangle$ is the only correlated equilibrium. The associated total expected utility at p^* is $\frac{7}{6}$, the same as that of the mixed-strategy Nash equilibrium $\langle \frac{1}{2}, \frac{1}{3}\rangle$. This is not really surprising, since computation of the outcome probabilities associated with the mixed-strategy equilibrium shows that the it coincides with p^*.

Another advantage of the notion of correlated equilibrium over that of Nash equilibrium is that the correlated equilibrium does not require the agents to have perfect knowledge. To attain a correlated equilibrium, an agent need not know the outcome utilities of the other agents. In this respect, the suggestion in (Myerson, 1985, p. 252) that the notion of correlated equilibrium presumes complete information is misleading: In the above account, the *mediator* is completely informed with respect to the outcome utilities of agents, but the agents themselves are not. What the agents must know in the above model is the correlated strategy employed by the mediator, or at least that part of it that pertains to the mediator's advice to the agent. In the case of Ex. 5.12, this is the row of (the matrix representation of) the correlated strategy (p_{ij}) that corresponds with the mediator's advice. More generally, in a game with $n \geq 2$ agents with m strategies for each agent, a correlated strategy is an n-dimensional hypercube of m^n probabilities, and the knowledge that agent i must have of this hypercube is the intersection of this hypercube with the hyperplane $s_i = s_i^*$, where s_i^* is the mediator's advice to agent i. This section of the hypercube can be viewed as the joint, conditional probability distribution of agent i on the (pure) strategies that the other agents will choose, under the assumption that they will follow the mediator's advice.

The kinship between IG and correlated equilibria

Since the notions of pure and mixed Nash equilibria presume perfect knowledge, they bear little resemblance to the equilibria attained in IG mechanisms. The same applies to other refinements of the original notion of Nash equilibrium, such as subgame-perfect equilibria. Our above account of Myerson's interpretation of correlated equilibria reveals that IG equilibria (in the absence of negotiation weariness) are significantly more akin to correlated equilibria. First, neither concept requires agents to have perfect information. Second, both concepts involve indirect communication between agents. Third, and most important, we can rephrase the above characterization of the information supplied by the mediator as

> The mediator provides the agents with information (viz. the mediator's correlated strategy and the recommended strategy) from which the agent can derive a conditional probability distribution on the *joint* response of the other agents (i.e., on the *profile* of proposals that will be submitted), assuming that these responses will be in line with the's recommendations communicated by the mediator.

Likewise, one can describe the information supplied by the auctioneer in an IG auction as

> The auctioneer provides the agents with information (viz. the market profile) from which the agent can derive a conditional probability distribution on the *collective* response of the other agents (i.e., on the *bag* of proposals that will be submitted), assuming that this collective response will be in line with the market profile communicated by the auctioneer.

The similarities between these two characterizations are clear. It is also clear what the difference is: The rationality notions associated with the two frameworks differ in their *descriptive level* (cf. Sect. 5.4.3). The inferences of agents in the correlated-equilibrium model are on the level of individual agent behavior, while those of IG agents pertain to collective behavior.

As mentioned in Sect. 5.4.3, some rationality notions employ yet another level of description: that of inferences about the inferences of other agents.[21] This third descriptive level is used in the framework of Bayesian games, and for historic reasons,[22] it is still popular in MAS research.

In an informed world,	agents ask themselves	"Will some agent do Y?"
		⇑
In a correlated world,	agents ask themselves	"What will agent X do?"
		⇑
In a Bayesian world,	agents ask themselves	"What does agent X think?"

Table 5.12: The reasoning levels of Bayesian games, the correlated equilibrium model, and IG.

Summarizing, the spectrum of rationality notions involves three different levels of agent reasoning: hypothesizing on the hypotheses, the behavior, and the collective behavior of other agents. These are the descriptive levels of the Bayesian, correlated, and IG equilibrium notions respectively. A reformulation of this categorization, in terms of the kind of questions which the agents focus on, is presented in Table 5.12.

As suggested in Table 5.12, the Bayesian games model describes multi-agent decision process on the level of the agents' thought processes. Agents in a Bayesian game attach probabilities to what specific other agents believe (Harsanyani, 1968; Mertens & Zamir, 1985; Myerson, 1985). This is the basis of their inferences about the behavior of other agents, which, in turn, is the basis of their strategy selection.

[21] Inferring other agents' inferences amounts to the same thing as hypothesizing about their hypotheses, or holding beliefs about their beliefs.

[22] "Hypothesizing about hypotheses" is one way to express what (D)AI's symbolic paradigm entails.

The original description of the correlated-equilibrium model in (Aumann, 1974) is too abstract to associate it with any specific level of agent reasoning, but Aumann's later (1987) qualification of correlated equilibria as "the *result* of Bayesian rationality", and Myerson's (1985) account on the relationship between correlated equilibria and mechanism design (cf. Sect. 5.8.1) describe correlated equilibria on the level of agents attaching probabilities to the *behavior* of (specific) other agents, rather than to the agent beliefs which cause this behavior.

To arrive at estimates of proposal success probabilities, IG agents hypothesize about the submitted *bag* of proposals — rather than the proposal profile — behind a given market profile. As such, the agents' hypotheses pertain to collective, rather than individual agent behavior.

Apart from its level, the reasoning of agents in IG differs from that of agents in the correlated-equilibrium framework — and other game-theoretic models — in another respect. In the correlated-equilibrium framework, the agents exhibit perfect rationality. It is not evident that IG's fictitious rationality can be regarded as such.

5.8.2 Characterization of fictitious rationality

We characterize fictitious rationality by determining the extent to which it can be labeled as perfect. For the reader's convenience we recall the black-box definition of perfect Bayesian rationality provided in Sect. 5.4.1.

> An agent with uncertain information on the consequences of its decisions exhibits perfect rationality if it always chooses a strategy such that the conditional expectation of the utility of the outcome, given the available information, is maximal.

The above definition of perfect rationality is sufficiently sharp to *pose* the question whether fictitious rationality is perfect, but the question is rather hard to answer. The reason to suspect that fictitious rationality is not perfect is that fictitiously rational agents ignore part of the information that is available to them:

- the coordination messages $m^1, \ldots, m^{t-1}$ communicated to the agent by the coordination module prior to m^t;
- the information (or, more appropriately, knowledge) that the adjustment process will not terminate as long as the agent never submits the same proposal in two subsequent rounds.

The first kind of unused information can — in principle — be used to formulate hypotheses about the presence of certain proposals in the proposal profile which underlies the current market profile. Suppose, for example, that an agent which has submitted $\{c \rightleftharpoons a\}$ in all rounds has made the following observations with respect to, say, the last twenty rounds:

1. The demand for type-a tools was always 2.
2. The supply of type-a tools was always 1.
3. The supply of type-b tools was always 1.
4. The supply of type-c tools fluctuated between 1 and 3, has dropped to 1 five times, and is now 2.
5. The supply of tool type d fluctuated between 0 and 4, has dropped to zero two times, and is now 3.
6. The supply of tool type e fluctuated between 0 and 4, has dropped to zero three times, and is now 4.

From these observations, a smart agent might infer that its most likely adversary in the competition for the type-a tool is the agent endowed with a type-b tool, since the demand for a never changed, and this agent is the only agent — apart from itself — that has never refrained from bidding. In contrast, the probability which a fictitiously rational agent will assign to the event that its adversary is endowed with x is — in this case — proportional to the present supply of x. Hence, a fictitiously rational agent will regard the agents endowed with e or d as its most likely adversaries (with probabilities $\frac{4}{9}$ and $\frac{1}{3}$ respectively), and it will assign a probability of only $\frac{1}{9}$ to the event that its adversary is endowed with b.

This illustrates that ignoring the information present in the sequence of market profiles *can* have a significant impact on the computed estimates of proposal success probabilities. However, it seems to be rather difficult to *compute* the conditional probability that the competing proposal is $\{b \rightleftharpoons a\}$, given the above observations, let alone, to come up with a general formula for the subjective success probabilities of proposals that takes all information present in the sequence of communicated market profiles into account. Furthermore, there are situations in which the information embodied in the sequence of earlier market profiles is irrelevant. A simple example is that of an agent with negligible utilities for all tool types except for its first preference. It is difficult to assess how often this will be the case, but it is at least *possible* that the information in previous market profiles is seldom relevant in the sense that it leads to another strategy being selected.

In combination with intuitions based on our experience with the interactive IG testbed, the above argumentation leads us to the following *postulate*.

> The information present in the sequence of market profiles communicated before the current one is difficult to use, and seldom decisive for strategy selection.

This postulate is also plausible with respect to the other neglected information. With stationarity as the sole termination condition, an IG agent can be certain that the current round will not be the last one. The agent can use this certainty as an opportunity to respond strategically rather than tactically. This entails that the agent plans ahead, with the purpose of arriving at a more profitable market profile.[23] Obviously, such an endeavor constitutes an attempt to manipulate the other agents. It is, as yet, rather unclear how effective such a strategy can be within the context of an IG mechanism. It seems likely, however, that a random trial-and-error approach to manipulation will seldom be effective. To manipulate other agents effectively, an agent should at least have some capability of predicting how other agents will respond. Such prediction is even more cumbersome than the formation of hypotheses about the sequence of proposal profiles behind a given sequence of market profiles. As such, we contend that the above characterization of market-profile sequence information as "difficult to use, and unlikely to have significant impact" also applies to the manipulation of the other agents.

Summarizing, with our *declarative* semantics for perfect rationality, fictitious rationality is unlikely to be perfect in general. However, *procedurally*, it is much more akin to perfect rationality than to aforementioned forms of bounded rationality such as strategy-space filtering, or calculative rationality. Furthermore, there are probably many situations in which it *is* perfect. In view of this, we characterize fictitious rationality as *near-perfect*. A general description of near-perfect rationality is provided in the following informal definition.

Definition 5.13 (near-perfect rationality)
An agent exhibits near-perfect rationality if

(i) it always selects a strategy that maximizes the conditional expectation of its outcome utility, given a precisely demarcated portion *of the available information, and*

(ii) the complementary (unused) portion of the available information is difficult to employ, and unlikely to make a difference.

△

Note that, whereas the first half of this definition is sharp, the second half is inherently vague. Yet, we believe this definition to be useful. Not only does it single out the essential properties of fictitious rationality; it also helps to emphasize that the main idea behind fictitious rationality is widely applicable: it is often easy to compute a good *estimate* of the expected utility of an action, in cases where exact

[23]Here, 'more profitable' means that the expected utility of the proposal(s) that are optimal under this market profile exceeds the expected utility of the profile(s) which are optimal under the current profile.

computation of the expected utility would be extremely complex, or even impossible in finite time. Typically, such estimation involves discarding information that is principally there, and easy to demarcate, but difficult to employ, and unlikely to have a significant impact.

5.8.3 Game-theoretic properties of Informed Gambling

The different points of departure of MAT and game theory cause many game-theoretic notions and principles to be "not entirely appropriate" for MAT in general and "utterly inappropriate" for CMAT. Yet, to characterize IG, we would like to know whether IG's performance correspondence has certain game-theoretic properties.

We address two questions in this respect.

- Are solutions rendered by IG guaranteed to be individually rational?
- Are solutions rendered by IG guaranteed to be Pareto optimal?

We answer these questions only for the IG *reassignment* mechanism defined in Sect. 5.6.5. The first question is an easy one. The answer is "No, unless ...". The key ingredient of IG is "commitment under uncertainty". This implies that an agent must be prepared to take risks so as to achieve its goals. If things turn out badly, the agent may therefore end up with a tool of lower utility than its initial endowment. The recapitulation of Example 3.18 at page 196 provides an example: Agents 1 and 4 end up with a tool of negligible utility, whereas their initial endowment rendered them a utility of 10%.

However, this conclusion with respect to the *basic* mechanism of Sect. 5.6.5 is not generalizable to the class of IG mechanisms as a whole. One of the parameters that can be varied in our IG testbed is the maximal utility decrease an agent is prepared to accept. If we set this parameter to zero, then — obviously — the solution is guaranteed to be individually rational. However, such a constraint may incur a significant decrease in the *average* agent utility (which we chose as our primary measure of solution quality in Chapter 2). The severity of this effect is one of the issues that will be investigated experimentally in Chapter 7.

The second question, whether the solutions rendered by IG are guaranteed to be Pareto optimal, is more difficult to answer, but the answer itself is simple: no. An example is given in Table 5.13 below.

In the utility matrix of Table 5.13, the underlined values indicate the initial endowments of the agents, and the constants ε and δ are small, positive numbers. The dots in the matrix denote zero utilities.

U	1	2	3	4	5	6	7	8
a	100-ε	100	100	.	.	.	.	.
b	100	100-ε	.	100	.	.	.	.
c	.	.	δ	δ	.	.	.	.
d	.	.	.	.	δ	δ	.	.
e	.	.	.	.	100	.	100-ε	100
f	.	.	.	.	.	100	100	100-ε

Table 5.13: A problem with a Pareto-suboptimal IG solution.

The problem involves 8 agents, 4 of which are endowed with tools of utility $100 - \varepsilon$, that is, close to 100%, while the other 4 have an endowment of utility δ, close to zero. The problem is highly symmetrical. Agents 1 and 2 would like to exchange their tools, and so would agents 7 and 8. Agents 3, 4, 5, and 6 would like to obtain one of the tools of agents 1, 2, 7, and 8. The initial proposal profile associated with this problem is shown in Table 5.14a, and the associated market and scarcity profiles are shown in Table 5.14b.

agent	1	2	3	4	5	6	7	8
endowment	a	b	c	c	d	d	e	f
proposal	b	a	a	b	e	f	f	e

(a) initial proposal profile

tool	a	b	c	d	e	f
demand	2	2	0	0	2	2
supply	1	1	2	2	1	1
scarcity	1	1	-2	-2	1	1

(b) market profile and scarcities

Table 5.14: Proposal and market profiles after the initial bidding round.

From the market profile, agent 1 can derive that its most-preferred tool type (b) is contested, that there is one actual competitor for this tool type, and 6 potential competitors, 2 of which (viz., the possessors of type-e and f tools) are equally eligible as agent 1 itself. This implies that there is a positive chance — say p% — that its proposal $\{a \rightleftharpoons b\}$ will fail. In that case, it will receive one of the oversupplied tool types c or d in exchange for its a, and end up with zero utility. Hence, the expected utility of submitting $\{a \rightleftharpoons b\}$ equals $(100 - p)$%, which is less than the expected utility $100 - \varepsilon$% of $\{-\}$, if $0 < \varepsilon < p$. Hence, if we choose a value of ε between 0 and p, then agent 1 will submit $\{-\}$ in round 2. In view of the symmetries in the problem, the same conclusion can be drawn for agents 2, 7, and 8.

Agent 3 can deduce from the market profile that its most-preferred tool type (a) is contested, that there is one actual competitor for a, and 6 potential ones, 3 of which are of equal eligibility as agent 3. Hence, the success probability of $\{c \rightleftharpoons a\}$ — say q — is positive. Consequently, the expected utility of submitting $\{c \rightleftharpoons a\}$, $100q$%, exceeds that of submitting $\{-\}$ (viz., $\delta\%$), if we choose δ between 0 and $100q$. Let us do that. Then we can be sure — again, in view of the symmetry in the problem — that each of the agents 3, 4, 5, and 6 will stick to their previous proposals as long as agents 1, 2, 7, and 8 stick to theirs. However, we have just concluded that agents 1, 2, 7, and 8 will switch to $\{-\}$ in round 2, This implies that agent 3 switches to $\{-\}$ in round 3, since it can deduce from the market profile computed from round-2 proposals that tool type a is no longer supplied. and the same conclusion can be drawn for agents 4, 5, and 6.

Hence, the response pattern that unfolds is

In round 1 all agents propose to exchange their endowment for their most-preferred tool type.

In round 2 agents 1, 2, 7, and 8 switch to $\{-\}$, while the other agents stick to their previous proposals.

In round 3 agents 1, 2, 7, and 8 stick to $\{-\}$, and the other agents switch to $\{-\}$.

In round 4 all agents stick to $\{-\}$.

Apparently, the outcome is the empty reallocation. The final allocation equals the initial allocation. However, the initial allocation is Pareto-dominated by [*baccddfe*]. Hence, the outcome is not Pareto optimal.

5.9 Application of IG Mechanisms

In Sect. 1.1.3, Informed Gambling was characterized as a framework that is primarily a vehicle for fundamental study, but also offers some application prospects. In the next two sections, I will explain why and how I believe IG to be useful as a basis for the development of practical instruments to tackle real-life problems, in a CMAT and OMAT context respectively.

Because we shall not delve deeply into any specific real-life problem setting, the argumentation in the sequel is often conjectural. Yet, I believe that it is valuable, not only for future application of IG, but also because an experimental study (such as that of Chapter 7) loses much of its value in the absence of clear connections between the necessarily simple experimentation domain and the more complex problems which are encountered in practice.

5.9.1 Closed IG mechanisms

In the context of CMAT, IG can serve as a basis to design entire multi-agent mechanisms for constrained optimization. In this case, IG tackles optimization problems in the same manner as MOP, that is, by reformulating them as (re)allocation problems. If IG is to be applied to the — typically full-fledged — tool reallocation problems associated with real-life constrained optimization domains (cf. Sect. 2.1.3), fictitious agent rationality is too demanding. For one thing, it entails the exhaustive investigation of the strategy space. In full-fledged reallocation problems, the agents' strategy spaces are typically too large for such an approach to be efficient. Here, bounded rationality is called for. The kind of bounded rationality we deem to be suited in many optimization domains is strategy-space expansion. This entails that IG auctioneering is interleaved with — rather than preceded by — the translation of the optimization into a reallocation problem. Below, we use the problem domain of transportation across a finite-capacity network (Ex. 2.3 on page 23) to illustrate why and how such a technique must be used.

Consider a passenger routing problem like that in Ex. 2.3 where each passengers must be assigned a route across a unidirectional network with finite link capacities. Our description of fictitious rationality in Sect. 5.6 presumes that agents know the utility of every possible outcome. For an agent involved in reallocation, this implies that the agent must know the utility of any tool endowment. In the context of passenger routing, a tool endowment is a bag of unit transportation capacities over route segments. Hence, even in a small-scale problem with a network of, say, 20 route segments and a uniform transportation capacity of 9 units per segment, a complete specification of the tool bag utilities of a *single* agent comprises as many as 10^{20} utility values. The large majority of these tool bags have utility zero, since they do not contain any complete path from source to destination. However, even if we omit these zero-utility bags, the strategy spaces are still unnecessarily large. Especially if the destination D is close (e.g., adjacent) to the source S, it would be ludicrous to compute all the strategies (i.e., all bags comprising paths from S to D), before investigating the feasibility of transportation along the shortest path.

The alternative we propose, strategy-space expansion, entails that reformulation and solution are interleaved. An agent starts with a world model that comprises only a fraction of the network, possibly only a single shortest path from its present node to its destination node. From this restricted world model, it determines what its strategies and tool-bag utilities are. In the course of the relaxation process, the agent can expand its current world model by adding network paths, and expand its strategy and utility space accordingly. A somewhat more detailed account is given below.

Suppose that the agent needs to travel from S to D, that there is a path of (minimal) length 3 from S to D, comprising the network links a, b, and c, and that its current endowment is the bag (of network links) λ. In this case, the initial strategy space is the set of all possible decompositions (bags of elementary proposals) of the composite proposal $\{\lambda \rightleftharpoons \{a, b, c\}\}$, and its utility for a tool bag γ is 100% if $\{a, b, c\} \in \gamma$, and

0% otherwise. If the subsequent market profile reveals that its chances of obtaining an endowment that includes the entire path ($\{a, b, c\}$) to its destination are low, the agent can decide to expand its partial network model by including (possibly longer) paths of network segments composed of links which are less scarce. It then computes the associated tool-bag utilities — which are now no longer binary, but depend on the length(s) of the path(s) to D present in the tool bag —, and replaces some of its previous proposals by proposals that request transportation capacity along the newly added paths.

Obviously, the basic idea behind the alternative approach to transportation problems, interleaving problem translation with auctioneering, is also applicable to other constrained optimization problems. The *specific form* of bounded rationality, however, will *not* be the same, because the translation procedure — which is now an integral part of the agents' reasoning — will differ for different problem domains.

Even with strategy-space expansion, we still face the problem — mentioned earlier in Sect. 5.6.4 — that the exact computation of the success probability of a specific proposal *bag* is complex, due to the fact that the success probabilities of the constituent elementary proposals — for which Prop. 5.6 specifies a formula — are generally not independent. If one ignores this, and computes the success probability of the entire proposal bag by multiplying the success probabilities of the constituent elementary proposals, one generally introduces an error. Such errors are systematic: If they occur, a computed success probability is always an *optimistic* estimate of the actual (subjective) probability that all of the elementary proposals will be executed. The following example explains why this is so.

tool type	a	b	c	d	e	f
demand	1	2	1	2	3	0
supply	1	1	1	1	1	4
scarcity	0	1	0	1	2	-4

Table 5.15: Market profile associated with Ex. 5.14.

Example 5.14 (Naive computation of a success probability)
Let 007 be an agent who wishes to compute the success probability of the proposal bag $\{\{a \rightleftharpoons b\}, \{c \rightleftharpoons d\}\}$*, on the basis of the market profile shown in Table 5.15, and its recollection of having submitted the same proposal bag in the previous round. Let* $\widehat{P}\{\, \mathrm{exec}(c \rightleftharpoons d)\, \}$ *denote the subjective success probability of 007's proposal* $\{c \rightleftharpoons d\}$*, as defined by* $\widehat{P}_y$ *in Prop. 5.6. This is the conditional probability that the proposal succeeds, given the information embodied in*

1. *the last-sent market profile;*

2. the proposal bag submitted by 007 in the previous round;

3. the other proposals (in this case, $\{a \rightleftharpoons b\}$) which 007 intends to submit in the current round.

Although $\widehat{P}\{\ \text{exec}(c \rightleftharpoons d)\ \}$ is formally a conditional probability, we refer to it in the sequel as "the 'unconditional' success probability" to distinguish it from $\widehat{P}\{\ \text{exec}(c \rightleftharpoons d) \mid \text{exec}(a \rightleftharpoons b)\ \}$, which entails the additional condition that 007's other proposal $\{a \rightleftharpoons b\}$ succeeds. Irrespective of any dependency between proposals, we know that

$$\begin{aligned} \widehat{P}\{\quad & \text{Exec}(\{\{a \rightleftharpoons b\}, \{c \rightleftharpoons d\}\})\ \} = \\ & \widehat{P}\{\ \text{exec}(a \rightleftharpoons b)\ \} \cdot \widehat{P}\{\ \text{exec}(c \rightleftharpoons d) \mid \text{exec}(a \rightleftharpoons b)\ \} \end{aligned} \tag{5.27}$$

The interesting part of this formula is the 'conditional' probability in the right member. If our conjecture is correct, that is, if assuming independence never leads to a pessimistic estimate, then we should have that

$$\widehat{P}\{\ \text{exec}(c \rightleftharpoons d) \mid \text{exec}(a \rightleftharpoons b)\ \} \leq \widehat{P}\{\ \text{exec}(c \rightleftharpoons d)\ \}$$

In the present example, the 'unconditional' success probability $\widehat{P}\{\ \text{exec}(c \rightleftharpoons d)\ \}$ is easy to compute. Since the tool type d features supply 1 and demand 2, 007 has one competitor. The bag $\hat{\gamma}$ of potential adversaries equals $\mu^- \setminus \{d, c\} = \{a, b, e, f : 4\}$. Two of these seven potential adversaries (viz. b and e) are more eligible than 007's c; the others are less eligible. Hence, the 'unconditional' success probability of $\{c \rightleftharpoons d\}$ equals $\frac{5}{7}$. As for the 'conditional' success probability, the information present in the datum $\text{exec}(\{a \rightleftharpoons b\})$ is that the sole competitor for b apparently did not offer d or e in return, for in that case the proposal $\{a \rightleftharpoons b\}$ would have failed. This implies that, in the context of $\widehat{P}\{\ \text{exec}(c \rightleftharpoons d) \mid \text{exec}(a \rightleftharpoons b)\ \}$, there are only six potential adversaries for d, two of which (viz. b and e) are more eligible than 007's c. Hence, the 'conditional' probability that $\{c \rightleftharpoons d\}$ succeeds, given that $\{a \rightleftharpoons b\}$ does, equals $\frac{2}{3}$; less *than the 'unconditional' probability of $\frac{5}{7}$.* △

The example is clearly generalizable: If an agent deliberates the submission of the proposal bag β, β_2 is an arbitrary subbag of β such that $\emptyset \subsetneq \beta_2 \subsetneq \beta$, and γ_2 is a proposal in $\beta \setminus \beta_2$ — then the only information in the assumption $\text{Exec}(\beta_2)$ which is relevant for the success probability of γ_2 is of the form "Not more than ... tool types with an eligibility higher than X have been offered by competitors for tools requested in the proposals of β_2", where X is some tool type offered by the agent in some proposal of β_2. In other words, if an agent considers the submission of a proposal bag β, then

$$(\forall \beta_i \subsetneq \beta)\,(\forall \gamma_i \in \beta \setminus \beta_i)\ \ \widehat{P}\{\ \text{exec}(\gamma_i)\ \geq\ \widehat{P}\{\ \text{exec}(\gamma_i) \mid \text{Exec}(\beta_i)\ \}$$

Consequently, if we denote the 'unconditional' success probabilities of the elementary proposals $\gamma_1, \gamma_2, \cdots, \gamma_k$ in a proposal bag β by $P_1, P_2, \cdots, P_k$ respectively, and define $\beta_i \triangleq \bigcup_{j<i}\{\gamma_j\}$, we have that

$$\widehat{P}\{\ \text{Exec}(\beta)\ \} = \prod_{i=1}^{k} \widehat{P}\{\ \text{exec}(\gamma_i)\ \mid\ \text{Exec}(\beta_i)\ \} \ \leq\ \prod_{i=1}^{k} P_i \tag{5.28}$$

In words, Eq. 5.28 expresses that the unwarranted assumption of independence between the 'unconditional' success probabilities of the elementary proposals in an agent's proposal bag *never* leads to a pessimistic estimate of the probability that all of these proposals will succeed. Assuming independence either leads to a proper estimate for the probability that the entire bag is executed, or to an optimistic one. Intuitively, it is clear that the latter is more likely. Equality in Eq. 5.28 implies that the datum $\text{Exec}(\beta_i)$ contains no information at all. This is the case only if all proposals in β_i are certain to succeed.

More important than the likeliness of deviations is their magnitude. An interesting observation in this respect is that the impact of assuming independence on the estimation of expected utilities is inversely proportional to the size of the agent population. In the above example, where the population is small — between 2 and 15, and most likely around 5 agents[24] — the assumption of independence incurs a deviation in the success-probability estimate of about 0.05 (73% instead of 68%). In larger populations, the individual contribution of an agent to the total trade volume is (percentually) smaller, so the impact of misperceptions with respect to their own contributions is smaller also. Conversely, the impact increases proportionally with the average size of proposal bags. If we translate this to a concrete problem domain like transportation of passengers or cargo across a finite-capacity network, it entails that enlarging the *network* increases the average impact of assuming independence, while enlargement of the *population* decreases the impact. All in all, we stipulate that the dependence between the success probabilities of elementary proposals in a proposal bag is less of a problem than it seemed to be at first sight: At least in *some* cases,[25] ignoring the dependence of elementary success probabilities will have little impact on the computed expected utilities, and hence, little effect on the attained solution quality. The latter inference is somewhat tentative, but we shall verify it experimentally in Chapter 7.

The above account shows that it is principally possible to use IG as a vehicle to solve constrained optimization problems. It also reveals that IG mechanisms for full-fledged reallocation require more sophisticated agents than IG reassignment

[24] The estimate of 15 concerns the extreme case where all the other agents have submitted a single, unconditional proposal; the more realistic assumption that agents submit 2 exchange proposals on average leads to an estimate of 4.5.

[25] In the case of transportation: if the destinations tend to be relatively close to the sources, or if the network links have high capacities, and are heavily used.

mechanisms, or MOP mechanisms for reallocation. In this respect, IG is an attractive alternative to MOP in cases where application of the latter *falters*, for example because equilibria are hard to find or often absent.

The greater need for agent sophistication in IG is a consequence of the fact that problem *decomposition* is easier in (continuous) systems[26] of subsystems linked by one or more *continuous* 'adhesive' variables — such as artificial money in MOP — than in (continuous) systems where the adhesive variables are (essentially) *discrete* (such as tool scarcities in IG). This breakdown of (de)composability in the face of discretization is a very general phenomenon, that can be observed in seemingly unrelated research areas like (qualitative reasoning on) differential-equation systems, (the chain-store paradox in) game theory, and (the lottery paradox in) nonmonotonic logic (Lenting, 1992a).

5.9.2 Open IG mechanisms

We envision three architectural schemes to use IG mechanisms within OMAT, that is, as coordination modules for existing, external agents. The schemes differ in the degree to which the external agents are kept informed of the present state of IG's adjustment process. They are described below, in increasing order of informedness.

trusted brokers The first option is to use CMAT-IG agents as brokers for external agents in an OMAT-IG mechanism. In this scheme, an external agent communicates strategies and outcome utilities to its broker, who then perform the same computations as a CMAT-IG agent to select a strategy, submits the associated proposals, and communicates the submitted proposals and their expected utility to the external agent. Obviously, the private information of the external agents is not truly private in this scheme: At least some of it is communicated to the broker (which is why we speak of *trusted* brokers).

untrusted brokers A second option is to let the external agents communicate candidate proposals only, to which the broker responds by transmitting the associated success probabilities (under the last-communicated market profile). In this scheme, the external agents do not receive any market-profile information.

no brokers The third option is to have the external agents play the same role as IG agents in a CMAT-IG mechanism. As such, they interact directly with the auctioneer in this scheme.

The main difference between CMAT and OMAT is that, in CMAT, the system designer designs the entire system, while in OMAT, at least some of the agents are

[26] In practice, 'continuous' means that the system features variables with a large value space, such as the agent-utility variables in IG.

external, and therefore principally outside of the designer's control. As a consequence, an agent model for external agents is a descriptive rather than a normative model. An OMAT agent model for external agents describes how these agents are *assumed* — rather than designed — to behave. If an IG mechanisms is to be adequate within an open environment such as the Internet, these assumptions should be plausible in the following sense. As in CMAT, mechanism design in OMAT aims to ensure a certain level of performance. Typically, such a guarantee is based on the assumptions embodied in the agent model. Hence, the actual behavior of external agents should at least be approximately consistent with these assumptions. A suitable rule of thumb in this respect is that, in an open system, there should not be any *likely* deviations from the agent model which can incur a significant deterioration of mechanism performance.

It is difficult to translate this into general *structural* conditions for the adequacy of an agent model. Some 'golden standard' has arisen in DAI literature, however. This standard entails that the following assumptions on external agents are considered appropriate.

autonomy External agents cannot be *commanded* to behave in any specific manner.

rationality External agents behave in accordance with a single top-level goal: the maximization of their own satisfaction.

self-centeredness The top-level goal of an external agent does not involve any desire to please other agents, nor any other inclination that seems to have no other purpose than to prevent or mitigate anti-social behavior.

Whereas this golden standard is obviously not a good model for human behavior in general, it is a plausible model for *computational* agents in an open environment such as the Internet.[27]

The IG agent models proposed in earlier sections of the current chapter feature fictitious rationality in combination with increasing weariness, either as a precise behavior specification, or as a basis for boundedly-rational behavior. If we evaluate this basic IG-agent-behavior specification against the background of the above golden standard, there are three 'irrationalities' in IG behavior that seem to conflict with the standard.

amnesia IG agents ignore the information in past market profiles.

honesty IG agents do not attempt to manipulate the other agents by submitting false proposals.

fatigue IG agents exhibit weariness in case of prolonged negotiation.

[27] Its only serious deficiency in this respect is that it does not (explicitly) consider the possibility that an external agent may have been designed to wreak havoc.

In the first (trusted-brokers) scheme for OMAT-IG, none of these irrationalities are relevant, since the trusted brokers are internal agents of the IG mechanism, whose behavior can be specified by the auction designer. However, this scheme has the serious disadvantage that it does not guarantee the privacy of the external agents' utilities.

In the second (untrusted-brokers) scheme, which does safeguard this privacy, amnesia is not an issue, since it pertains to the computation of success probabilities, which is performed by the (internal) brokers. The other two irrationalities are relevant. Whereas the untrusted-brokers scheme is perfectly suited for IG reassignment mechanisms, the absence of market-profile information can constitute a serious hindrance for agents facing full-fledged reallocation problems. In the passenger transportation domain (see Sect. 5.9.1), for instance, the market profile can be a valuable guideline to select promising candidate routes for strategy-space expansion (cf. Sect. 5.9.1).

In the last (no-brokers) scheme, the external agents have access to all the information that is available to a CMAT-IG agent. Consequently, the aforementioned hindrance does not exist. The other side of the coin is that each of the three irrationalities conflict with the golden standard. In the sequel, we confine our attention to this scheme, since it is apparently the most problematic one with respect to the admissibility of its behavior assumptions.

The prospect of utility gain from speculation by hysteretic agents

It is relatively easy to conceive an example sequence of market profiles that provides sufficiently valuable clues on the underlying sequence of proposal profiles to allow a hysteretic agent to come up with a better response than a fictitiously rational one. However, it tends to be difficult to extract useful information from an 'average' sequence of market profiles, cumbersome to quantify such information in terms of success probabilities, and unfathomable to specify a strategy-selection procedure of acceptable complexity that is perfect for *all conceivable* profile sequences. The only effective method for such perfect information extraction that we can think of involves the enumeration of all problem instances consistent with the sequence of market profiles and the agent's private information. Obviously, the computational complexity of such a procedure is overwhelming.

Furthermore, even this scheme does not work properly, if the relaxation procedure features asynchronous parallelism (of the sort described in Sect. 5.7), since most market profiles comprise information that is actually outdated for some (*varying* subset) of the agents. In other words, in the context of asynchronous proposal submission, it is very unlikely that a hysteretic agent, no matter how sophisticated it is, will ever compute a completely correct estimate of the proposal profile behind the current market profile.

Finally, upon implementing and applying such a procedure, one may well discover that the information present in market-profile sequences is seldom sufficiently rich to make a difference in strategy selection.

From the above argumentation, we conclude that the prospect for a hysteretic agent to gain utility from speculation is sufficiently discouraging in the IG framework for the assumption of agent amnesia to be admissible in open systems.

The prospect of utility gain from manipulation.

In principle, it can be profitable for an agent to communicate false proposals (i.e., proposals that it cannot comply with) to *lure* other agents into relaxation. Consider an agent that would like to obtain some tool type, has had reasonable chances to obtain it in earlier relaxation rounds, but has seen its chances diminished in the last few rounds. The agent can try to jolt the relaxation process by offering large quantities of some scarce tool type X (which it does not possess), and withdrawing the associated proposals again in a later round. If "reaching a market equilibrium" is not among the termination conditions, the agent can do this safely: It knows that it will not be obliged to keep the commitments expressed in its proposals as long as it refrains from resubmitting its last proposal, and no deadline is impending.

Without focusing on any particular application domain, it is not feasible to predict how often such behavior will be worthwhile. This kind of manipulation can be viewed as providing an autonomous[28] dynamic system (viz. the community of *other* agents) with input during some time interval, after which the system is left to its own dynamics again. Hence, it depends on the attractors (and their basins) of the autonomous system whether the jolt has the desired effect. If there is but a single, global attractor, it will have no effect whatsoever. If their are multiple attractors, the jolt has an effect only if it is sufficiently strong to lead the process state into the basin of another attractor, and, depending on this attractor, the effect can be utility gain as well as utility loss.

In the passenger transportation domain, manipulation seems — at first sight — unlikely to be effective. Suppose that an agent faces the situation in which there is a very high excess demand for some link(s) on the shortest path p to its destination X, while the second-best path to X is much longer. Such an agent can gain considerably utility if it *succeeds* in manipulating the other agents to pick an alternative route disjoint with p. However, while it is easy for the manipulator to attain a *temporary* decrease of traffic across such links (e.g., by submitting false proposals that offer large amounts of transportation capacity across nearby links) the traffic on p is bound to increase again after the agent has withdrawn its false proposals. Hence, to succeed, the manipulator must somehow prevent the other agents from responding

[28] In the theory of dynamic systems, 'autonomous' means 'without external input'.

to the decreased scarcity of p-links that results from the withdrawal of its false proposals.

At present, I can think of only one way to accomplish this. This concerns the case in which all other agents behave tropistically, and there is a fixed deadline D for the relaxation process (possibly combined with other termination conditions), which is known to the agents. Say that the two termination conditions are stationarity and deadline transgression. In this case, postponement of the withdrawal of false proposals until round D is an effective manipulatory strategy: since the other agents do not get the opportunity to respond to this last-moment action, the manipulator will acquire the desired transportation capacity on p.

The remedy against this form of manipulatory behavior (which is extremely damaging to the solution speed and likely to incur a significant deterioration of the solution quality) should obviously be sought in the termination condition. If the deadline is random rather than fixed, the above manipulation strategy is ineffective: the manipulator is more likely to be committed to its false proposals than to reach its objective. Hence, for open systems, it is imperative to turn "termination due to deadline transgression" from a boolean-valued function of the round number into a stochastic one. To prevent frequent deadline transgression, the probability of such termination should be low or nil in an early stage of the relaxation process. A deadline value of D could, for example, be implemented as termination with probability zero in rounds 0 to $D/2$, and probability $(2r - D)/D$ in subsequent rounds r.

The prospect of utility gain from sustained wariness

Of the three irrationalities in basic IG agent behavior, the increasing-weariness characteristic constitutes the most flagrant deviation from the golden standard. It is an adaptation of fictitiously rational behavior that was incorporated with the sole purpose of ensuring timely termination of the negotiation process, and there is no a-priori bound to the magnitude of the utility concessions which it may incur. Decreasing wariness due to increasing weariness is a common — and often vital — characteristic in inter-human negotiations. It is less plausible, however, as a behavior characteristic of autonomous *computational* agents, since computer programs do not tend to experience fatigue. Moreover, a computational agent which does not exhibit weariness (in IG auctions where other agents do exhibit it) is bound to gain at least some utility from its deviant behavior in the long run, and since its tool-bag utilities are private, this cannot be detected by any other computational entity.

(Kraus *et al.*, 1995) tackle this problem by focusing on problems where time pressure ensures that lengthy negotiation is not merely costly in a computational sense, but also incurs a significant decrease of the utilities which the agents beget from the outcome. Thus, negotiation weariness is rational behavior in their approach, grace to the nature of the problem domain. However, in view of the computational power of present-day computers, I suspect that cases where the time pressure is sufficiently high to ensure that autonomous *computational* agents do not waste time in negotiation are rare. Furthermore, it will seldom be the case that the time pressure is equally strong for all agents. If a few of the agents are in much less of a hurry than the others, the former group can force the latter into submission. As we have shown in our discussion of trust formation within the Walrasian auction (in Sect. 3.3.8), this can lead to allocations of very low quality (in terms of the average agent satisfaction).

Another way to turn negotiation weariness into rational agent behavior is to associate a fee with proposal submission or proposal adjustment. In such a scheme, it is rational for agent *designers* to incorporate negotiation weariness, since the cost of taking part in the auction can become unnecessarily high if they don't. However, asking a monetary fee for auction participation is at odds with our design decision to abstain from using money (cf. Sect. 5.2). Furthermore, to ensure that the fee is sufficiently high to be effective in this respect, and not too high to allow for adequate relaxation, one needs to have at least some idea of the (unnormalized) tool-bag utilities of the agents. Hence, this approach is at odds with the requirement of informational decentralization.

The above OMAT requirement of a stochastic instead of a deterministic deadline provides a better solution to the problem. Without involving money, this too provides an incentive for agent designers to incorporate weariness, if — as seems likely — termination due to deadline transgression tends to render less satisfactory final outcomes than termination due to stationarity. The specific probabilistic definition of the deadline that was given above is less suited in this respect, since it provides an incentive for agent designers to postpone the employment of the weariness rule until round $D/2$. However, slightly different schemes (e.g., choosing a random value between 1 and D instead of the fixed value $D/2$ for the termination probability to increase) do not provide such an undesirable incentive.

5.10 Chapter Summary

In this chapter, we have formally defined IG mechanisms. We have explained what the notion of mechanism entails in our approach and how and why IG mechanisms differ from mechanisms in game theory, social choice theory and economics. These differences are mainly due to the different requirements of multi-agent technology: unlike the aforementioned mathematical fields, MAT mechanism design calls for performance to be expressed in numerical measures of solution quality and speed, rather than specific properties such as individual rationality and Pareto optimality. Similarly, where game theory searches for definitions of agent rationality that are adequate from the viewpoint of a mathematician who aims to model human decision-making, MAT calls for definitions which are *pragmatic* for software engineering. Within CMAT, this entails that efficiency comes first, and modularity second. Within OMAT, it entails that a balance must be sought between the conflicting demands of computational efficiency and descriptive accuracy.

These considerations led to the conception of *fictitious rationality* as a suitable specification of agent response to coordination messages in an iterative mechanism. We have shown that fictitious rationality is computationally feasible for IG reassignment mechanisms, and explained — by means of a transportation scenario — why it is too demanding for complex reallocation problems derived from constrained optimization problems. Such problem domains call for a form of *bounded* rationality that interleaves IG mechanism application with problem translation (from transportation to reallocation). Finally, we argued why the specific form of bounded rationality used in the transportation scenario is not suited for all transportation problems, let alone for all constrained optimization problems that can be reformulated as reallocation problems. As such, bounded rationality for IG reallocation mechanisms is domain-dependent.

Because of this domain-dependence, the agent response correspondence, which serves as the carrier of agent rationality in iterative mechanisms, is an *abstract* parameter in our general definition of IG reallocation mechanisms. However, IG mechanisms for *reassignment* problems do not require a domain-dependent rationality specification, and fictitious rationality is adequate for any reassignment problem. As such, we were able to provide a precise, concrete description of the agent response correspondence in IG reassignment mechanisms.

Starting out with a procedural definition of the mechanism's key component, the pseudo-composition protocol, we derived the corresponding declarative definition, in the form of a probabilistic input-outcome correspondence. The agent response correspondence was then defined by taking *some* of the information available to an agent into account, namely the information present in the most recently received coordination message.

Next to the agent response correspondence, the general definition of IG mechanisms features two other variable components: the termination criterion for the message passing process that takes care of proposal relaxation, and a definition of the relative eligibility of proposals as a function of tool scarcity.

The variability of the eligibility definition is not essential, but it creates the opportunity to experiment with different architectural variants, and assess the influence of the resolution of protocol rules on the quality of solutions. In contrast, the presence of a termination parameter is a grim necessity. The termination of the relaxation process in IG is a problematic issue. Our analysis revealed that the adjustment process in an IG reassignment mechanism may not terminate, even if a correlated equilibrium exists, and irrespective of whether the mechanism features synchronously parallel, asynchronously parallel, or sequential agent response. Although the adaptation process is more likely to terminate in the latter two cases, we provided an example which shows that termination is not *guaranteed*. Termination problems have been addressed by other researchers in DAI, and in various other fields that feature dynamic systems for problem solving, such as neural networks. We conceived *negotiation weariness* as the countermeasure of our choice. Negotiation weariness constitutes a relaxation of the requirement that agents behave rationally which is anthropomorphically plausible. In political negotiations, human agents tend to become less fussy as the negotiation drags on. This is approximately what happens with IG agents featuring negotiation weariness.

To characterize IG mechanisms in game-theoretic terms, we investigated game-theoretic equilibrium concepts with respect to their appropriateness in the context of iterative mechanisms, and IG mechanisms in particular. This led to the conclusion that, among game-theoretic equilibrium notions, the notion of correlated equilibrium is most akin to the equilibria in IG mechanisms. If the agents in an iterative mechanism exhibit fictitious rationality *without* weariness, the stationarity states of the relaxation process can be viewed as correlated equilibria.

However, one should not conclude from this characterization that IG is akin to game-theoretic mechanisms. In this respect, labeling IG equilibria as correlated equilibria is comparable to describing a giraffe as "a kind of cow" (with a relatively long neck): there are some communalities, but also important differences between equilibria in IG and correlated equilibria. For one thing, the correlated-equilibrium and the IG framework have different descriptive levels: Agents in the correlated-equilibrium framework reason about the behavior to be expected from other *individual* agents, while IG agents reason about *collective* behavior. Furthermore. the correlated-equilibrium framework assumes that agents exhibit perfect (Bayesian) rationality, while IG agents exhibit near-perfect rationality, in combination with weariness. The latter behavior characteristic is *required* to ensure the effectiveness of IG. Indeed, IG mechanisms differ fundamentally from mechanisms in game theory and economics, in these and other respects. The solutions rendered by IG mechanisms are, for example, neither guaranteedly individually rational, nor Pareto optimal.

While a game-theorist would not *dream* of proposing a mechanism that does not guarantee either of these properties, it is of little concern to us. A property like Pareto optimality would be 'nice' for IG, but it is nowhere near the top of our list. Our first and foremost criteria for MAT mechanism performance are solution quality (in terms of the average profit or agent satisfaction), and solution speed. These call for experimental, rather than theoretical validation.

Chapter 6

Experimentation

6.1 Chapter Overview

In this chapter, we discuss the means and methods of our experimentation with IG. Our means of experimentation is the Informed Gambling Reassignment Testbed (IGRT). As this name indicates, the experiments pertain to *reassignment* problems rather than general reallocation. The motivation for this restriction is that it enables us to perform an extensive, systematic exploration of the entire problem domain. This is not feasible for general reallocation problems, for two reasons.

First, the computation of an optimal solution is tractable for reassignment, but not for reallocation in general (Papadimitriou & Steiglitz, 1982). Hence, the evaluation of mechanism performance is problematic in the general-reallocation domain.

Second, reassignment problems are sufficiently simple to categorize them in terms of a small number of parameters. This allows us to be fairly certain that our experiments cover any conceivable kind of problem in the domain. Because of the far greater complexity of reallocation problems,[1] such a categorization is not feasible for general reallocation.

The contents of this chapter is as follows. In Sect. 6.2, we describe the structure of the interactive testbed and the semantics of the parameters that can be adjusted by the user. While the interactive testbed has been useful to obtain a preliminary impression of the influences of testbed parameters, we developed an batch-oriented version of the testbed to gather performance statistics. This version generates a desired number of problem instances with certain - adjustable - properties, and produces raw data for each investigated problem instance as well as statistics on the entire batch.

Because the batch-oriented testbed differs only marginally from the interactive one,

[1]The space complexity of reassignment is $O(n^2)$, where n denotes the number of agents, while that of general reallocation is $O(2^{2^n})$.

as far as the parameter interface is concerned, we describe the batch-oriented testbed by explaining how it differs from the interactive version. This is done in Sect. 6.3. Sect. 6.4 contains a survey of the questions that we address in our experiments. In Sect. 6.5, we describe the approach we use to address these questions experimentally, and in Sect. 6.6, we discuss some methodological issues to justify the chosen approach. Except for Sect. 6.6, all sections of the current chapter are recommended reading for a good understanding of the discussion of our findings in Chapter 7.

6.2 The Informed Gambling Reassignment Testbed

The interactive Informed Gambling Reassignment Testbed (IGRT) offers the opportunity to experiment with architectural variants of the basic Informed Gambling mechanism for tool reassignment described in Sect. 5.6.5.

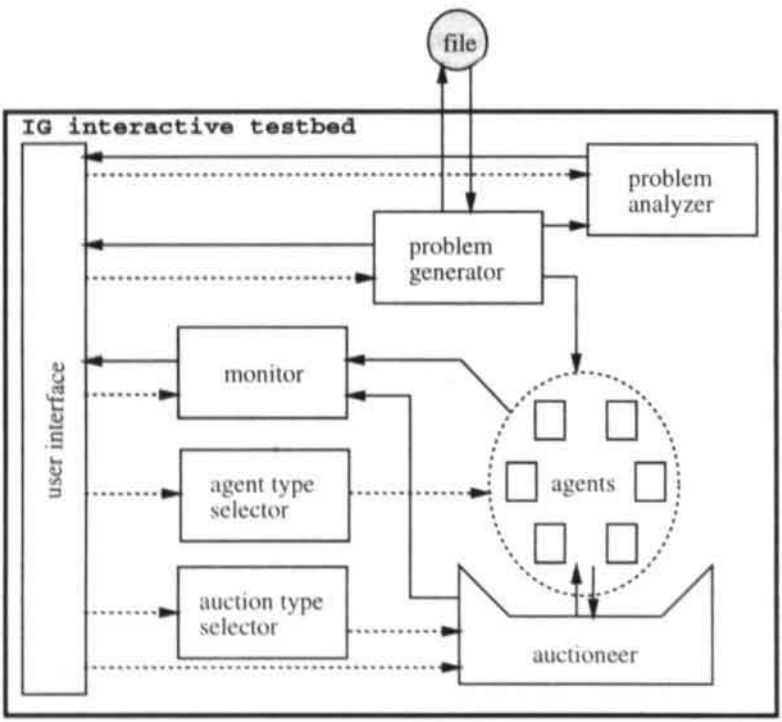


Figure 6.1: The structure of the IGRT.

Figure 6.1 shows the conceptual structure of the interactive IGRT and the communication that occurs between its components. Solid arrows denote data being communicated, whereas dotted arrows represent control via parameter communication. Below, we describe the testbed parameters componentwise. The aim is to enable the reader to understand our discussion of the experimental results. Hence, we do not provide implementational details.

The two most vital testbed components are the agent type selector and the auction type selector. The remainder of this section describes the semantics of the parameters associated with each of the components, and the semantics of the output generated by the auction monitoring and problem analysis components. A survey of the parameters associated with the various components is shown in Table 6.1.

component	**parameters**
problem generator	number of agents, number of tool types, tool bag, allocation, utility mode, utility file, utility range, zero utility, low alternative, high alternative
problem analyzer	walras mode
monitor	tracking mode
agent type selector	caution, volatility, maxloss, weariness
auction type selector	resolution, asynchrony, deadline

Table 6.1: Parameters associated with testbed components.

6.2.1 The problem generator

The problem generator is the component that defines the reassignment problem that is to be generated and solved. As apparent from Table 6.1, problem generation is parametrized by the size of the agent community ("number of agents"), the number of different tool types available within the community ("number of tool types"), the community's tool bag ("tool bag"), the initial allocation ("allocation"), and the utility matrix. The latter is either read from file or generated randomly, under the constraints defined by the problem generator parameters "utility range", "zero utility", "low alternative", and "high alternative". In the following, these and other parameter names will be typeset in `this font`, whenever they occur in the text.

Representation of tool types, bags, and assignments

Tool types are represented in the user interface by lower-case letters. Consequently, tool bags and assignments can be represented as lower-case strings. If the string denotes a bag, the order of letters in the string is irrelevant. If a tool bag is generated by the user interface (e.g., when the user changes the number of agents or tool types), the letters are displayed in alphabetic order. In an assignment string, the letter at position i denotes the tool type currently possessed by agent i.

Utility matrix generation

To further processing speed and storage economy, we represent real-valued parameters by integer approximations, whenever this is not too restrictive. Thus, agent utility, which is formally real-valued, is represented as an integer number. Moreover, the range of utility values is kept relatively small (between zero and 99 inclusive), to ensure that the presentation of a utility matrix in the interactive testbed is surveyable.

The primary parameter for utility matrix generation is `utility mode`, a toggle parameter which can be set to "file", "shuffle", or "random". In file mode, the utility matrix is read from the file `utility file`.

In shuffle mode, a new matrix is generated by permuting the rows of the current matrix randomly. This can be interpreted as agents being confronted with new tasks, while the bag of task types within the community remains the same. Within the IGRT, agent populations are homogeneous, in the sense that any two agents that take part in the same auction and have the same utilities and endowment exhibit the same bidding behavior. As such, any two such agents are effectively *identical* within the IGRT. This implies that shuffling a utility matrix is equivalent to permuting the assignment (cf. Fig 4.6 at page 121).

In random mode, the agent utilities which form the rows of the utility matrix are generated randomly, subject to the following parametrized constraints.

1. One randomly chosen tool type renders the maximal utility specified by `utility range`.

2. A percentage of the other tool types, specified as `zero utility`, render minimal (zero) utility.

3. All matrix entries that have not yet been assigned a value at this point are assigned a (uniformly distributed) random integer value between `low alternative` and `high alternative` (inclusive).

With `utility mode` set to "random", the generation of agent utilities is performed independently for different agents, but the same generation constraints are applied to all agents. In particular, the percentage of useless (zero-utility) tool types is the same for each agent. We refer to such a utility matrix with an equal number of zero's in each row as *homogemic*.[2]

[2] $\gamma\varepsilon\mu o\varsigma$ = burden.

The testbed user can also generate *heterogemic* utility matrices, that is, matrices in which the number of zero entries per row differs between rows. There are three ways to do this. The first one is to read a heterogemic matrix from file, and subsequently change utility mode to "shuffle". Thus, one effectively experiments with a single heterogemic matrix with varying initial allocations.

It is also possible to generate heterogemic matrices randomly. If a *negative* value z is assigned to `zero utility` — while `utility mode` = `random` —, the problem generator produces a matrix in which the first $n/2$ rows contain $-z\%$ zero entries, and the other rows $(100 + z)\%$. Note that both percentages pertain to $m - 1$, the number of tool types that do not render maximal utility to the agent.

Finally, it is possible to vary the number of zero entries per row in a less controlled fashion by setting `low alternative` to zero. Of course, the resulting variation will only be significant if the value of `high alternative` is relatively low. If, for example, an agent community comprises 10 agents with tools of 4 different types, a parameter setting of `utility range` = 2, `zero utility` = 0%, `low alternative` = 0, and `high alternative` = 1 will produce utility matrices with a number of zero entries per row between 0 and 3 (inclusive), with probabilities $\frac{1}{6}$, $\frac{1}{3}$, $\frac{1}{3}$, and $\frac{1}{6}$ for rows with 0, 1, 2, and 3 zero entries, respectively.

6.2.2 The problem analyzer

Whenever the reassignment problem faced by the community changes (i.e. when the user changes either the utility matrix or the initial assignment), the problem analyzer computes several *solution quality ranges*, which express the difficulty of the reassignment problem defined by the matrix and the current endowment. These ranges comprise the minimum and maximum solution quality of assignments subject to various constraints. Unless stated otherwise, solution quality is expressed in terms of the community utility, that is, the average *normalized* agent utility (see Sect. 2.3.1).

Depending on the setting of the `walras` parameter, the problem analyzer computes three to four ranges. The *unconstrained range* (UR) simply specifies the lowest and highest possible community utilities. The *autonomy range* (AR) is the community-utility range of assignments subject to the constraint that any agent which already possesses a tool of maximal utility keeps its tool. The *individual-rationality range* (IR) is the quality range of assignments subject to the constraint of individual rationality, that is, no agent utility is allowed to decrease due to reassignment. In problem instances that are generated randomly by the IGRT, there is a single tool type of maximal utility for each agent. In this case, any reassignment that is individually rational is also autonomous. Hence, if `utility mode` = "random", IR $\subset$ AR $\subset$ UR. If a problem instance is *viable*, in the sense of Def. 4.42, no agent is initially endowed with a tool of maximum utility. Hence, for viable problems, AR = UR. Because the batch-oriented testbed generates only viable problem instances, it does not compute autonomy ranges.

All of the above solution-quality ranges are computed by means of a polynomial-time algorithm, which is an adaptation of an existing algorithm for untyped assignment problems (Jonker & Volgenant, 1987) to *typed* reassignment problems with constraints.

If the `walras` parameter is set to "all", an additional range is computed. This *TTC solution range* indicates the lowest and highest possible community utility associated with Top-Trading-Cycle allocations. The TTC solution range is important as a reference for two reasons. First, as discussed in Sect. 3.3.5, the set of Walrasian allocations is a subset of the set of TTC allocations. Consequently, the TTC solution range is an upper bound on the allocation quality that can be expected from a Walrasian auction. Second, Top-Trading-Cycle allocations are also rendered in CDN (Sathi & Fox, 1989), a reallocation mechanism discussed in Sect. 3.2.1.

The motivation for conditionalizing the computation of the TTC solution range is that such computation is often costly. The computation of a single TTC solution is a matter of microseconds, but to compute the TTC solution *range* we must compute *all* TTC solutions. This requires another algorithm, and may take minutes, even for a relatively small reassignment problem involving, say, 15 agents and 5 tool types. A delay of several minutes per problem instance is clearly not acceptable, neither in interactive simulation of a single problem instance, nor in a batch comprising tens of thousands of problem instances.

If the `walras` parameter is set to one, the program computes a single Walrasian allocation. Using the stratification algorithm described in Sect. 4.3.3, a Walrasian allocation can be computed for stratifiable SPAMs within an (interactively) acceptable time interval. The (preferential) assignment market $\langle e, P\rangle$ associated with a (utilitarian) reassignment problem $\langle e, U\rangle$ is in SPAM iff the matrix U is *row monotonic*, that is,

$$(\forall i \in I)\,(\forall j, k \in R)\;\; j \neq k \;\Rightarrow\; U_{ij} \neq U_{ik} \tag{6.1}$$

Any utility matrix that is generated in the IGRT with moderate or high `zero utility` does not comply with (6.1), due to the presence of multiple zero entries per row. However, as we stipulated earlier in Sect. 4.7, the algorithm can be used also if all of the indifferences in agent preferences pertain to tool types which the agent does not prefer over its current endowment. Hence, multiple zero entries do not matter. For the algorithm to be applicable, it suffices that the matrix U is *weakly row monotonic*, that is,

$$(\forall i \in I)\,(\forall j, k \in R)\;\; (j \neq k \wedge U_{ij} \neq 0) \;\Rightarrow\; U_{ij} \neq U_{ik} \tag{6.2}$$

If the utility matrix is not weakly row-monotonic, the associated preferential assignment market lies in AM\SPAM. Such a market can have more than one Walrasian allocation. We could compute a Walrasian quality range by applying the stratifiability test to each of the markets in the SPAM enumeration of the assignment market, and compute the community utility for each of the Walrasian allocations found. However, this can incur excessive computational cost. We have therefore chosen to apply the stratifiability algorithm only to a single market in the SPAM enumeration. The preference profile of this market is the lexicographic monotonization of the preference profile associated with the matrix U. Formally, it is defined as follows.

Definition 6.1 (lexicographic monotonization)
Let $U = \{u_i\}_{i \in I}$ be a utility matrix that is not (weakly) row-monotonic, and let P denote the preference profile associated with U. Then the lexicographic monotonization $\tilde{P}$ of P is the tuple of strict agent preferences $\succ_i$, defined by

$$(\forall i \in I)\; x \succ_i y \;\equiv\; u_i(x) > u_i(y) \,\vee\, (u_i(x) = u_i(y) \,\wedge\, x > y) \tag{6.3}$$

Here, the symbol $>$ in $x > y$ denotes the usual alphabetic order on tool types (i.e., $a < b < c < \cdots$) △

If the SPAM market associated with a reassignment problem does not possess a Walrasian allocation, the IGRT renders "⟨ none ⟩" as the Walrasian allocation and -1 as its community utility. As apparent from the above, a message that no Walrasian allocation exists may be *formally incorrect* if the utility matrix is not weakly row-monotonic. The message does make sense, however, because — as we have shown in Chapter 5 — a Walrasian auction will have great difficulty in *finding* a Walrasian allocation, if the SPAM enumeration contains some, but *not many* Walrasian markets. Since this is typically the case, the notification "Walras sol.: ⟨ none ⟩" by the IGRT can be interpreted as "A Walrasian auction is likely to be ineffective for this problem.".

6.2.3 The auction type selector

The basic sequence of events in an IG auction was described formally in Sect. 5.6. For an assignment market, an informal account of the sequence of events is the following. An auction consists of one or more bidding phases. Each phase ends with a partial reassignment, after which the agents that have not acquired the tool specified in their last proposal are allowed to engage in additional bidding phases. A phase consists of the following steps.

1. The auctioneer invites the agents to propose their preferred exchange, and the agents respond.

2. The auctioneer aggregates the proposals into a market profile, which comprises the demand and supply of each tool type. This market profile is revealed to the agents.
3. The agents respond to the last sent market profile, by selecting a proposal of maximal expected utility, given their tool utilities, their current endowment, and their knowledge of the decision rules used by the auctioneer.
4. Steps 2 and 3 are repeated until either a *market equilibrium* (a state in which supply equals demand for all tool types), or a correlated equilibrium (a state in which the proposal of every agent equals the proposal it submitted in the previous round), or the deadline is reached.
5. Reallocation takes place according to the proposals and the market profile in the last round.

This basic sequence of events is parametrized in the testbed by four parameters, as described below.

Resolution

We recall from Sect. 5.5.3 that the eligibility of an exchange proposal is a non-decreasing function of the scarcity of the tool offered. The resolution parameter determines to what extent different scarcities give rise to different eligibilities. The IGRT offers a choice between three levels of resolution. Low resolution entails that the eligibility equals the *sign* of the scarcity. With high resolution, the eligibility equals the scarcity itself. Mixed resolution amounts to low resolution for oversupplied tool types and high resolution for scarce ones. The converse mix would be less appropriate for reassignment problems, because an agent cannot adjust its supply (since it only possesses one tool), while it can adjust its demand.

Asynchrony

The IGRT is a monolithic, sequential program. Hence, asynchronously parallel agent response in IG is *simulated* in the IGRT. Below, we explain *what* we simulate.

While the agents submit their proposals asynchronously, the auctioneer synchronizes the bidding to a certain extent by waiting for *some percentage* of the agents to submit a proposal before it reveals an updated market profile. The asynchrony parameter **async** equals 100 minus this percentage, with a minimum of one agent. As an example, in a population of 20 agents, **async** values of 0, 25, 50, and 100% correspond with new profiles being revealed after the auctioneer has received 20, 15, 10, and 1 proposal(s) respectively. The agents specify, together with each proposal, the round number associated with the market profile on which the proposal is based.

This enables the auctioneer to skip proposals that are based on an outdated market profile.

With asynchrony defined in this manner, high asynchrony implies that there is little parallelism; in the extreme case of maximal (100%) asynchrony, each round comprises the processing of only one proposal. Any other responses to the same market profile are simply discarded. Yet, proposal processing with maximal asynchrony is not the same thing as sequential processing. With maximal asynchrony, it is possible that all proposals processed in, say, the last ten rounds stem from one and the same agent. This never occurs with sequential processing; in this case, all submitted proposals are processed, and agents submit proposals in a fixed order.

The asynchrony parameter dictates the auctioneer's profile revelation behavior in all rounds, except for the first and the last round in a phase. Bidding in these rounds takes place synchronously, because the responses of *all* agents must be processed. In the first round, this is not really imperative, but it does not harm either. However, skipping agents in the last round can imply that they are asked or forced to accept an exchange to which they have not willingly committed themselves.[3]

The above definition of asynchronous parallelism differs essentially from definitions in literature on distributed processing. In (Bertsekas & Tsitsiklis, 1988), for example, parallelism is qualified as synchronous whenever the ratios of the response times of different agents are bounded by some constant. This is a useful definition to study issues like productivity in the context of possible processor failures, but it is not adequate for our purposes.

The asynchrony parameter may also be assigned a negative value. A value of $-v$ has the same meaning as a value of v, except that an updated market profile is revealed after each bid. Again, $100 - v$ percent of the agents bid in each round, in a random order, but the agents' bids are now based on a fully up-to-date market profile. This is almost the same as sequential bidding. The only differences are that the agents do submit proposals in a random order, and a round comprises the submission of $\frac{v}{100} \cdot |I|$ proposals instead of $|I|$. As such, comparison of the outcomes with `async` $= v$, and `async` $= -v$ enables us to assess the — possibly detrimental — effect of information backlog in the normal, positively asynchronous mode of operation.

Deadline

The declarative semantics of the deadline parameter is "the maximal number of rounds in an IG bidding phase". If negotiation weariness is not incorporated in the agent behavior, a phase is terminated by the auctioneer whenever round `deadline` is reached. In this case, the value of `deadline` is known only to the auctioneer. By

[3] Such noncommitment can occur if an agent's last proposal was based on a market profile that differs from the present profile.

default, negotiation weariness is incorporated in the IG mechanism of the IGRT, In this case, `deadline` determines how soon agents grow weary of proposal adjustment. Since `deadline` is essentially an agent parameter in this case, and the auctioneer need not know its value, we discuss its effect on agent behavior together with the other agent parameters in Sect. 6.2.4.

To assess the consequences of *omitting* weariness from the agent behavior, the testbed user can assign a negative value to `deadline`. In this case, agents will never grow weary of prolonged bidding, and the absolute value of the deadline parameter signifies the number of bidding rounds after which a phase is terminated (if it did not end earlier by reaching a correlated equilibrium or a market equilibrium). The IG protocol is then executed on the basis of the current proposal profile. If `deadline` is negative, it does not influence agent behavior. The agents then behave as if there were no deadline at all.

6.2.4 The agent type selector

The agent type selector determines the agents' attitude in bidding, especially with respect to risk. In Informed Gambling, the basis of an agent's rational responses to the auctioneer is the maximization of expected utility, under the constraints of fictitious rationality. Agents will submit a proposal to exchange their tool endowment x for an other tool y if their estimate Δu of the expected utility increase associated with this proposal is maximal, and positive. Δu is defined as

$$\Delta u(\{x \rightleftharpoons y\}) = \widehat{P}_y \cdot u(y) + (1 - \widehat{P}_y) \cdot U_- - u(x) \tag{6.4}$$

Here, $\widehat{P}_y$ denotes the subjective success probability of the proposal $\{x \rightleftharpoons y\}$ in view of the current market profile, as defined in Prop. 5.6 on page 191. $u(\cdot)$ represents the agent's utility function, and U_- denotes the expected utility upon failure, that is, the weighted average of the agent's utilities for the oversupplied tools.

However, (6.4) merely describes our definition of *basic* agent rationality. The agent rationality that is implemented in the testbed is a parametric variation thereof. The four parameters that modify the agents' rationality are discussed below.

Caution

The caution parameter determines how much optimism (or pessimism) the agents display when estimating the success probability $\widehat{P}_y$ (cf. Eq. 6.4). The default value of `caution` is 1. This renders an unbiased estimate P. Other values of `caution` correspond with optimistic or pessimistic estimates, in the following manner. Let P denote the unbiased estimate of proposal success probability obtained from Prop. 5.6

on page 191. If C denotes the agent's caution, then the distorted estimate $\wp$ equals

$$\wp(P,C) = \begin{cases} 1.0, & \text{if } C = 0.0 \\ P^C, & \text{if } C \geq 1.0 \\ P^{-1/C}, & \text{if } C \leq -1.0 \\ undefined, & otherwise \end{cases} \tag{6.5}$$

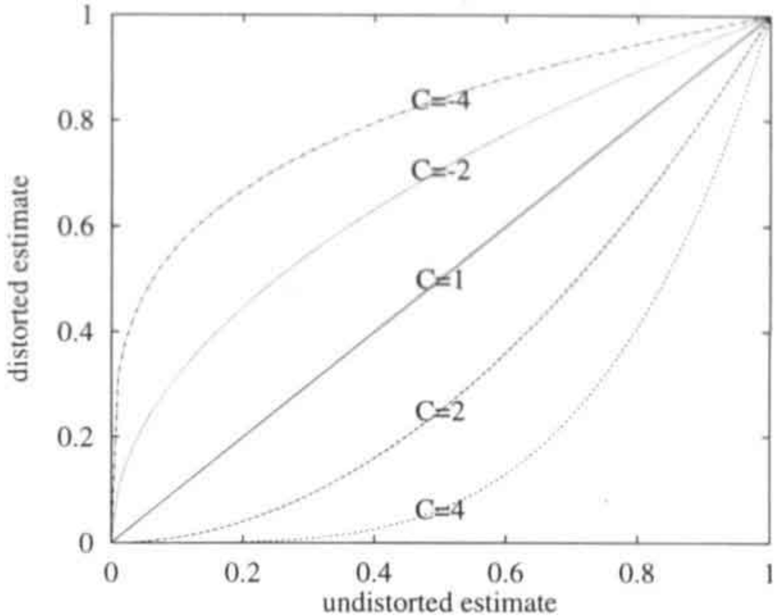


Figure 6.2: Probability estimate distortion as a function of caution.

As an example, for $C = -2, -1, 0, 1, 2, 2.5$, and 3, the distorted estimates $\wp$ equal $\sqrt{P}, P, 1.0, P, P^2, P^2\sqrt{P}$ and P^3 respectively. The effect of using `caution` values other than 1 is shown graphically in Figure 6.2. As apparent from the figure, values of `caution` above 1 render pessimistic (risk-aversive) agents, values below -1 render optimistic ones, and a value of 1 (or -1) leads to risk-neutral agents. A caution value of zero indicates that the agents are utterly reckless. They simply assume that every proposal will succeed. Caution values between -1 and 1, other than 0, are illegal.

By Eq. 6.5,

$$\wp(P,0) = \lim_{C \to -\infty} \wp(P,C)$$

Hence, if `caution`$= 0$, the agents are *less* cautious than if `caution`$= c < 0$.

In the interactive testbed, we have chosen to use integer-valued parameters whenever possible. However, as shown in Fig. 6.2, there is a rather large difference between the curves corresponding with adjacent integer values such as $C = 1$ and $C = 2$. We would prefer to be able to tune the agents' caution a little more finely. For this reason, the `caution` parameter in the testbed interface equals $10 \cdot C$ instead of C. Thus, the user can specify an inbetween value like $C = 1.5$ by setting the testbed's `caution` parameter to 15. Any `caution` values mentioned in the remainder of this chapter reflect the setting in the testbed interface. Thus, in the following, `caution`=10 corresponds with risk-neutral agents.

Maxloss

Human decision making involving risk usually does not only concern the probability of failure, but also the *impact* thereof. The maxloss parameter defines an acceptance limit in this respect. Whenever there is a possibility that the agent's (normalized) utility will decrease by more than `maxloss`% as a consequence of proposal failure, the agent will not submit the proposal. As an example, if `maxloss`= 50%, then an agent with a current utility of 80% will not submit any proposal with a (subjective) success probability below 1.0 if there exists an oversupplied tool type with a utility of less than 30%. The default value of `maxloss` is 100%. In other words, in the default setting, agents pay no attention to worst-case impact. In the following, we refer to an agent's "`maxloss` value" as its *daringness*.

Volatility

The volatility parameter determines the *desire for change* in the agents in the following sense. With `volatility` at its default value of 0, an agent will consider as viable alternatives ("options") for its current tool only those tools that render a higher utility. The effect of a positive volatility is that the agent depreciates its current endowment by subtracting `volatility`% of its normalized utility. As an example, with volatility set at 10%, an agent endowed with a tool of 80% utility will be interested in any tool type with a utility of more than 70%.

The motivation for the volatility parameter is that a nonzero volatility may cause an agent to submit a proposal, where it would normally refrain from doing so. Thus, a positive volatility can lead to better opportunities for *other* agents to improve their utility. As such, it can be regarded as a weak form of altruism: A volatile agent is prone to 'help' other agents, as long as it does not hurt its own utility (too much). Note, however, that this kind of altruism — unlike altruism in human agents — is not a matter of conscience or morals. The altruistic behavior is emergent; it does not require the agent to be *perceptive* of the needs of other agents.

Like `caution`, `volatility` is an integer-valued parameter. It denotes a percentage that corresponds with a floating-point variable $\tau \in [0, 1]$. In other words,

$\tau \triangleq \frac{\text{volatility}}{100}$, and its role in determining the set of options O_x for an agent with a normalized tool utility function $u(\cdot)$ and current endowment x is described by

$$O_x \triangleq \{y \in R \mid u(y) > u(x) - \tau\} \tag{6.6}$$

However, the depreciation τ of the agent's current endowment x extends beyond option selection. It is also applied when comparing the expected utility of a bid on y with the empty proposal '$-$'. With `volatility` at its default value of zero, an agent will prefer a bid on y over the empty proposal if $\hat{u}(y) > u(x)$.[4] With a positive value for `volatility`, this condition is relaxed to $\hat{u}(y) > u(x) - \tau$.

The fact that the raw (i.e., unnormalized) agent utilities in the utility matrix are *integer* numbers between 0 and `utility range` (inclusive) implies that different values of `volatility` sometimes have an identical effect on the option set, in *any* problem instance that is generated randomly with a specific setting of the problem generation parameters. Suppose, for example, that `utility range`$= 10$. Then every value for `volatility` in the set $\{1, 2, \ldots, 9\}$ will have the effect that the default option set $(\{y \in R \mid u(y) > u(x)\} \cup \{x\})$ of an agent endowed with an x is enlarged with the tool types $z \neq x$ such that $u(z) = u(x)$. More generally, different values V_i for `volatility` will lead to equal option sets if the associated rational numbers $\frac{V_i}{100} *$ `utility_range` have the same integral part, while none of them are integer numbers. However, this does not imply that such values V_i are completely equivalent. With the above value of 10 for `utility range`, $V_1 = 1$ and $V_2 = 5$ always render the same option set, but the ultimate choices *from* the option set may differ: It may occur that V_2 leads to a bid on $y \neq x$, while V_1 leads to '$-$' (viz. if $\tau_1 < \hat{u}(y) - u(x) < \tau_2$).

While values of $\tau \gg \frac{1}{\text{utility range}}$ are likely to incur a significant decrease of agent utility, a small, but positive value will not do much damage in this respect. In fact, a `volatility` of 1 amounts to enlarging the agents' option sets with those tool types that render the *same* utility as the ones that are currently allocated, and can therefore decrease the agent's utility only if its proposal is rejected.[5] Because a positive volatility can be proficient for other agents, especially those with few options, a `volatility` of 1 may well lead to a higher average community utility.

Weariness

If the value of the deadline parameter is positive, negotiation weariness is incorporated in the agents' bidding behavior. By default, this is the case. negotiation weariness entails that agents gradually grow weary from prolonged negotiation. In

[4]This pertains to the comparisons made to select a bid of maximal expected utility, that is, *before* the agent's weariness and daringness are taken into account.

[5]This is true for any value of `utility range` that is allowed in the testbed ($\{2, \ldots, 99\}$).

the long run, they do not take the trouble anymore to adjust their proposal to changes in the market profile which have little impact on the utility they can expect to obtain. Thus, weariness can influence the behavior of agents that contemplate a proposal which differs from their previous one. Before it actually submits a deliberated proposal (i.e., a proposal which is optimal in the sense of Eq. 6.4, taking into account the values of `caution`, `maxloss`, and `volatility`), the agent checks whether the excess (normalized) utility it expects to gain by deviating from its previous proposal exceeds its current indifference level. It can adapt its proposal (i.e., submit the deliberated proposal instead of its last-submitted one) only if this indifference level is exceeded. We shall refer to this as the *weariness constraint*. and to the agent's indifference level as its weariness treshold. The weariness treshold starts at zero, and increases linearly with the round number. If the round number reaches the deadline, the agent has become completely indifferent. Thus, the termination of the proposal adaptation process due to stationarity is guaranteed to occur at or before the deadline. Formally, the weariness treshold L equals

$$L = \frac{r}{D} \tag{6.7}$$

Here, r denotes the current round number, and D the value of the deadline parameter.

It is possible that the weariness constraint conflicts with the daringness constraint specified by `maxloss`. By sticking to its previous proposal, the agent may run some risk (which it did not run under the *previous* market profile) of ending up with a tool type with a normalized agent utility that is more than `maxloss`% below that of its current endowment. Any such conflict is treated in the following manner. Say that a bid on tool type x would render maximal expected utility, but weariness dictates the agent to stick to its previous bid on y. If sticking to its bid on y conflicts with the daringness constraint, the agent deliberates the alternative to submit an empty proposal, that is, to stick to its current endowment z. If the difference between the expected utility of its bid on x and its utility for z does not exceed its weariness treshold, it submits '−'. Otherwise it violates the weariness constraint and bids on x. We choose to violate the weariness, rather than the daringness constraint, because this does not endanger the desirable property of bid convergence at or before round `deadline`.[6] In contrast, a violation of the daringness constraint can be final, in the sense that it may lead to a final outcome which comprises a decrease of normalized agent utility of more than `maxloss`%.

[6] except, possibly, when `deadline` is extremely low.

6.2.5 The monitor

The value of the tracking parameter determines to what extent the auction is monitored. If `tracking` = `off`, then only final auction results are displayed. With partial `tracking`, the monitor also displays the proposal and market profiles after each round. The subwindow marked `auction status` in Fig. 6.3 shows what this monitor output is like, for round 2 of phase 1. If the user opts for full `tracking`, the monitor also displays a detailed account of the deliberations of each agent that has contributed to the market profile (cf. Sect. 6.2.3). An example is the subwindow marked `agent 2` in Fig. 6.3, which reveals the deliberations of agent 2 in round 3 of phase 1.

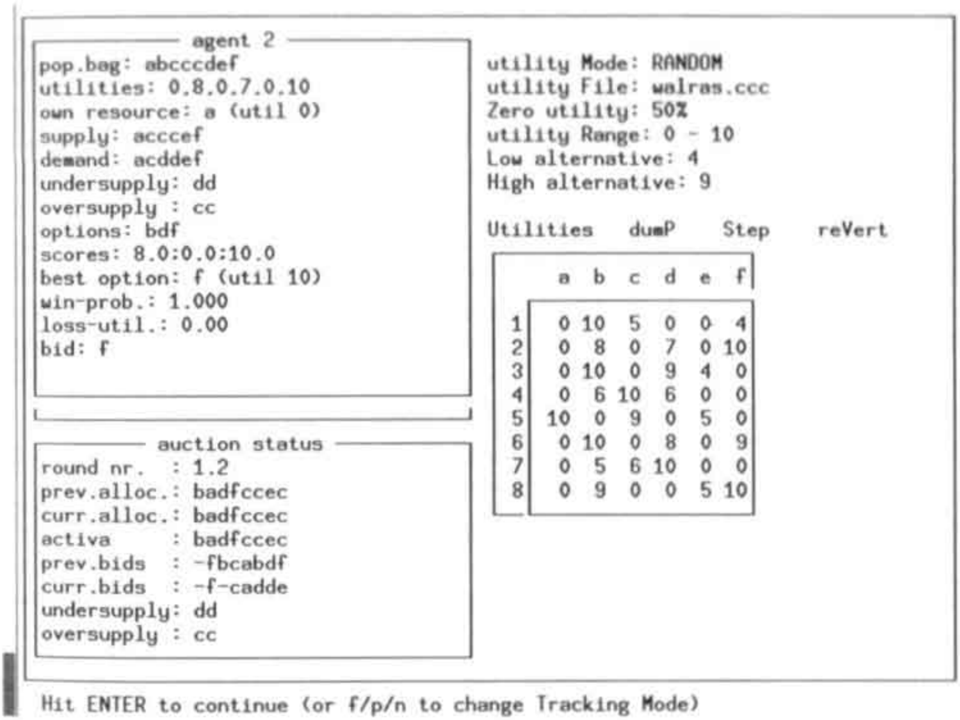


Figure 6.3: Screen image of monitor output with full tracking.

For the evaluation of IG performance, the final auction results are all that matters. Hence, we only discuss the output rendered by the monitor if `tracking` = `off`.

The auction results reported by the monitor comprise the following information:

1. the initial assignment, with its community utility, and its effectiveness;

2. the final assignment, with its community utility, and its effectiveness;
3. the total number of rounds in all phases;
4. the number of rounds in the longest phase;
5. the phase termination string (T_p);
6. the total utility decrease in the population (U^-);
7. the worst agent utility decrease (u^*_-);
8. an agent that suffered this worst decrease;
9. the highest agent-utility concession due to weariness (u^*_c).

The phase termination string consists of characters 'd', 'm', and 'b', which indicate the (primary) termination condition that led to the termination of each phase. As an example, T_p = 'dbm' signifies that the first phase ended due to the deadline being reached, the second due to bid convergence (i.e., reaching a correlated equilibrium), and the third and last phase due to market equilibrium. The order in which the IGRT checks these three sufficient termination conditions is b-m-d. This implies that a phase that ends in a state which is a correlated equilibrium as well as a market equilibrium is reported to have ended due to reaching a correlated equilibrium.

The total utility decrease in the population is the summation of the decreases in normalized agent utility over all agents whose utility decreased.

The highest weariness concession u^*_c is defined as follows. Let t_{ij} denote the tool type mentioned in the proposal by agent i in round j, where — for simplicity — we do not reset the round counter upon commencing a new phase. Let t^*_{ij} denote the tool type that would have been mentioned by agent i in round j if the weariness constraint would not have been applied by the agent in that round, and let $\hat{u}_i(x)$ denote the expected utility of a bid on x in the face of the market profile of round i.[7] Then the highest *weariness concession* of that auction equals

$$u^*_c \triangleq \max_{i,j}(\hat{u}_i(t^*_{ij}) - \hat{u}_i(t_{ij})) \tag{6.8}$$

The other descriptions of auction results are presumed self-explanatory.

[7] Here, "the profile of round i" should be interpreted as "the profile that is used by the agents in round i", that is, the profile computed from the proposals in round $i-1$.

6.3 The Batch-Oriented Testbed

The batch-oriented version of the IGRT differs from the interactive version in three respects.

1. Some design flaws in the interactive version were corrected.

2. The interactive user interface is replaced with a command-line interface.

3. The batch-oriented version calculates statistics on various performance attributes.

The design flaws in the interactive version are a consequence of the fact that the interactive testbed was conceived in an early stage of the project, before the first versions of chapters 4 and 5 were written. Consequently, the insights gathered in the process of writing these chapters have not influenced its design. The design improvements implemented in the batch-oriented testbed are the following.

problem viability In the interactive testbed, generated problem instances are not always viable. In the batch-oriented testbed, they are. As a consequence, the distinction between the unconstrained and the autonomy range (of community utility) no longer exists.

user-specified matrices If `utility mode = shuffle`, the interactive version always generates the first utility matrix randomly. In the batch-oriented version, the matrix is generated randomly only if the file specified by `utility file` does not exist.

market clearance In the interactive testbed, market clearance (i.e., reaching a state of equilibrium between supply and demand) is always a sufficient condition for phase termination. By default, this is not the case in the batch-oriented version. The user can, however, activate the test for market clearance by assigning a negative value to `maxloss`.

The command-line interface of the batch-oriented testbed is much simpler than the user interface in the interactive version, because testbed parameters are not set in the command line. Instead, they are read from a parameter file that can be constructed by means of an auxiliary program. In this program, the testbed parameters can be set via an interface similar to that of the interactive IGRT.

The command-line interface comprises the following options.

1. The user must specify a parameter definition file.

2. The user can specify the sizes of the sample sets.

3. The user can control how the random number generator that is used to generate problem instances is initialized.

4. The user can control how the random number generator that is used to take nondeterministic decisions and simulate asynchrony is initialized.

5. The user can specify that the results of the individual auctions should be saved to a file.

6. The user can specify that either the k-th problem instance of the sample, or the first k instances used should be written to a file.

7. The user can ask for a trace of each auction, which shows the allocation effectiveness that would have been obtained, if the deadline were invoked after round k (for $k = 1, 2, 3, \ldots$ up to the final round).

8. The user can choose between six modes of simulation.

A few of these added features require some clarification.

(3): The control that can be exerted on the random number generator used for problem instance generation can be used to ensure that different invocations of the batch testbed use the same set of sample problems. This is desirable when comparing IG performance with different settings of some agent or mechanism parameter.

(4): The control over the initialization of the other random number generator serves little purpose for performance evaluation. It was incorporated primarily as a debugging option.

(5): Storing the results of individual auctions enables the computation of frequency histograms, next to statistics. This will prove to be a valuable feature in the sequel.

(6): In combination with options 3 and 5, this options enables us to retrieve specific problem instances in the sample (e.g., a sample problem on which IG performed very badly), without having to record all problem instances in the sample.

(7): Effectiveness tracing was incorporated to get an impression of the efficiency of the collective search process. It proved to have little information value: the effectiveness trace moves up and down erratically, and the final effectiveness is often not the highest one in the trace. In retrospect, it is a bit foolish to expect any search efficiency from an algorithm that comes down to a heuristic random walk.

(8): The six simulation modes constitute the most important feature of the batch-oriented testbed. They are explained below.

linear sampling: The program gathers statistics (minima, maxima, averages and variances) on the performance of IG on the indicated number of problem instances, all of which are generated randomly within the constraints imposed by the values of problem generation parameters.

grouped sampling: The program gathers statistics as with linear sampling, except that the sample consists of several subsamples, and the variance is computed from the subsample averages instead of data from problem instances. This reduces the variances and ensures that the stochastic variables from which samples (i.e., subsample averages) are taken are (approximately) normally distributed. This is convenient for the assessment of the statistical significance of observed differences in performance.

fixed matrix-simulation The program gathers statistics (by means of grouped or linear sampling) on a specified number of viable problem instances, which share the same utility matrix.

exhaustive fixed matrix-simulation The program gathers statistics (by means of linear sampling) on *all* viable problem instances associated with a specific utility matrix.

tentative fixed-instance simulation The program gathers statistics on repeated application of IG to a single problem instance, thus portraying the extent to which the nondeterminism in Informed Gambling is responsible for the variances of performance attributes. The invocation of IG is repeated as long as the number of invocations is less than k times the number of different solutions encountered (for some specified positive value of k).

systematic fixed-instance simulation The program gathers statistics on a specified number of invocations of IG to each of a specified number of randomly generated problem instances. The statistics, in this case, are the number of different solutions found, and the minimal, maximal, and average values of the primary performance attributes (solution effectiveness and auction duration in terms of the total number of rounds).

The performance attributes on which statistics are gathered comprise most of the final auction results reported by the monitor in the interactive testbed. A complete survey of the performance attributes of which the IGRT computes statistics is listed in Table 6.2. We distinguish between three types of performance attributes. These types differ in the number of values which the attribute can have if the experiment involves *only one* auction. In such a case, a cardinal attribute can have one of many values, a boolean attribute only one of two, and a conditional attribute need not have a value at all. We provide an example of each type.

attribute label	attr. type	(sample) semantics
effectiveness	cardinal	(avg. of) effectiveness of the final allocation
rounds	cardinal	(avg. of) total number of rounds in all phases
zero-loss	boolean	(% of) auction(s) without utility loss
losers	conditional	(avg. of) % of agents which lose utility
average loss	conditional	(avg. of) average utility loss among losers
highest loss	conditional	(avg. of) highest utility loss
zero-weariness	boolean	(% of) auction(s) without weariness
weariness concession	conditional	(avg. of) highest weariness concession
convergence	boolean	(% of) phase termination due to bid convergence
clearance	boolean	(% of) phase termination due to market clearance
deadline	boolean	(% of) phase termination due to deadline excession

Table 6.2: Performance attributes covered by the batch-oriented IGRT.

cardinal: The `rounds` attribute for a single auction can — in principle — have any integer value greater than one.

boolean: In case of a single auction, the value of the `zero-loss` attribute can only be 0% (signifying that some agent(s) experienced utility loss) or 100% (if no utility loss occurred in the auction). Note that the phase termination attributes pertain only to the termination of the *first* phase.[8] Hence, they too are boolean attributes.

conditional: The `average loss` attribute is defined only if the auction features `zero-loss = 0%`, that is, if at least one agent utility decreases due to reassignment. The same applies to the other conditional attributes. Hence, the semantics of `losers` for a sample of auctions is "the average percentage-of-agents-in-the-population-who-lost-some-utility over those auctions in the sample in which utility loss occurred".

The reason that we distinguish between different attribute types in this manner is that our estimates of the mean value of an attribute tend to be relatively reliable for cardinal attributes, but less so for boolean and conditional attributes. This is due to the fact that the IGRT computes statistics in a uniform manner, without paying attention to the type of performance attribute. We return to this issue when discussing the computation of confidence limits in Sect. 6.5.4.

The primary performance attributes are solution effectiveness, and auction duration (in terms of the total number of rounds in all phases). In the discussion of experimental outcomes that follows, these two primary performance attributes are always included. Secondary performance attributes are discussed only if the associated outcomes are surprising or useful to explain the statistics on the primary attributes.

[8]In virtually all parameter settings, this is by far the longest, and hence, the most relevant phase.

6.4 Questions to be Answered

The batch-oriented version of IGRT has been used to perform a wide variety of experiments. The questions that motivated this experimentation roughly comprise six categories.

(A) How well does the default IG mechanism perform on the default problem space?

(B) To what extent does IG performance depend on the kind of problems to which it is applied, that is, on the setting of the problem generation parameters?

(C) Is the chosen set of problem-generation parameters adequate (1) to ensure that the generated samples are not too easy, and (2) to characterize the tool-reassignment problems that are most difficult for IG?

(D) How does the setting of the agent and mechanism parameters influence the performance of IG?

(E) How well does IG perform in comparison with other reassignment algorithms, such as the Walrasian auction, delegated negotiation, and mediated negotiation?

(F) What portion of the variation in IG performance is due to nondeterminism, and to what extent can this be reduced by changing the mechanism parameters `asynchrony` and `resolution`?

Preliminary experimental investigation of the questions in category (D) revealed that many parameters did not exert a *statistically significant* influence on solution quality, due to very large sample variances. There are two possible causes for this phenomenon: Possibly, the influence of parameter changes varies with the kind of problem instance in a manner that is not captured by our problem generation parameters. This prompted us to pose the questions in category (C). On the other hand, the nondeterminism in IG could be a major source of variance. The degree of nondeterminism, in turn, is liable of being influenced by some of the mechanism parameters. This prompted us to pose the questions in category (F).

6.5 Methodology

6.5.1 General approach

In most cases, we have used the following approach to answer the questions formulated in the previous section.

1. Pick a suitable sample space.

2. Draw a sample of problem instances from this space.

3. Produce performance statistics (for IG and/or other mechanisms) by applying the mechanism(s) to each problem instance.

The reason that we draw problem instances from specific sample spaces, rather than simply using the overall problem space as our sample space, is that the latter approach tends to lead to samples with too many easy problems. This is explained more elaborately in Sect. 6.6.

6.5.2 Exploration of the parameter space

Since the caution parameter is essentially real-valued, and the deadline parameter is principally unbounded, the parameter space is — at first instance — uncountably infinite. By discretizing `caution` and bounding `deadline`, the IGRT turns this space into a finite one. However, this finite space is still too large to gather performance statistics with every allowed parameter setting. We have therefore limited our exploration of the parameter space by focusing on a default setting of IG's agent and mechanism parameters, and a default sample space. As an example, nearly all questions in category D (on the dependence of IG performance on the parameter setting), are addressed by using this default sample space, and investigating the effects of changing one parameter at a time, with all other parameters at their default value.

The defaults for the parameter setting are based on observations during experimentation with the interactive version of the IGRT. They are listed in Table 6.3, together with the value ranges of the parameters.

parameter	IGRT value range	default value
caution	$\{-80, \ldots, -11, -10, 0, 10, 11, \ldots, 80\}$	10
volatility	$\{0, 1, 2, \ldots, 100\}$	0
maxloss	$\{0, 1, 2, \ldots, 100\}$	100
weariness	{`TRUE`,`FALSE`}	`TRUE`
deadline	$\{1, 2, \ldots, 2000\}$	500
clearance	{`TRUE`,`FALSE`}	`FALSE`
resolution	{`LOW`, `MIXED`, `HIGH`}	`MIXED`
asynchrony	$\{0, 1, 2, \ldots, 100\}$	50

Table 6.3: Value ranges of agent and mechanism parameters.

The clearance parameter in Table 6.3, is only present in the batch-oriented version of the IGRT. It indicates whether market clearance is treated as a sufficient condition to terminate a phase. As apparent from the table, this is not the case in the default setting. The limits imposed on the value ranges of `caution` and `deadline` are more or less arbitrary. Experimentation with the interactive testbed had made clear that it would not be worthwhile to expand these ranges.

6.5.3 Sample space selection

Like the parameter space, the IGRT problem space is already constrained to a finite grid on the infinite space of reassignment problems by the architecture of the IGRT. Problem dimension (i.e., the size n of the population and m of the tool set) is confined to the set $\{(n,m) \parallel 2 \leq n \leq 20 \wedge 2 \leq m \leq n\}$, and the entries of utility matrices are confined to the finite set $\{0, 1, 2, \ldots, 99\}$.

A sample space is a subspace of the problem space from which the IGRT draws its sample of problem instances. With `utility mode` set to `random`, the sample space is defined by the setting of the problem generation parameters `nr of agents`, `nr of tool types`, `tool bag`, `utility range`, `zero utility`, `low alternative`, and `high alternative`.

parameter	symbol	value range	default value
nr of agents	n	$\{2, \ldots, 20\}$	12
nr of tool types	m	$\{2, \ldots, n\}$	5
tool bag	Γ	$\mathfrak{B}(n, m)$	$\{a, a, a, b, b, b, c, c, d, d, e, e\}$
max. utility	M	$\{1, \ldots, 99\}$	10
zero utility		$\{-100\%, \ldots, 100\%\}$	-25%
nr of zeroes	Z	$\{0, \ldots, m-1\}$	1(/3)
high alternative	H	$\{1, \ldots, M\}$	9
low alternative	L	$\{0, \ldots, H\}$	3

Table 6.4: Value ranges and defaults of problem generation parameters.

The value ranges and defaults of the problem generation parameters are listed in Table 6.4. Single-character symbols are associated with the parameters to arrive at concise formulas for the size of problem spaces, which we compute in the Sect. 6.6.2. In this context, it is convenient to work with the *number* (rather than the percentage) of zero entries per row. This explains why there is no symbol for the `zero utility` parameter, while there is a symbol (Z) for the number of zeroes per row. By default, `zero utility` equals -25%. In the context of the default values 12 and 5 for n and m, this corresponds with 1 zero entry in each of the uppermost six rows of the matrix, and 3 zero entries in each of the other rows. This explains the entry '1(/3)' in Table 6.4.

The `zero utility` parameter is an exception to the rule that we never vary more than one parameter simultaneously, in the sense that we use two default values for this variable. The primary default −25%, is listed in Table 6.4. Next to this primary default, which leads to a heterogemic utility matrix, we often also use the value 50% as a secondary default. Originally, we planned to use `zero utility` = 50% as the (sole) default. Later, we realized that matrices with the same number of zeroes in each row may be too peculiar to draw general conclusions, and added the option to generate heterogemic matrices.

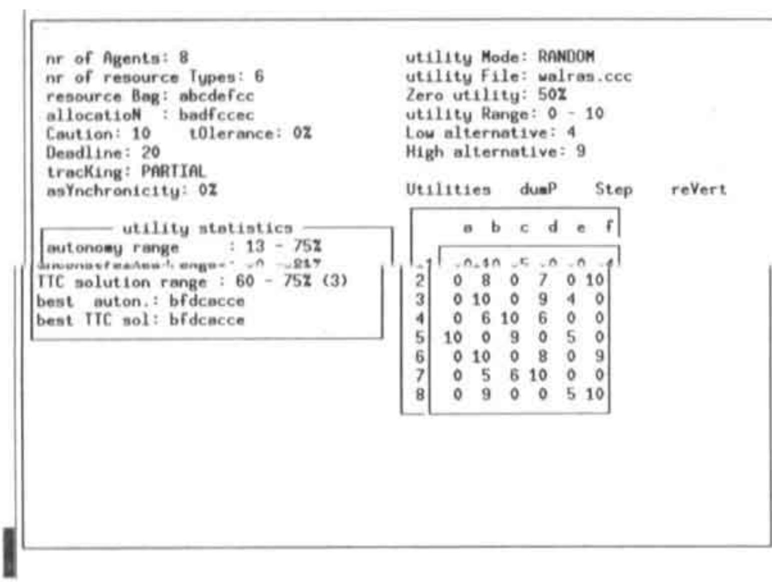


Figure 6.4: User interface of the interactive testbed.

Most of the value ranges and defaults for the parameters in the batch-oriented testbed stem from the constraints imposed by, and the experience gained with interactive simulation.

The general guideline which led to the default setting, listed in Table 6.4, has been that the typical generated problem instance should be 'average', without being too easy, since this would blur the performance differences due to different parameter settings. The latter constraint was originally checked only tentatively in the interactive testbed, but we will corroborate it in Sect. 7.3 by investigating the performance of the default IG mechanism with various sample spaces.

In the following, an expression of the form $S(\alpha, M, Z, L, H)$ denotes the sample space associated with a specific setting of the problem generation parameters (to α, M, Z, L, and H). Here, α represents the multiplicity type of the community tool bag Γ.

We use α rather than Γ to express that a problem instance is an equivalence class of problem representations under alphabetic variation (i.e., agent renumbering and/or tool-type renaming).[9] Two problem representations with different tool bags may well be equivalent, but two representations with different multiplicity types always denote essentially different problem instances.

Expressions like $S(\alpha, M)$ and $S(\alpha, L, H)$ denote problem spaces that are the union of the sample spaces with fixed values for the listed parameters, and variable values for those that are omitted. Formally,

$$S(\alpha, M) \triangleq \bigcup_{Z,L,H} S(\alpha, M, Z, L, H) \tag{6.9}$$

and

$$S(\alpha, L, H) \triangleq \bigcup_{M,Z} S(\alpha, M, Z, L, H) \tag{6.10}$$

6.5.4 Confidence intervals

The outcomes of our experiments are presented in the form of sample averages of performance attributes. These are pictured graphically. To indicate to what extent the observable differences between sample averages are statistically significant, we plot confidence intervals around the data points. The endpoints of these intervals are computed in the following manner.

We have gathered statistics by means of grouped sampling. As pointed out in Sect 6.3, this entails that the estimates for the mean value of the attribute and for its confidence limits are not computed directly from the obtained measurements $\{X_k\}_k$, but from group averages $Y_i \triangleq 1/G \cdot \sum_j X_{ij}$. This has little effect on the ultimate estimate, but the sample variance in $\{Y_i\}_i$ is much lower than that in $\{X_{ij}\}_{ij}$. Since the size of the confidence intervals depends linearly on the standard deviation of the sample, grouped sampling leads to smaller confidence intervals. In fact, grouped sampling is not merely convenient, but *required* to compute reliable confidence limits in cases where the distribution of X_k is not known to be normal. No matter what this distribution is like, we can be sure that the distribution of Y_i is approximately normal, if the group size G is sufficiently large.

In most of the experiments, the sample consists of 200 groups of 100 measurements each. Provided that the group size of 100 is sufficient for Y_i to be approximately normally distributed, we can compute confidence limits by making use of the fact that, if the distribution of Y_i is $\mathcal{N}(\mu, \sigma)$, and $\hat{\mu}$ and $\hat{\sigma}^2$ denote the sample mean and the sample variance, then the distribution of $Z_i \triangleq \sqrt{200} \cdot \frac{Y_i - \hat{\mu}}{\hat{\sigma}}$ resembles the Student t distribution with 199 degrees of freedom. We have used the Student t

[9] Examples of alphabetic variants are provided in Sect. 6.6.

table in (Kreyszig, 1970, p. 454) to compute confidence limits with a confidence level of 95%.

The crucial assumption in our method is that a group size of 100 is indeed sufficiently high for the distribution of Y_i to be approximately normal, and identical for different i. It is suggested in (Kreyszig, 1970, p. 192), that the smallest allowable group size is 30, if the distribution of X_k is "not too skew". Unfortunately, the skewness in the distributions of our performance attributes varies significantly with the *type* of attribute.

In general, the employed group size of 100 is sufficiently high to ensure that skewness is not a problem for the cardinal and boolean performance attributes. Only the auction duration (i.e., the `rounds` attribute in Table 6.2 on page 250) can be problematic in this respect. This only occurs, however, if negotiation weariness is not incorporated, while the deadline is set to a high value. In this case, auction duration is relatively low (in comparison with the deadline) for most auctions, but in some cases the deadline is reached. This amounts to a skew distribution of the number of required rounds.

The distribution of X_k for a *boolean* attribute is skew, but the distribution of Y_k is the binomial distribution $B(100, p)$, which closely resembles the normal distribution $\mathcal{N}(100p, 100p(1-p))$. Thus, while the confidence intervals are generally larger for boolean attributes than for cardinal ones, the confidence limits are reliable.

The conditional attributes are the most problematic ones. This is not so much a matter of skewness of the attributes' distributions, but a consequence of the fact that conditional attributes are not always defined. Because we treat them in the same manner as the other attributes, this implies that the *effective* group size for a conditional attribute can vary, and is often well below 100.

As an example, consider the `weariness concession` attribute. This attribute is the average of the highest utility concession due to weariness that takes place during an auction, where the average is computed over those auctions in which at least one such concession occurs. In many cases, however, weariness does not influence the behavior of agents. If the deadline is high, a phase will often end due to bid convergence in an early stage, before the weariness constraint is ever invoked. If this happens, no value is assigned to `weariness concession`. The boolean `zero weariness` attribute describes how often this has occurred. Hence, if the outcome of a simulation involves a high `zero weariness` — of, say, 90% — the computed value (and the confidence limits) of `weariness concession` is not very reliable, because the effective group size employed in this computation is not a fixed value, but a random one, which is binomially distributed, with an average of 10 instead of 100.

While it is principally possible to adapt the testbed in a way that ensures the reliability of the conditional attributes under all circumstances, we have decided not to do this, because the unreliability is confined to those cases where the attributes have little significance anyway. More elaborate justification for this decision is provided in the next section.

6.6 Methodological Justification

In this section, we motivate some of the decisions mentioned in the previous section (such as drawing sample problems from specific *subspaces* of the problem space), and verify the assumptions underlying our statistical methods. Knowledge of the contents of this section is not required to understand the presentation and analysis of experimental outcomes, so readers primarily interested in the outcomes of our experiments should feel free to skip this section, and proceed with the next chapter (at page 269).

6.6.1 The importance of sample-space selection

Alternative algorithms for a problem domain can be compared by gathering performance statistics on a set of sample problems. It is usually easier to generate sample problems by means of a computer than to collect a representative set of real-life problems. This is certainly the case in the reassignment domain. However, when problem instances are generated by a computer program, the outcome of performance evaluation may depend heavily on the particular sample space that is chosen. This has been observed in literature concerning various problem domains, such as game tree search (Plaat, 1996, pp. 81,109), constraint satisfaction (Williams & Hogg, 1992), and the theory of cooperative games (Derks & Kuipers, 1996, pp. 11-14). There are two criteria which should be met by a sample space in such domains.

- The generated problems should be realistic.
- The generated problems should not be too easy.

In most domains, not every randomly generated problem representation constitutes a realistic problem. In some domains, randomly generated problems are *seldom* realistic, if the sample space is not chosen carefully. The domain of game tree search is a good example in this respect. In (Plaat, 1996, p. 81), four properties are identified which a game tree must have so as to constitute a realistic problem. One of these is that the game 'tree' should not be a tree.[10]

In the case of game tree search, it is relatively easy to identify properties that are necessary and (approximately) sufficient for a problem representation to be qualified as realistic, because the associated real-life domain (viz., games such as chess,

[10] In board games such as chess and checkers, most game states can be arrived at via different paths.

checkers, four in a row, ...) is relatively sharply demarcated, and formal in nature. For tool reassignment problems, the identification of such properties is much more difficult. The only such property that is immediately apparent is the one we have already discussed: A TR problem instance must be *viable*).

In retrospect, there is one aspect of our problem-generation method that can be qualified as a shortcoming as far as realism is concerned: In the samples generated by the IGRT, the correlation between preferences of different agents is zero on average. In real life, there tends to be at least some positive correlation between preferences of different persons. It is not feasible to say anything definite about the amount of correlation that should be present in a sample to qualify it as realistic, but a problem-generation method where the average correlation between preferences is one of the problem-generation parameters would seem preferable.

We believe that we did take sufficient measures to ensure that the other criterion (that the generated problems should not be too easy) is met. This criterion is important, because the average performance of IG will be high on a sample with many easy problems, no matter which parameter setting we use. In other words, a high proportion of trivial problems will blur the effects of agent and mechanism parameters on IG's performance.

Our primary performance measure is the allocation effectiveness, the portion of the optimal community utility[11] that is attained by the mechanism. If IG obtains a low effectiveness on some problem, this indicates that the problem is difficult *for IG*. In contrast, a low optimal community utility implies that the problem is difficult *in general*, in the sense that allocations with a high average agent satisfaction do not exist.

The above criterion that sample problems should not be too easy refers to an intermediate form of problem difficulty: The generated sample problems should not be too easy for algorithms *like* IG, that is, for (strongly) *informationally decentralized* mechanisms. An essential characteristic of such algorithms is that it is difficult — and may often not be possible at all — for the agents and the coordination module to avoid suboptimal decisions. Hence, it is reasonable to expect that the *impact* of such unavoidable mistakes is the main factor in the challenge posed by a problem instance. The generation parameters that are most influential with respect to this impact are `zero utility`, `low alternative` and `high alternative`.

In problems generated with a low `zero utility` and a high `high alternative` setting, an agent generally has an abundance of good alternatives to choose from. Hence, the average utility loss induced by a suboptimal decision is relatively low. The same is true for problems generated with a low `zero utility` setting, or with a high `zero utility`, but low `high alternative` setting. In this case, there is little

[11] cf. Def. 2.35 on page 42

hope (for most agents) to attain a significant improvement of their utility anyway.

Computation of confidence limits

When computing confidence limits for the measured averages of the performance attributes, we treat all attributes in the same manner. As a consequence, the confidence limits associated with conditional attributes are not always reliable.

To ensure that *all* computed confidence limits are reliable, we would have to perform additional simulations to obtain sufficient data on the conditional attributes, such as the `weariness concession` attribute. These simulations would be relatively costly, and especially so when their outcomes are least relevant: in sample spaces with very high zero-weariness, that is, if weariness rarely plays any role. Such simulations are not worth the effort.

We shall not issue an explicit warning for every graph that features some unreliable confidence intervals, because such cases of unsubstantiable computation tend to stick out clearly: the confidence intervals are either extremely large, or adjacent intervals vary erratically in size. As such, plotting the intervals, even when the computed confidence limits are largely devoid of meaning, does serve some purpose. It indicates graphically that the associated data points are not very reliable.

6.6.2 The size of the sample spaces

The procedure we use to compute confidence limits hinges on the assumption that the sample size is negligible with respect to the size of the sample space. To validate this assumption, we determine lower bounds on the sizes of the two most frequently used sample spaces: the default space (which features heterogemic utility matrices), and the default homogemic sample space.

Unfortunately, we cannot determine the size of these sample spaces by means of a simple combinatorialization. This is due to the fact that we represent a reassignment problem instance as an ordered pair $\langle e, U\rangle$ of an initial endowment e and a utility matrix U. This representation is natural and clear-cut, but it possesses many symmetries. In other words, different representations often correspond with one and the same problem. Hence, if we would estimate the size of a sample space by counting the number of different representations that are consistent with some problem generation parameter setting, the resulting estimate is an upper bound of the actual size. Since we need a lower bound, we must correct for the representation symmetries.

As an example of representation symmetry, consider the problem instances $\langle e, U_1\rangle$ and $\langle e, U_2\rangle$ where e is the assignment $[aabbcde]$, and the matrices U_1 and U_2 are defined as

$$U_1 = \begin{pmatrix} 6 & 10 & 0 & 8 & 3 \\ 0 & 7 & 10 & 0 & 4 \\ 1 & 4 & 6 & 10 & 0 \\ 3 & 5 & 0 & 3 & 10 \\ 10 & 4 & 3 & 0 & 7 \\ 10 & 6 & 9 & 8 & 0 \\ 0 & 10 & 3 & 4 & 6 \end{pmatrix} \qquad U_2 = \begin{pmatrix} 4 & 1 & 10 & 0 & 6 \\ 5 & 3 & 3 & 10 & 0 \\ 10 & 6 & 8 & 3 & 0 \\ 7 & 0 & 0 & 4 & 10 \\ 6 & 10 & 8 & 0 & 9 \\ 10 & 0 & 4 & 6 & 3 \\ 4 & 10 & 0 & 7 & 3 \end{pmatrix} \qquad (6.11)$$

The matrices U_1 and U_2 differ considerably, but $\langle e, U_1\rangle$ and $\langle e, U_2\rangle$ represent the same reassignment problem. Two kinds of representation symmetry have been used to transform $\langle e, U_1\rangle$ into $\langle e, U_2\rangle$.

tool type renaming A reassignment problem does not change if we rename the tool types. In the above example, we applied the renaming $abcde \to badec$, thus transforming $\langle e, U_1\rangle$ into the intermediary representation $\langle e', U_3\rangle$ (not shown in (6.11)). Here, $e' = [bbaadec]$, and U_3 is the result of permuting the columns of U_1 accordingly.

agent renaming A reassignment problem does not change if we rename the agents, that is apply some permutation $\pi(\cdot)$ to the initial assignment as well as to the rows of the utility matrix. In the above example, we used the permutation that transforms e' back to e. Applying this permutation to the rows of U_3 renders U_2.

While it is fairly easy to get rid of *one* of the symmetries (by employing a suitable canonical form), it appears to be difficult to get rid of both. Hence, to obtain a lower bound on the size of a problem space, we count the number of different representations in the problem space, and correct for representation symmetry by dividing this number by the maximal number of alphabetic variants any single problem instance can have.

As in Sect. 6.5.3, the expression $S(\alpha, M, Z, L, H)$ denotes the problem space associated with a specific setting of the problem generation parameters (to α, M, Z, L, and H), where α represents the multiplicity type of the community tool bag Γ, and M, Z, L, and H denote the values of the other problem-generation parameters `max. utility`, `nr of zeroes/row`, `high alternative` and `low alternative`. Expressions like $S(\alpha, M)$ and $S(\alpha, L, H)$ denote problem spaces that are the union of the sample spaces with fixed values for the listed parameters, and variable values for those that are omitted.

Sample spaces associated with different settings of α, M, Z, L, and H in the IGRT need not be disjoint. In particular, for any fixed multiplicity type α, the sample space $S^*(\alpha) \triangleq S(\alpha, 99, 0, 0, 98)$ contains all of the sample spaces $S(\alpha, M, Z, L, H)$.[12] This would seem to imply that, to get an impression of the average performance of IG, we should take our samples from the sample spaces $S(\alpha)$. However, it is doubtful whether a random sample from such a space is representative for reassignment problems that occur in practice. As we have remarked earlier, and will corroborate in Sect. 7.3, the number of reasonably high entries per row of the utility matrix is a good measure of problem difficulty (for IG as well as for other informationally decentralized mechanisms). In this respect, the average performance of an algorithm on problem instances of $S(\alpha)$ is hardly interesting, because it is very easy to obtain a solution of high effectiveness on the majority of such problem instances. In other words, the average problem difficulty in $S(\alpha)$ is generally (for most, and especially for high values of m) too low. In our experimentation, we have therefore chosen to take samples from problem spaces where at least 50% of the agents have few good alternatives for their most-preferred tool type. We either use homogemic matrices with 2 alternatives per agent (with a normalized utility between 30% and 90%), or heterogemic matrices where half of the agent population has only one such alternative (while the other half has $m-2$ alternatives).

Since the only thing we need to know about the sample spaces is that they are much larger than the samples drawn from these spaces, it suffices to compute a lower bound on the sample space size. Such a lower bound is provided by Prop. 6.2 for problems with homogemic utility matrices, and by Prop. 6.3 for heterogemic ones.

Proposition 6.2
Let $\alpha = \xi(\Gamma)$ denote the multiplicity type of a tool bag Γ of size n, and carrier size m. Let $M > 1$, $0 \leq Z \leq m-1$, and $0 < L \leq H < M$. Then a lower bound for the size of the IGRT sample space $S(\alpha, M, Z, L, H)$ of homogemic reassignment problems is

$$\frac{\left(\binom{m-1}{1} \cdot \binom{m-1}{Z} \cdot (H-L+1)^{m-Z-1}\right)^n}{\left(\prod_{x\in\alpha} x!\right) \cdot \left(\prod_{x\in\xi(\alpha)} x!\right)} \tag{6.12}$$

Proof.
One of the consequences of problem invariance under agent renaming is that, to compute the size of the space of viable problems $\langle e, U\rangle$ for some setting of the problem generation parameters Γ, M, Z, L, and H, it suffices to count the number of problems $\langle e^*, U\rangle$, where e^* is some *fixed* initial assignment. Hence, the size of

[12] This is a white lie. It would be true if the entries of the utility matrix were real numbers. Since they are integers, we should state that $S^*(\alpha, M)$ contains all problem spaces $S(\alpha, M', Z, L, H)$ with $M \bmod M' = 0$.

$S(\alpha, M, Z, L, H)$ equals the number of different, viable problems $\langle e^*, U\rangle$ that can be formed by combining a utility matrix compatible with M, Z, L, and H with some *fixed* endowment e^*, which is compatible with α. To fill in a row of the utility matrix we must pick one column for the agent's first preference out of $m-1$ columns.[13] From the remaining $m-1$ columns, we choose Z zero entries, and for each of the other $m-Z-1$ columns, we can choose any value k such that $L \leq k \leq H$. Hence the number K of different matrix rows that ensure viability and comply with the constraints associated with m, Z, L, and H equals

$$K = \binom{m-1}{1} \cdot \binom{m-1}{Z} \cdot (H-L+1)^{m-Z-1} \tag{6.13}$$

The number of different utility matrices that can be formed from such rows equals K^n, but some of the K^n problem representations $\langle e^*, U\rangle$ are alphabetic variants. If, for example, $e^* = [aaabbcd]$, and U' equals U, except that the first three rows occur in a different order, then $\langle e^*, U\rangle \sim \langle e^*, U'\rangle$ where $\sim$ denotes equivalence due to agent renaming. In general, we can ensure that none of these equivalence classes are counted more than once, if we divide K^n by

$$\prod_{x \in \alpha} x!$$

However, this division is not sufficient to correct for *all* representation symmetries. In combination with the fixation of the endowment, it corrects for alphabetic variants due to agent renaming, but two-tiered combinations of tool type and agent renaming, such as the two matrices in Eq. 6.11 are not dealt with yet. To deal with this kind of symmetry also, we observe that — grace to the fact that we have fixated the endowment — any tool type renaming that is involved in such representation symmetry is confined to tool types that have the same multiplicity in the community tool bag Γ. In the context of $\alpha = \xi(\Gamma)$, the number of such two-tiered alphabetic variants equals at most

$$\prod_{x \in \xi(\alpha)} x!$$

Performing the two above corrections leads to (6.12) as a lower bound for the size of $S(\alpha, M, Z, L, H)$. Even in extreme cases like $S(\alpha, M, m-1, L, H)$ or $S(\alpha, M, Z, L, L)$, the denominator of (6.12) is very small in comparison with the numerator. Since the replacement of the denominator by 1 renders an upper bound, this implies that the lower bound is reasonably sharp.

■

Below, we use the expression $\tilde{S}(\alpha, M, Z, L, H)$ to denote the sample space of heterogemic reassignment problems with Z zero entries per row in the uppermost $n/2$

[13]The column corresponding with the agent's endowment is forbidden, to ensure viability.

rows (where the symbol '/' denotes *integer* division), and $m-1-Z$ zero entries per row in the other rows,

Proposition 6.3
Let $\alpha = \xi(\Gamma)$ denote the multiplicity type of a tool bag Γ of size n, and carrier size m. Let $M > 1$, $0 \leq Z \leq m-1$, and $0 < L \leq H < M$. Then a lower bound for the size of the IGRT sample space $\tilde{S}(\alpha, M, Z, L, H)$ of heterogemic reassignment problems is

$$\frac{K_1^{n/2} \cdot K_2^{n-n/2}}{\left(\prod_{x \in \alpha} x!\right) \cdot \left(\prod_{x \in \xi(\alpha)} x!\right)} \tag{6.14}$$

where

$$K_1 = \binom{m-1}{1} \cdot \binom{m-1}{Z} \cdot (H-L+1)^{m-1-Z}$$

and

$$K_2 = \binom{m-1}{1} \cdot \binom{m-1}{m-1-Z} \cdot (H-L+1)^{Z}$$

Proof.
Eq. 6.14 differs from (6.12) in Prop. 6.2 in that there are now two factors in the numerator. The first factor, $K_1^{n/2}$, denotes the number of different upper halves of U that are consistent with the problem generation constraints (Z zero entries, and $m-1-Z$ alternatives), while the second factor, $K_2^{n-n/2}$, pertains to the lower half of U (where each row has $m-1-Z$ zero entries, and Z alternatives). Since $L > 0$, the matrix is truly heterogemic in the sense that the number of zero entries per row is different in the two halves of U. Consequently, no row in the upper half equals any row in the lower half. Hence, the fraction of alphabetic variants among the representations counted in the numerator is lower than in the corresponding homogemic sample space $S(\alpha, M, Z, L, H)$. Because the denominator in (6.14) equals that in (6.12), it follows that (6.14) is a lower bound on the size of $\tilde{S}(\alpha, M, Z, L, H)$.

■

Example 6.4 (Some sample space sizes)
To get an impression of the sizes of the sample spaces which we used in the experiments, we compute the lower bounds on the size of the default space for homogemic matrices, $S(\alpha, 10, 2, 3, 9)$, with $\alpha = \{3, 3, 2, 2, 2\}$, the associated enveloping space

$S^*(\alpha) \triangleq S(\alpha, 99, 0, 0, 98)$, and the heterogemic default space $\tilde{S}(\alpha, 10, 1, 3, 9)$. Applying Prop. 6.2, we get, as a lower bound on the size of the default homogemic problem space:

$$\frac{\left(\binom{4}{1}\cdot\binom{4}{2}\cdot 7^2\right)^{12}}{\left(\prod_{x\in\{3,3,2,2,2\}} x!\right)\cdot\left(\prod_{x\in\{2,3\}} x!\right)} = \frac{(4\cdot 6\cdot 49)^{12}}{(3!)^3\cdot(2!)^4} \approx \frac{7\cdot 10^{36}}{3456} \approx 2\cdot 10^{33}$$

Analogously, we obtain, for the size of the enveloping space $S^*(\alpha)$:

$$\frac{\left(\binom{4}{1}\cdot\binom{4}{0}\cdot 99^4\right)^{12}}{3456} \approx \frac{(4\cdot 1\cdot 9.6\cdot 10^7)^{12}}{3456} \approx \frac{10^{93}}{3456} \approx 3\cdot 10^{89}$$

Finally, Prop. 6.3 renders a lower bound on the size of the heterogemic default space $\tilde{S}(\alpha, 10, 1, 3, 9)$, which was used in the majority of the performed experiments. Substituting the appropriate values, we get

$$\begin{cases} K_1 = \binom{4}{1}\cdot\binom{4}{1}\cdot 7^3 = 5488 \\ K_2 = \binom{4}{1}\cdot\binom{4}{3}\cdot 7 = 112 \end{cases}$$

and hence,

$$|\tilde{S}(\alpha, 10, 1, 3, 9)| \geq \frac{5488^6\cdot 112^6}{3456} \approx 1.5\cdot 10^{31}$$

△

Statistical methods usually assume that the size of a sample is negligible in comparison with that of the whole population, and that the sample does not contain (m)any replicas. These assumptions also underlie our computation of confidence limits. If a sample contains most of the problem instances in the problem space, the computed confidence limits will be overly pessimistic. If most of the instances in a sample are the same, while the sample covers little of the problem space, computed confidence limits will be overly optimistic. Hence, we need to make sure that neither is the case. Ideally, we should ensure that all of the instances in the sample are different, while the size of the sample is negligible in comparison with the size of the problem space. However, the sample problems generated by the IGRT are drawn independently with repetition from the *representation* space. Hence, we cannot *preclude* that the same problem representation is generated more than once. Moreover, it is possible that different problem representations constitute alphabetic variants of the same problem. The analysis below shows, however, that this is unlikely to happen very often.

The outcome of the computations in Ex. 6.4 reveals that we need not worry about sample sizes approaching or exceeding the size of the sample space, if we use our (heterogemic or homogemic) default sample spaces. The size of the samples which we used to obtain performance statistics vary between 100,000 and 200,000, while the smallest default sample space $\tilde{S}(\alpha, 10, 1, 3, 9)$ contains more than 10^{31} different problem instances. Hence, computed confidence limits will not be overly pessimistic. To show that they are not overly optimistic either, we need to examine the probability that a sample contains multiple occurrences of the same problem representation, or different representations of the same problem. If we draw a random sample of 200,000 problem instances from a sample space of 10^{31} problem instances, the probability that the sample contains more than one occurrence of *any* problem instance is negligible.[14] Because the difference between the size of the *representation* space (from which we draw the sample problems) and the actual problem space is negligible in comparison with the size of the problem space, it is not relevant that we generate problem representations rather than problems. The probability that a sample comprises more than one alphabetic variant of the same problem instance is also negligible.

For the majority of our experiments, the above assertions are less crucial than they are in most statistical analyses, for the following reasons.

Confidence intervals usually serve to express the likelihood that the computed *sample* average is close to the actual *sample space* average. While this certainly also plays some role in our experiments, the primary purpose of most graphs in this chapter is *compare* the average performances obtained with different parameter settings, or to compare the performance obtained by IG with those of other mechanisms. In this context, issues such as the negligibility of the sample size in comparison with the sample space size, and the degree to which the sample is representative for the sample space are less vital, because we use the *same* sample for all of such experiments.

Of course, this is not the case for the experiments on the influence of the problem *generation* parameters. For the reliability of these impressions, the above methodological issues are vital. Furthermore, these experiments employ other then the default sample spaces, for which we did not yet verify that the sample size is negligible with respect to the size of the sample space. Hence, we will remain aloof in this respect, when we discuss the influence of generation parameters in Sect. 7.3.

[14]We do not prove this. Although exact calculation of the probability is not feasible, one can obtain a lower bound on the complementary probability that is very close to 1, by making use of fact that the functions $f_k(x) = (1-(1/x)^k)^x$ are monotonically increasing on $[1, \infty)$, for any $k \in \mathbb{N}$.

6.6.3 Sample representativity

We claim that the samples generated by the IGRT are reasonably representative for the sample spaces from which they are drawn. The only flaw in this respect is that we draw problem instances from a uniform distribution on the *representation* space associated with a problem space. Consequently, problems with few alphabetic variants have a smaller chance to occur in a sample. However, since the number of alphabetic variants is largely determined by the problem-generation parameters (α and Z, in particular), this is not much of a problem: within a specific sample space, the number of alphabetic variants will be roughly the same for most problem instances.

To allow our readers to make their own judgements on the randomness of our problem-generation procedure, we explain how this is implemented in the IGRT.

The generation of a problem instance by the IGRT comprises two steps:

1. the generation of a random utility matrix U, compliant with the setting of the generation parameters;

2. the generation of a random initial assignment e, compliant with the tool bag, such that the problem instance $\langle e, U\rangle$ is viable.

In both steps, random permutation is a key element. To generate a homogemic utility matrix we first generate a sequence s of m agent utilities that is compatible with the settings of `utility range`, `zero utility`, `low alternative`, and `high alternative`. The first Z elements of this sequence are zero, the last element equals `utility range`, and the others are chosen independently from a uniform distribution on the range of alternatives.

We then use the following procedure to permute s randomly.

```
procedure RandomPermute(s,m) ::=
for i := 1 to m-1 do
begin
"Pick a random integer k between i and m (inclusive)"
"Exchange the elements at positions i and k in s"
end
```

The random numbers k are generated by means of the random number generator `ran1` described in (Knuth, 1997), which renders floating point numbers that are uniformly distributed over the interval $[0,1)$. By defining k as the integral part of $i + (m-i+1)\cdot ran1()$, we obtain random integer numbers which are uniformly distributed over the set $\{i, i+1, \ldots, m\}$. This implies that the distribution of permutations acquired by subsequent invocations of the procedure `RandomPermute` is also uniform (on the set of permutations involving m elements).[15]

The generation of random alternative utilities and the random permutation of the resulting matrix row is repeated (with new random numbers) for each of the n rows.

Heterogemic matrices are generated in the same manner, except that the upper and lower halves are generated separately with the appropriate values for the number of zero entries per row.

A random viable assignment is then generated in two steps. First we generate a random assignment by applying the procedure `RandomPermute` to (a copy of) the tool bag. Then we check whether the resulting problem is viable by asserting that no agent is endowed with its first preference. If we encounter an agent i that is endowed with its first preference y, we look for another agent, with an endowment and first preference different from y. If there is such an agent j, we exchange the endowments of i and j. If no suitable exchange candidate is found, we generate a new random assignment.

The number of exchanges that are necessary to turn the random assignment into a viable one can be high if there is a large discrepancy between supply (i.e., the tool bag) and demand for tool types (i.e., the bag of first preferences). Because the search for exchange candidates always starts with agent 1, the tool types which are first preferences for many agents tend to end up predominantly in the first part of the assignment (as a string of characters). Hence, the initial assignments that are ultimately rendered are not uniformly distributed over the set of viable assignments. However, since there is no bias in the order of the matrix rows, this does not imply that the generated *problems* are biased. Only the generated *representations* are.

[15]A formal proof could be given by establishing, with induction to $m = 2, 3, \ldots$ and $l = m, m-1, m-2, \ldots, 1$, that the conditional distribution of the initial positions of the elements which turn up at final position l, given the initial positions of the elements at final positions before l, are uniform.

Chapter 7

Experimental Results

7.1 Chapter Overview

In this chapter, we present and analyze the experimental findings obtained with the IGRT. For a proper understanding of this analysis, perusal of the (relevant portions of) Chapter 6 is required, with the exception of Sect. 6.6.

7.2 IG Performance with the Default Parameter Setting

For the reader's convenience, we reproduce the default parameter setting, specified earlier in Tables 6.3 and 6.4, in Table 7.1 below.

Agent parameters:	
caution	10
volatility	0%
maxloss	100%
weariness	`TRUE`

Problem generation parameters:	
nr of agents	12
nr of tool types	5
tool bag	*aaabbbccddee*
max. utility	10
zero utility	-25%
high alternative	9
low alternative	3

Mechanism parameters:	
deadline	500
clearance	`FALSE`
resolution	`MIXED`
asynchrony	50%

Table 7.1: The default values of IGRT parameters.

In the following, we sometimes use the symbols n and m for the number of agents and the number of tool types respectively. In the table, we have used the phrase "max. utility" instead of "utility range", because the `utility range` parameter actually defines the upper bound of the range only. The lower bound is determined by `zero utility` or by `low alternative`.[1]

[1]The lower bound of the utilities is zero if `zero utility` is nonzero, and equal to `low`

Recall that, in the context of 12 agents and 5 tool types, the default value for `zero utility` of −25% signifies that the utility matrix is heterogemic with 1 zero entry (25% of 5 − 1) in the uppermost 6 rows, and 3 in the lowermost ones. If not stated otherwise, the simulations are performed with grouped sampling, generating 200 groups of 100 problem instances. The statistics output by the IGRT, if all parameters are set to their default value, are pictured in Table 7.2. To facilitate discussion of some of this output, the lines are numbered.

```
STATISTICS OVER 200 x 100 AUCTIONS (with various matrices):

 1 opt. unconstrained community utility: 72.50-100.00 (avg=93.12; var=0.1206)
 2 opt. ind.rat. effectiveness: 81.82-100.00 (avg=99.64; var=0.0126)
 3 initial allocation effectiveness:  0.00-69.91 (avg=32.21; var=0.7897)

 4 final allocation effectiveness: 70.00-100.00 (avg=92.45; var=0.2761)
 5 perc. of zero-loss auctions: 65.00-86.00 (avg=76.81; var=17.8139)
 6 avg. norm. utility loss per loser: 10.00-90.00 (avg=40.91; var=19.1457)
 7 highest norm. utility loss: 10.00-90.00 (avg=41.43; var=18.6954)
 8 percentage of losers:  8.33-25.00 (avg= 8.73; var=0.1450)
 9 perc. of zero-weariness auctions: 81.00-97.00 (avg=90.08; var=8.5794)
10 max. norm. weariness concession:  0.00-12.41 (avg= 1.06; var=0.2691)

11 nr of bids: 30.00-742.00 (avg=55.06; var=13.4168)
12 nr of rounds:  2.00-67.00 (avg= 5.17; var=0.1258)
13 nr of phases:  1.00- 3.00 (avg= 1.70; var=0.0019)

   Termination of first phase:
14 perc. of bid convergence: 100.00-100.00 (avg=100.00; var=0.0000)
15 perc. of market clearance:  0.00- 0.00 (avg= 0.00; var=0.0000)
16 perc. of deadline exceeded:  0.00- 0.00 (avg= 0.00; var=0.0000)
```

Table 7.2: The statistics rendered by the IGRT in its default setting.

Each line of the table specifies the minimum, maximum, average, and variance of the data obtained on a specific problem attribute (the first three lines) or performance attribute (the other lines). The semantics of the average values are fixed, but those of the minima and maxima depend on the attribute type. For problem attributes and cardinal[2] performance attributes (lines 1-4, 11-13), the minima and maxima are the extremes of all 20,000 measurements. For the boolean attributes (lines 5, 9, 14-16), the minima and maxima pertain to the 200 group averages instead of the raw measurements.[3] Finally, the minima and maxima of the conditional attributes

alternative otherwise.

[2]For the precise meaning of "cardinal attribute", see Table 6.2 on page 250.

[3]The raw value of a boolean attribute is either 0 or 100%, so the raw extremes of boolean

are obviously restricted to those auctions in which they are defined. As such, the minimum of 8.33 for the percentage of losers tells us that, the lowest number of losers in auctions with at least some utility loss was 1 (8.33% of 12).

Some lines in Table 7.2 deserve further explanation. Lines 1-3 have no immediate bearing on IG performance. They describe sample space properties rather than algorithmic performance attributes. The values specified on these lines can be interpreted as indicators of the average problem difficulty in the chosen sample space, where "problem difficulty" has a slightly different meaning on the three lines. The optimal community utility in line 1 reflects the difficulty of obtaining a high level of average agent satisfaction. Its value is an upper bound that cannot be improved on by any algorithm.

The optimal effectiveness under individual rationality in line 2 is the percentage of optimal community utility that can be obtained (at most) if we demand solutions to be individually rational. As such, it defines a lower bound on the cost of demanding individual rationality. As line 2 shows, this minimal cost is very low if individual rationality is the *only* constraint. We shall see later that the cost tends to be much higher if we also impose strong informational decentralization, as we do in IG and in the Walrasian (exchange) auction.

Finally, the effectiveness of the initial allocation in line 3 provides a reference value for *low* algorithmic performance. These values constitute lower bounds on the performance statistics of the least sophisticated reassignment algorithm that one can think of: random selection of an assignment.[4]

Strong informational decentralization is bound to incur some deterioration of the community utility that can be obtained by an algorithm. As line 4 shows, the *average* deterioration is low with IG: it obtained more than 92% effectiveness on average in the default sample space. The worst-case effectiveness of 70% (over 20,000 auctions), however, is much lower than the average.

A disadvantage of IG in comparison with Walrasian exchange is the fact that it does not guarantee that solutions are individually rational. Agents that take part in an IG auction run the risk of losing some utility instead of gaining some. However, as apparent from line 5, the solutions rendered by IG are individually rational in roughly three out of four cases, even if we do not constrain `maxloss` at all. From the minimum, maximum and average values of `losers` in line 8, we conclude that there is seldom more than one loser, and never more than three. However, *when* the utility of an agent decreases, it tends to decrease considerably: In those (23%) of the auctions that feature utility loss, the normalized utility of the foremost victim

attributes have no information value.

[4]These lower bounds are rather *loose*, because the initial assignments in the testbed are such that no agent is completely satisfied (due to the constraint that all generated problem instances must be viable).

decreases by as much as 43% on average. This high value is mainly due to the default setting of `zero utility` to −25%. This implies that, for half of the agent population, 3 of the 5 tool types are utterly useless. In this respect, auctions are relatively risky with the default parameter setting.

From line 12, we conclude that the auction duration (in rounds) is pleasurably low in comparison with the deadline of 500 rounds per phase. The average duration in rounds is only approximately 1% of the deadline, and even the longest duration among the 20,000 auctions is quite modest (67 rounds, that is, less than 14% of the maximally allowed number of a rounds in a single phase). Apparently, negotiation weariness can constrain the duration of the auction effectively, even at a low weariness treshold level.

Indeed, we observe, from line 10, that the highest weariness concession ever done by an agent in one of the 20,000 auctions was less than 13%, and the average weariness concession in those 1984 auctions[5] where weariness played any role at all is as low as 1%. In Chapter 5, we articulated our worries that weariness would constitute a considerable deviation from rational agent behavior, when applied at a high treshold level. From the above observations, we can conclude that there is little reason to worry, if we use the default parameter setting. The 'irrationality' appears to be marginal in this case.

As for lines 11 and 12, the observant reader may have noticed a discrepancy between the minimum number of rounds of 2, and the minimum number of bids of 30. How can 2 rounds comprise 30 bids if there are only 12 agents? The cause of this anomaly lies in a minor inefficiency in the implementation. With `async` at its default value of 50%, the auctioneer processes 6 (50% of 12) proposals per round, except in the first and last rounds of a phase, which should comprise the proposals of all agents (cf. the explanation of asynchrony in Sect. 6.2.3). In a maximally efficient implementation, the auctioneer would, upon discovering that each of the 6 agents whose proposals were processed stuck to its previous proposal, wait for the other 6 agents to respond to the last market profile before checking for bid convergence. In the current implementation, the auctioneer communicates the (same) market profile again, and then waits for all 12 agents to respond. This leads to the observed total of 12 + 6 + 12 = 30 bids.

We have not bothered to eliminate this inefficiency, because the total number of submitted bids hardly plays any role in our evaluation: the auction duration is nearly always expressed in rounds.

In the default setting, lines 14-16 do not have any informative value. Since `clearance = false`, the IGRT does not check for market clearance in this case, and the setting `weariness = true` *ensures* that the deadline condition is never

[5]The number 1984 stems from line 9, which tells us that wariness played a role in 9.92% of the 20,000 auctions.

invoked either.

7.3 Influences of Problem-Generation Parameters

The setting of the problem-generation parameters determines the sample space from which the IGRT draws its problem instances. Investigation of the influences of these parameters serves three purposes.

1. corroboration of our hypothesis (based on intuition and interactive simulation) that the default (heterogemic and homogemic) sample spaces are adequate for the investigation of the influences of the other (agent and mechanism) parameters, in the sense that problem instances drawn from these spaces are not too easy;

2. determination where the really hard problems lie, in terms of the values of the problem-generation parameters;

3. evaluation of the adequacy of the chosen set of problem-generation parameters in this respect: To what extent are they capable of singling out the hard problems?

As noted earlier, we do not vary all parameters simultaneously. For the problem-generation parameters, the experimentation scheme is as follows.

1. variation of `zero utility`, `low alternative` and `high alternative`

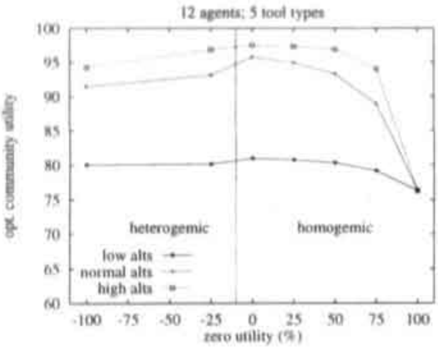
2. variation of the multiplicity type of the tool bag

3. variation of the number of agents and the number of tool types

In all of these cases, we picture the influence of the parameters on IG performance against the background of their influence on 'general problem difficulty'. This is done in terms of the IGRT sample-space attributes "optimal community utility" and "initial effectiveness". The first of these attributes indicates to what extent it is feasible to satisfy all agents, while the second is a measure of the average effectiveness attainable by means of random reallocation. Since the initial effectiveness is computed by averaging over *viable* allocations (i.e., cases in which *no* agent is endowed with its first preference), this measure is, in fact, a very loose lower bound of what random reallocation will attain on average. Hence, it has little value as a reference value to evaluate mechanism performance in an absolute sense. Like the optimal community utility, however, it does provide information on the *relative* difficulty of sample spaces.

7.3.1 Zero utility, low alternative, and high alternative

The `zero utility` parameter specifies the percentage of tool alternatives (i.e., tool types other than the agent's first preference), that are utterly useless for an agent. Obviously, differences in `zero utility` are meaningful only if the values are sufficiently wide apart. Since the number of tool types in the testbed never exceeds 20, the `zero utility` values of 0% and 1% are always equivalent. In the default context of 5 tool types, there are 7 different (equivalence classes of) `zero utility` settings. The values 0%, 25%, 50%, 75% and 100% give rise to homogemic utility matrices with 0, 1, 2, 3, and 4 zero elements per row respectively. The negative settings −100% and −25% render heterogemic matrices with 0 or 4 and 1 or 3 zero elements respectively.



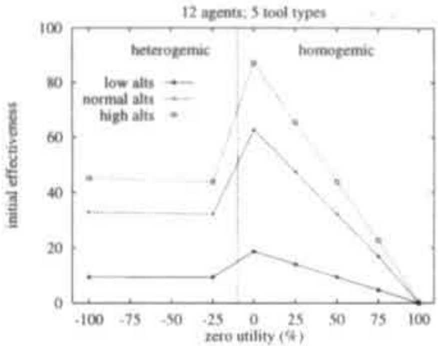


Figure 7.1: Influence of `zero utility` on problem difficulty.

Before we go into the effects of `zero utility` variation on IG performance, we discuss its effects on the sample-space attributes "optimal community utility" and "initial effectiveness".

The graphs in Fig. 7.1 portray these effects for three different settings of the range of alternative utilities. The low-alts curves correspond with the setting `low alternative` = 1 and `high alternative` = 2. The normal-alts curves correspond with the default range (alternative utilities between 3 and 9 inclusive), and the high-alts curves with alternative utilities between 8 and 9 inclusive.

The data points in the graphs are connected by lines to help to reader to distinguish between the three settings for the range of alternatives. Please note that these connecting lines have *no meaning* other than this, since there are no intermediate values of `zero utility` between any two data points. Also, the ordering of the data points along the horizontal axis is partly arbitrary. It reflects a sample space ordering in the homogemic part of the graphs *only*. The problem instances associated with the data points at `zero utility` = −25 are, in fact, closer (i.e., more similar) to the instances associated with **zero utility** = 25 than to those with **zero utility** = 0. Hence, it would be nonsensical to speak (or think) of "a steep increase of initial effectiveness in the **zero utility** interval between −25 and 0". To stress this, we have drawn a vertical line that separates the homogemic and heterogemic parts of the graphs. We discuss these two parts separately.

Confidence intervals were originally plotted around each data point, but turned out to be so small that they were hardly visible. We left them out to be able to use point labels.[6]

The homogemic parts of the graphs hold no real surprises. The average agent satisfaction decreases with increasing **zero utility** in both cases. This is understandable. An algorithm that searches for the best assignment (in terms of average agent utility) tends to avoid the zero entries of the utility matrix. This, however, becomes progressively more difficult if the percentage of zeroes increases. As for the linearity of the decrease in the second graph, the average community utility obtained with a random, viable initial allocation will decrease linearly with increasing **zero utility**. And because the variation (due to **zero utility**) in the average community utility of initial allocations is much greater than the corresponding variation in optimal community utility, the average effectiveness of the initial assignment, which approximately equals the fraction of these two, is approximately linear in **zero utility** also.

That the three community utility curves have a common endpoint is not surprising either. At **zero utility = 100%**, the alternative-range setting does not have any influence on the optimal community utility, since there are no tool alternatives. All entries of the utility matrix other than those of maximal utility are zero.

As for the heterogemic parts of the graphs, we observe that the heterogemic settings of **zero utility = -k%**, which are a mixture of the settings of $k\%$ and $100 - k\%$, are also inbetween settings in terms of optimal community utility and initial effectiveness. It does seem, however, that the matrix half with many zero entries is dominant in this respect.

Fig. 7.2 depicts the influence of **zero utility** on IG performance, for the three different ranges of alternative utilities. As in Fig. 7.1, we removed the (almost invisible) confidence intervals of the effectiveness graph to get a proper point-label legenda. This legenda also applies to the other graphs, that is, the data point marking is the same in all graphs: The points of the high-alts curves are marked with squares, and those of the normal-alts curves with plussigns. The low-alt curves are easier to recognize from the solid connecting lines then from their (diamond-shaped) point markers.

[6] The gnuplot package cannot handle point labels in combination with confidence intervals in graphs with multiple curves.

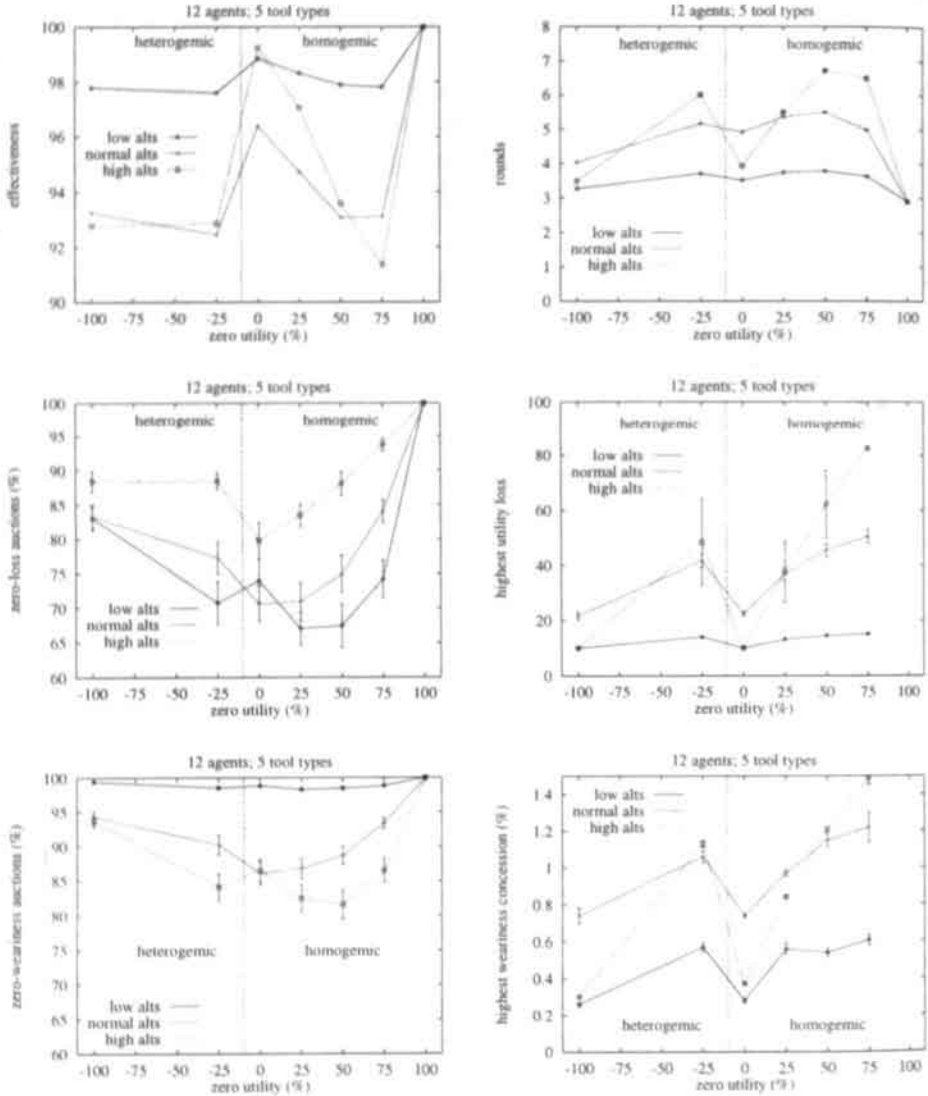


Figure 7.2: Influence of `zero utility` on IG performance.

Note that the confidence intervals (and hence the data points) of the curves in the highest-loss graph are unreliable for those data points which correspond with a high percentage of zero-loss auctions (cf. Sect. 6.5.4). This is the case for the data points of the low-alts curve in particular.

We make the following general observations concerning the graphs in Fig. 7.2.

1. The impact of `low alternative` and `high alternative` on the *effects* of varying `zero utility` appears to be largely a matter of intensification. In most graphs, the three curves have roughly the same form, but the effects of varying `zero utility` are most pronounced in the high-alts curve, and least so in the low-alts curve.

2. IG performance at the default settings (represented by the data points of the normal-alts curve at `zero utility` values of −25 and 50%) appears to be relatively low in comparison with the other settings. This indicates that the default heterogemic and homogemic sample spaces are sufficiently challenging to be used in subsequent experiments.

3. The ordinates of the heterogemic data points in Fig. 7.2 lie between those of the associated homogemic value pairs, while the sparse half of the utility matrix seems to dominate the effects. In other words, the influence of `zero utility` in the heterogemic case seems to be approximately the same as in the homogemic case, if we match the two cases according to "the highest number of zero entries in any row". This was confirmed by another experiment (pictured in Fig. 7.3), involving problems with 20 agents and 9 tool types.

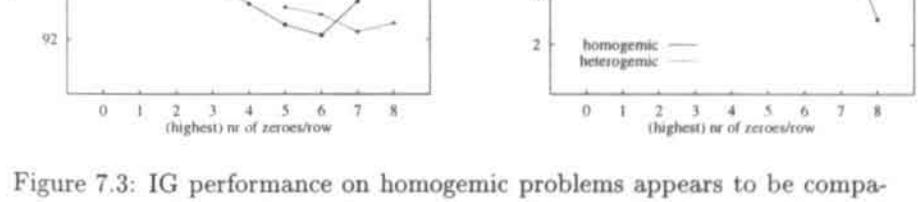


Figure 7.3: IG performance on homogemic problems appears to be comparable to that on heterogemic problems with the same (highest) number of zero entries per row.

With respect to observations pertaining to individual graphs, we confine ourselves to the primary performance attributes (i.e., allocation effectiveness and auction duration).

The effectiveness graph reveals that, with the exception of 100% `zero utility`, problems with many useless tool types per agent are generally more difficult for IG than those with few or no useless tool types. This is consistent with our hypothesis (cf. Sect. 6.6.1) that the performance of IG depends mainly on the impact of the errors which the agents are bound to make.

As for the exception at 100% `zero utility`, it is not surprising that IG obtains 100% effectiveness on such problem instances: there are no useful alternatives for the agents' preferred tool type, so every agent will stick to its original proposal in the second round. Execution of the pseudo-composition protocol will then render either maximal utility to an agent, or zero utility, and the community utility that is obtained depends only on the percentage of agents that obtain maximal utility. Since the protocol always assigns as many first preferences as possible, this percentage is maximal. In other words, IG attains 100% effectiveness.

The number of rounds initially *increases* with `zero utility`, attains a maximum at `zero utility = 50%`, and decreases again to a minimum of 3 rounds at 100% `zero utility`. An explanation for the minimum at 100% was provided in the previous paragraph. The observation that a moderate value of 50% tends to lead to a higher auction duration than the more extreme values of 25% and 75% is much more difficult to explain. All our attempts to do so led to explanations involving multiple competing influences which defy quantitative comparison, and are therefore highly conjectural.

Comparison of the zero-weariness curves and the rounds curves shows that, as one would expect, weariness is more often applied if the auction takes more time. The weariness concessions appear to be low even for high values of `zero utility`. Note that the curves show the *average* over the sample sets of the highest utility concession that was made by any agent in the auction. Whereas we usually confine ourselves to the sample set averages, we make an exception here, and also show some worst cases.[7]

zero utility	worst-case concession
-100%	10.00%
-25%	12.86%
0%	10.00%
25%	12.41%
50%	15.15%
75%	15.87%

Table 7.3: Worst-case weariness concessions.

The worst-case concessions observed with the default setting of the range for alternative utilities ('normal alts') are listed in Table 7.3. We conclude that the worst-case

[7] One of the reasons to make an exception is the relative unreliability of the estimates produced for conditional attributes, of which the weariness concession is one.

concessions for varying **zero utility** are never much larger than that of the default (−25%) setting. As such, the conclusion that we need not be overly worried by the onset of irrational behavior due to weariness (which we drew in Sect. 7.2 with respect to the default sample space) remains valid if we vary **zero utility**.

7.3.2 The multiplicity type of the tool bag

symbol	tool bag	multiplicity type	eccentricity	inhomogeneity
m0	aaabbbccddee	$\{3,3,2,2,2\}$	1	0.03
m1	aaabbbcccdde	$\{3,3,3,2,1\}$	2	0.08
m2	aaaabbccddee	$\{4,2,2,2,2\}$	2	0.08
m3	aaaabbbccdde	$\{4,3,2,2,1\}$	3	0.13
m4	aaaabbbcccde	$\{4,3,3,1,1\}$	3	0.18
m5	aaaabbbbccde	$\{4,4,2,1,1\}$	3	0.23
m6	aaaaabbccdde	$\{5,2,2,2,1\}$	4	0.23
m7	aaaaabbbccde	$\{5,3,2,1,1\}$	4	0.29
m8	aaaaabbbbcde	$\{5,4,1,1,1\}$	4	0.39
m9	aaaaaabbccde	$\{6,2,2,1,1\}$	5	0.44
m10	aaaaaabbbcde	$\{6,3,1,1,1\}$	5	0.49
m11	aaaaaaabbcde	$\{7,2,1,1,1\}$	6	0.69
m12	aaaaaaaabcde	$\{8,1,1,1,1\}$	7	1.00

Table 7.4: The multiplicity types for $n = 12$ and $m = 5$.

In Table 7.4, we have listed the 13 possible multiplicity types of the tool bag if the parameters **agents** and **tool types** are set to their default value of 12 and 5 respectively. The effects of using a multiplicity type different from the default one (m0) on IG performance is shown in Fig. 7.4.

Fig. 7.4 reveals that tool bag eccentricity has little impact on the duration of the auction, but the allocation effectiveness decreases considerably as the eccentricity of the tool bag increases. On the other hand, IG appears to perform better on problems with (very) eccentric tool bags in terms of the frequency of utility loss. A plausible explanation for these two performance effects is the following.

In the problem sample, the first preferences of an agent are evenly distributed over the tool types. Because there are 8 type-a tools and only one of each other tool type, type-a tools will nearly always be strongly oversupplied, while the other tool types tend to be undersupplied. Hence, agents endowed with a type-a tool will generally be less eligible than the other four agents. At first sight, one would expect these a-possessors to withdraw from the auction upon becoming aware of their low eligibility, so as to avoid utility loss. However, interactive simulation revealed that even a-

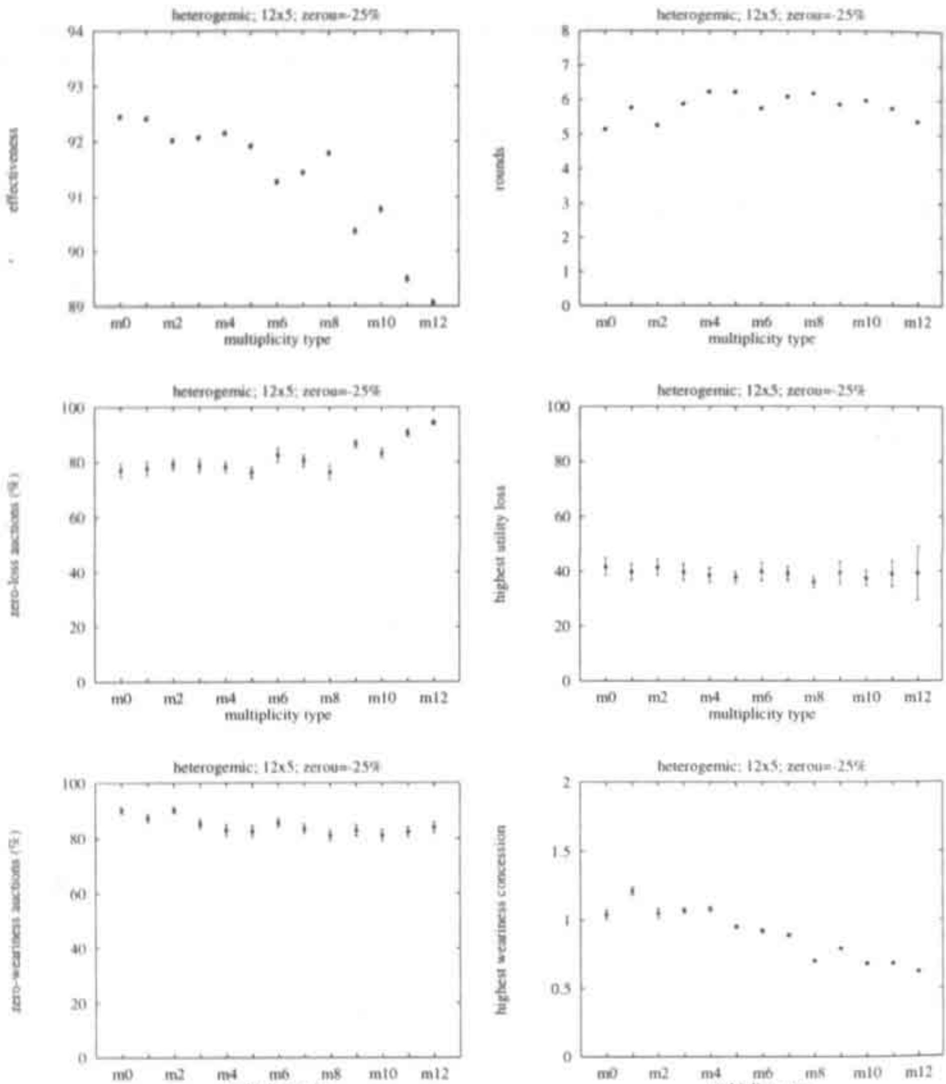


Figure 7.4: Influence of tool bag eccentricity on IG performance.

possessors with initial utilities as high as 80 or 90% stick to their original proposal, despite the extremely low success probability. The reason for this seemingly reckless behavior is that, in very eccentric markets, the most numerous tool type tends to be the *only* oversupplied one. As a consequence, agents endowed with this tool type can afford to be reckless. While the risk of failure is generally very high, the *impact* of failure is nil: if their proposal is rejected, they keep their initial endowment, since it is the only oversupplied tool type.

In contrast, agents endowed with a tool type x, other than a, and a higher utility for x than for a will *always* lose utility if their proposal fails. And any type-a possessor that happens to share the first preference of these agents induces a decrease in their subjective estimate of the probability of success. As we have observed in the interactive simulator, the fierce competition by a-possessors often incents other agents to adjust their proposals or withdraw from the auction, despite the fact that they are more eligible, and would, in fact, have succeeded in obtaining their first preference had they been more courageous.

The combination of fearless a-possessors and intimidated others explains why there is little utility loss: The a-possessors cannot lose any utility, while the others seldom dare to take the risk. And the fact that the eligibility heuristic of IG has no effect on the bidding behavior of a-possessors, and an adverse effect on others explains why IG performs poorly in terms of allocation effectiveness.

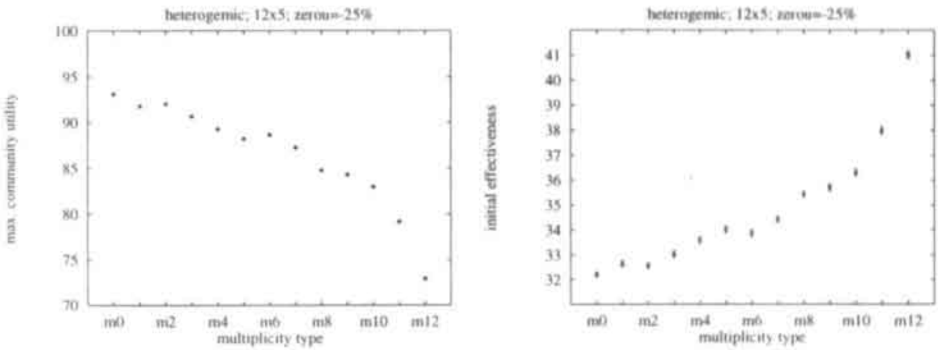


Figure 7.5: Influence of tool bag eccentricity on problem difficulty.

In Fig. 7.5, we see that reassignment problems with an eccentric tool bag are also difficult in a general sense, as far as the optimal community utility is concerned. The low value of the optimal community utility at m12 can be attributed to the fact that at least four of the eight a-possessors are *doomed* to end up with their current endowment.

The average-initial-effectiveness curve, however, seems to indicate that the effectiveness attainable with random reallocation *increases* with increasing eccentricity. Note, however, that the average initial effectiveness is not a good indicator for problem difficulty in this case, since it is computed over *viable* initial assignments only. Thus, it indicates how much effectiveness one can expect to obtain *from those tool types which are not first preferences*. However, in markets with low eccentricity, most of the attained community utility stems from first preferences. Hence, if we were to compute the average effectiveness obtained with random reassignment (i.e., *without* demanding viability) the resulting curve would most likely be decreasing with increasing eccentricity, just like the optimal-community-utility curve.

7.3.3 The number of tool types

The number of tool types (m) determines, together with the number of agents (n), whether a problem instance is typed or untyped. Up to this point we have used this distinction as an binary one: Problems with $m = n$ were labeled as untyped, while all other cases (i.e., $m < n$) were referred to as typed problems. It seems likely, however, that mechanism performance on problems that are 'nearly untyped' (e.g. with $m = n - 1$) differs little from performance on untyped ($m = n$) problems. Hence, we can say that the number of tool types determines (in relation to the number of agents) to what *extent* the problem instances in the sample space are typed.

If one investigates the impact of the 'typedness' of problems on IG performance by varying m while clamping all other parameters to their default setting, one is confronted with oscillatory fluctuations of the form shown in Fig. 7.6.

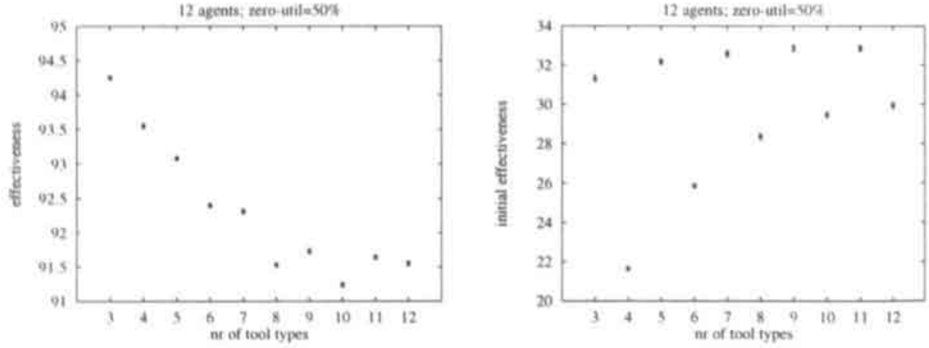


Figure 7.6: Oscillatory variations in performance and problem difficulty as functions of the number of tool types.

The oscillations are relatively minor in the effectiveness graph, but very pronounced in the initial-effectiveness graph. Apparently, the value $m = 4$ does not lead to the generation of problems of an inbetween nature of those generated with $m = 3$ and $m = 5$. The cause of these oscillations lies with the `zero utility` parameter, or rather, the interaction between this parameter and the number of tool types.

The `zero utility` parameter specifies the percentage of zero entries in the rows of the utility matrix. With the (homogemic default) `zero utility` value of 50%, the *actual* percentage of zero entries varies nonmonotonically with m due to rounding.

For odd m, 50% of $m-1$ is integer, while for even m, it is not. Hence, rounding takes place iff m is even. As a consequence, the actual percentage of zero elements per row of the utility matrix is more than 50% (of $m-1$) if m is even (e.g. 67, 60, and 57% for $m = 3, 5, 7$).

From our investigation of the effects of varying `zero utility` with a fixed number of tools, we know that both the effectiveness and the initial effectiveness are decreasing near `zero utility` = `50%`, but the rate of decrease is much higher[8] for the initial effectiveness attribute. This explains why the oscillation is more pronounced in the second (initial-effectiveness) graph of Fig. 7.6.

The oscillation disappears if we clamp the *number* N of useful alternatives instead of the *percentage* of useless ones. This has been done in the simulations that underly Fig. 7.7. The graphs in this figure reveal the influence of the number of tool types on IG performance and problem difficulty, if $N = 2$.

It is subject to discussion which of the two alternatives (the number or the percentage of (non)zero entries per matrix row) constitutes the best problem type indicator. This becomes most apparent if we look at a problem from the default homogemic sample space, and ask ourselves which sample space we are referring to if we speak of problems that are three times as large, but of the same type. Presuming that "three times as large" means three times as many (i.e., 36) agents and three times as many (i.e., 15) different tool types, the question is whether "of the same type" means that there are 2 or 7 tool alternatives for each agent.

While we will return to this question when discussing scaling effects, it is not necessary to answer it at present: From Fig. 7.6 and Fig. 7.7, we can conclude that — whichever option we choose — IG tends to perform better on problems that are (strongly) typed than on problems that are (nearly) untyped. The same is true with respect to the general difficulty in terms of the optimal community utility.

The latter phenomenon is the easiest one to understand. Our method of problem generation is such that the correlation between agent preferences for tool types is zero on average. Consequently, when the agents specify their first preferences in the first bidding round, the tool scarcities will be low in comparison with the total trade volume if the market is highly typed. Hence, a relatively large portion of the agents' first preferences can be allocated. This explains why the optimal community utility tends to be high in such markets.

It is less obvious why the average effectiveness is also higher in highly typed markets. A plausible explanation is the fact that relatively low scarcities imply relatively high success probabilities of the agent's proposals. Hence, agents will be less prone to withdraw from the bidding and stick to their current endowment (which usually

[8] To compare the rates of decrease, one should take the different vertical scalings of the three graphs into consideration.

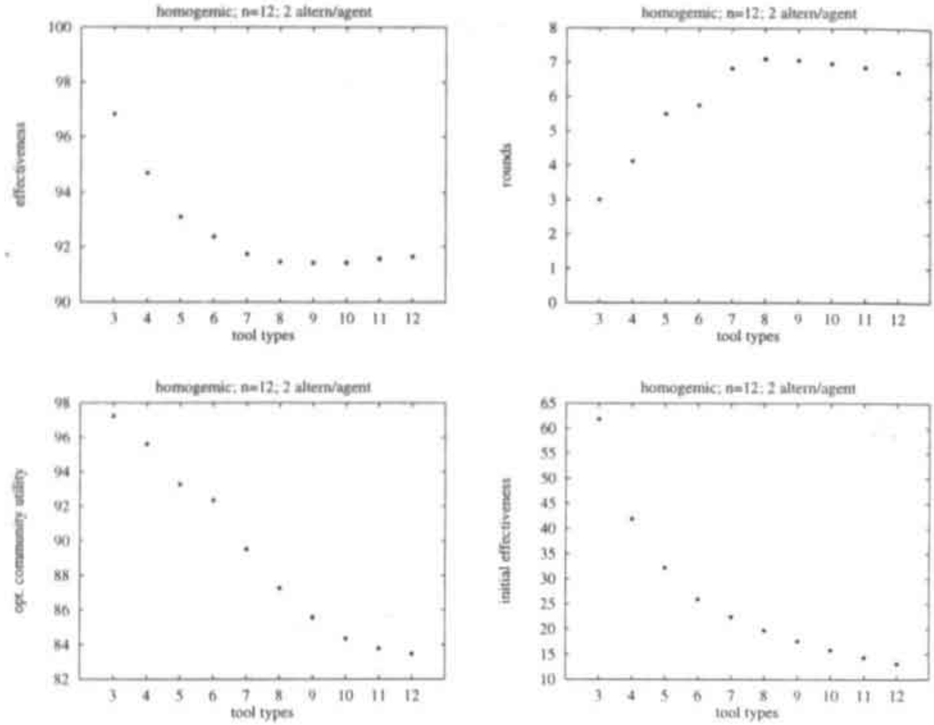


Figure 7.7: Influence of `nr of tool types` with a fixed number of tool alternatives.

hurts some other agent, and sometimes also the withdrawing agent itself).

7.3.4 Scaling of performance

In the context of *untyped* reassignment problems, the scale of problems is determined by a single parameter: the number of agents n. If we define the scale of a problem as a quantity proportional to the number of bits needed to represent it, the scale of a *typed* reassignment problem is roughly proportional to the product of the number of agents (n) and the number of tool types (m). However, it is not at all evident that any change in $n \cdot m$ constitutes a change of scale.

On one hand, there are good reasons to speak of an increase of scale, if the number of agents doubles, while the number of tool types does not change. To provide a (more or less) concrete example, we recall the software-engineering company that was used to describe our prototypical tool-reallocation problem (Sect. 2.1.1; page 19). In this context, an agent represents a project team of programmers, and a tool is a computer. Hence, if the number of agents doubles, the number of tools is likely to double — or at least increase — also. However, it is quite possible that the number of tool *types* does not change at all. Yet, one would definitely speak of an increase of scale.

On the other hand, we have qualified our experiments involving variation of m with fixed n as an investigation of the impact of the *typedness* of problems on IG performance. Typedness is not synonymous to scale. Indeed, it would be improper to qualify the difference between a problem with 12 agents and 12 tool types and a problem with 24 agents and 12 tool types as a matter of scale: The two problems differ in *nature*; the first one is untyped, while the second is typed.

The question how one should define 'variation of scale' in the context of typed reassignment is not an academical one, because the effect of an increase of scale on the quality of solutions (in terms of allocation effectiveness) *depends* on the particular definition that we choose to employ.

If we regard an increase of n with constant m as a scale increase, the average solution quality increases with scale, while it appears to be approximately constant if we define an increase of scale as an increase of $n \cdot m$ which does not involve any change in the typedness of the problems (i.e., the ratio $\frac{n}{m}$). Moreover, if we employ a variant of the latter scale-increase definition in which we fixate the *number* of useful tool alternatives per agent instead of the percentage (as defined by the `zero utility` parameter), the average solution quality decreases with increasing scale.

Because it depends on the specific real-life context of the reassignment problem which of these three definitions of scale is the most appropriate one, we have performed separate experiments for the three interpretations. Below, we discuss the outcomes in the following order.

1. the effects of increasing the population size n with constant m;
2. the effects of increasing n and m, with constant typedness $\frac{n}{m}$ and constant percentage of tool alternatives (i.e., constant `zero utility`;)
3. the effects of increasing the scale $n \cdot m$ with constant typedness, and a constant number of tool alternatives per agent.

7.3.5 Effects of increasing the population size

Fig. 7.8 portrays the effects of increasing n with constant m, for $m = 4$, 5, and 6. Both the average effectiveness and the average auction duration appears to increase with n. In other words, with increasing population size, IG performance tends to be lower in terms of computation time, but *higher* in terms of allocation effectiveness.

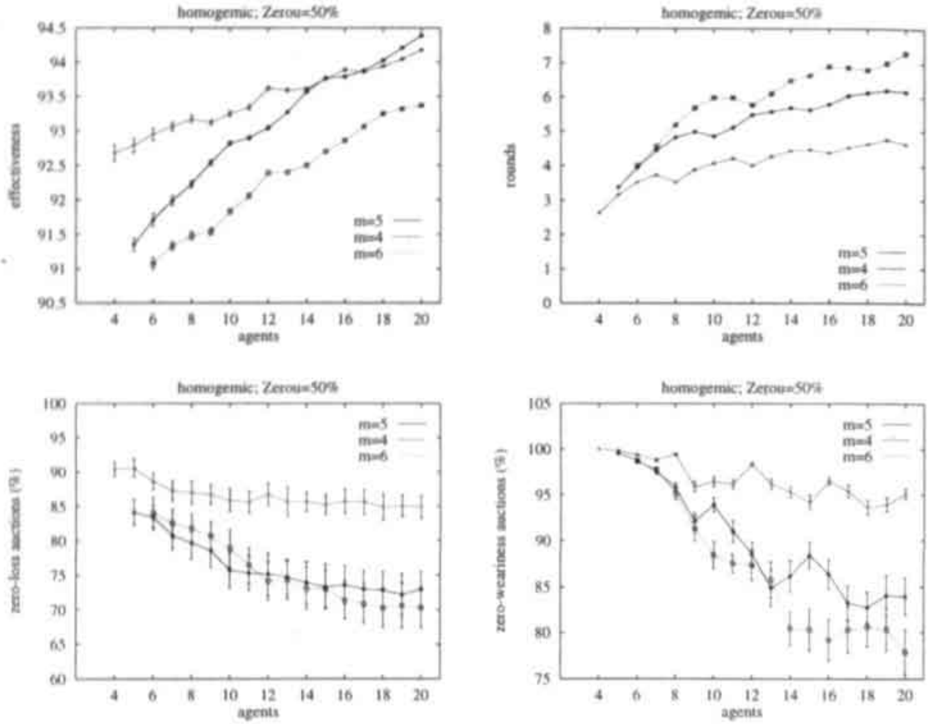


Figure 7.8: Influence of population size on IG performance.

This is conform our expectation. It is to be expected that it takes more time to reach an equilibrium state if there are more agents. As of the increase of allocation effectiveness with population size, this is not really an enigma either, upon realizing that increasing n with a fixed m increases the 'typedness' of the sample problems, that is, the proportion $\frac{n}{m}$. In Sect. 7.3.3, we provided a plausible explanation for the observation that the average effectiveness obtained by IG on highly typed problems is relatively high.

The curves in Fig. 7.8 seem to consist of an oscillation superimposed on a monotonic trend. This is most clearly visible in the **rounds** plot. The graphs in this plot exhibit bumps. Moreover, the width[9] of these bumps seems to correlate with m: it equals 4 for the lowermost ($m = 4$) curve, 5 for the middle ($m = 5$) curve, and 6 for the topmost ($m = 6$) curve.

These bumpy curves can be described as the superposition of an oscillation and a monotonic trend. In the sequel, we will refer to such curves as *semi-periodic*.

[9]The width of a bump is defined as the number of datapoints encountered as we move from one local minimum to the next.

The observation that the curves are semi-periodic with period m suggests that the quantity $n \bmod m$ should be viewed as a problem characteristic which exerts its own influence on IG performance, independent of the problem scale (and the typedness $\frac{n}{m}$).

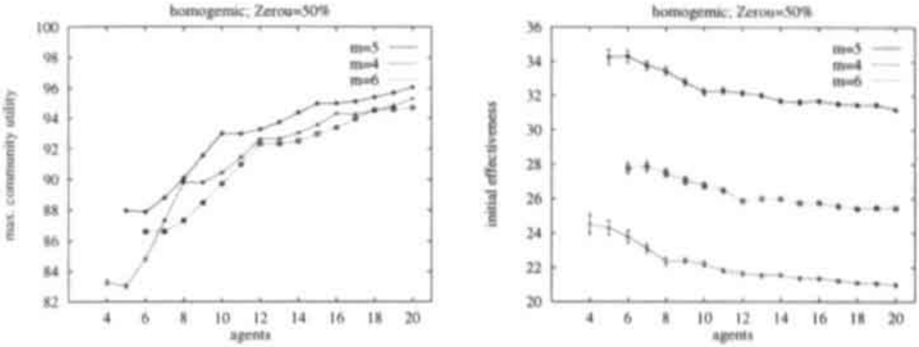


Figure 7.9: Influence of population size on problem difficulty.

That the semi-periodicity is indeed due to the variation of a problem attribute, rather than to properties of IG in particular is confirmed by Fig. 7.9, which pictures the influence of the population size on two aspects of general problem difficulty: the sample-space properties "optimal community utility" and "initial allocation effectiveness". Again, the curves appear to be semi-periodic, with a period of m (though the oscillation is less pronounced in the second graph).

7.3.6 Scaling effects with constant typedness

The above outcomes indicate that increasing the population with a fixed number of tool types does not only influence the scale, but also the *nature* of the generated problem instances. To get rid of the spurious oscillations, we have performed experiments on the effects of increasing m, with fixed $n \bmod m$ and $n \operatorname{div} m$. Thus, the typedness $\frac{n}{m}$ is approximately constant, and the quantity $n \bmod m$ does not change at all.

In the experiments underlying Fig. 7.10, $n \bmod m = n \operatorname{div} m = 2$. For the heterogemic curves, we used the default setting `zero utility` $= -25\%$. For the homogemic case, we have set `zero utility` to -50% instead of 50%, to reduce the downward bias at even values of m.[10] This change of the `zero utility` setting has no effect for odd m, while it decreases the average number of zero entries per row for even m from $m/2$ to $m/2 - 1/2$. With $m = 8$ (for example), the setting `zero`

[10]This downward bias was observed in Fig. 7.6, and diagnosed as a consequence of rounding of Z.

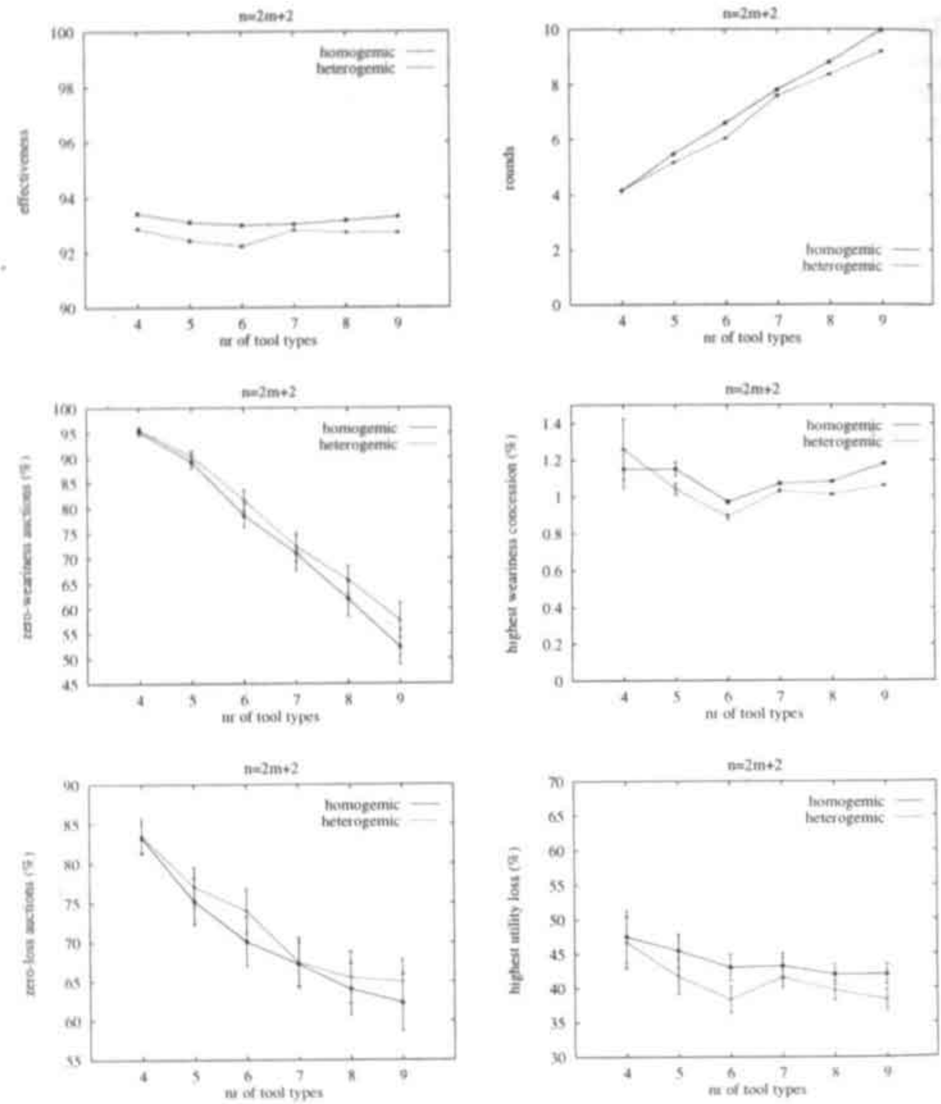


Figure 7.10: Scaling of IG performance with fixed `zero utility`.

`utility` = −50% renders matrices with 3 zero entries per row in the upper half and 4 per row in the lower half. Formally, such matrices are heterogemic, but they are nearly homogemic, so to speak.

There appears to be little difference between the average effectiveness obtained on the homogemic and the heterogemic problems. Also, the effectiveness obtained by

IG appears to be almost insensitive to problem scale. The average auction duration, however, rises relatively quickly with the scale of the problems. Not surprisingly, this is accompanied by a considerable decrease of the percentage of zero-weariness auctions, but the problem scale appears to have little impact on the highest weariness concession, which is approximately 1% (on average) for each of the investigated problem dimensions.

As for utility loss, the bottom two curves show a considerable decrease in the percentage of zero-loss auctions with increasing scale, while the amount of utility loss tends to decrease somewhat (though not monotonically). Please note that the fact that utility loss is more likely to occur if the auctions are larger does *not* imply that agents who take part in large auctions will suffer utility losses relatively often. We will argue later (in Sect. 7.3.8) why the probability of an individual agent experiencing utility loss is virtually independent of the problem scale.

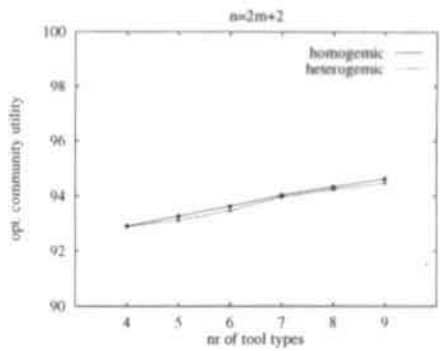


Figure 7.11: Scaling of problem difficulty with fixed `zero utility`.

7.3.7 Scaling effects with a fixed number of tool alternatives

In Fig. 7.11, we can observe that the average problem difficulty in the sample space (in terms of the maximally attainable community utility) *decreases* with increasing problem scale. This contradicts our intuition about the influence of scale on problem difficulty in real life. Is it really true that larger tool-reassignment problems are *easier* in the sense that they allow for a higher average degree of agent satisfaction to be obtained?

The answer to this question depends on our definition of scaling: what are "similar, but larger" problems? It turns out that such problems are indeed easier if "similar" means that the two samples are generated using the same value of `zero utility`, but they are more difficult if it means that the *number* of tool alternatives per agent is the same.

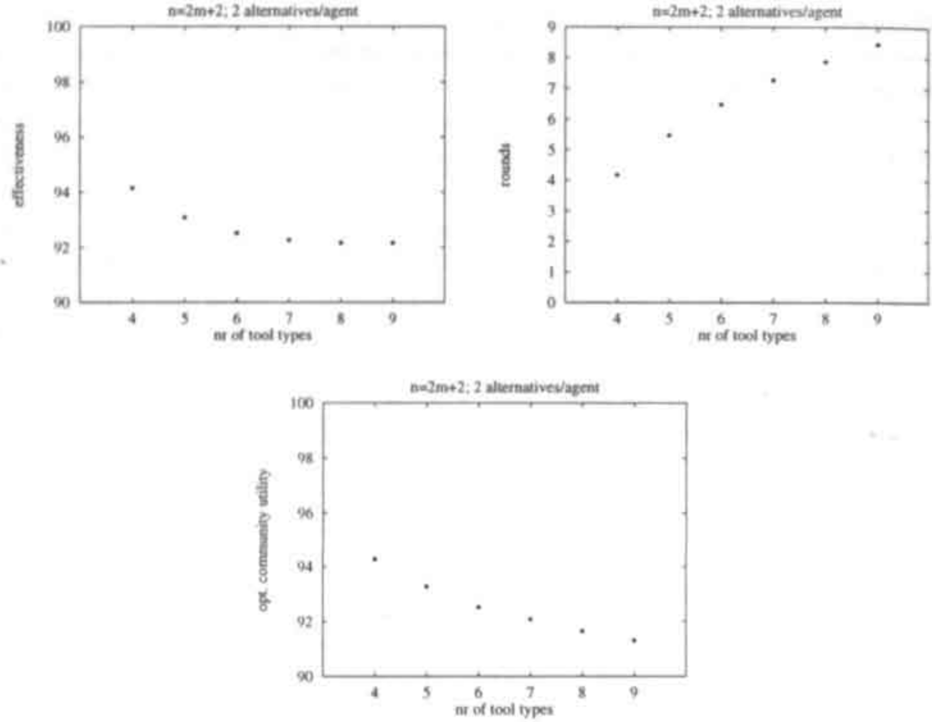


Figure 7.12: Scaling effects with a fixed number of useful alternatives.

The outcome of experimentation with the latter interpretation of 'similar' is shown in Fig. 7.12. In this case, the average problem difficulty (in terms of the optimal community utility) appears to increase with scale, while IG's average performance decreases, in terms of solution quality as well as auction duration.

7.3.8 The impact of scale on the risk of utility loss

Upon looking at the outcome of our experiments on scaling of IG performance with constant `zero utility` (cf. Fig. 7.10), one would be tempted to conclude that the *severity* of utility loss does not change a whole lot if the problem scale increases, but the *risk* of experiencing some utility loss increases considerably. This is not true, however. Even though the frequency of *auctions* with utility loss increases, the probability P_l of utility loss that an *agent* has to reckon with if it takes part in an IG auction is approximately 2% in all cases.

Denoting the average percentage of zero-loss auctions by ZL and the average percentage of losers (in case of utility loss) by PL, the relation between ZL, PL and

P_l is

$$P_l = \frac{100 - ZL}{100} \cdot PL \tag{7.1}$$

Table 7.5 contains a survey of the relevant statistics for heterogemic problems of different dimensions. As apparent from the fourth column, P_l is hardly affected by the scale of the problem, because the decrease of ZL with increasing scale is compensated by a decrease of PL.

dimension		ZL	PL	P_l	number of losers			hom. P_l
n	m				min.	avg.	max.	
10	4	83.5	10.3	1.7	1	1.03	3	1.7
12	5	77.1	8.7	2.0	1	1.05	3	2.2
14	6	74.0	7.6	2.0	1	1.07	3	2.3
16	7	67.4	6.9	2.2	1	1.10	3	2.3
18	8	65.5	6.2	2.1	1	1.11	3	2.2
20	9	65.0	5.6	2.0	1	1.12	3	2.2

Table 7.5: Utility loss statistics for different problem dimensions.

In the last column (labeled "hom. P_l"), we listed the values of P_l in the homogemic problem spaces. Except for the low value of 1.7% associated with the smallest problems, P_l is virtually the same for all problem dimensions in this case also.

7.4 Evaluation of the Chosen Problem Parameters

Up to this point, our interest in difficult problems was primarily due to the requirement that the sample space used to compare the performances of various architectural variants of the basic Informed Gambling mechanism should at least provide some challenge to IG. I claim that the default setting of the problem generation parameters is adequate in this respect. The experiments have rendered estimates of sample space averages for IG effectiveness that varied between 89% (the data point at m12 in Fig. 7.4) and 99% (the data points on the low-alts and high-alts curves at `zero utility = 0%` in Fig. 7.2.[11] As such, the default sample space, with its estimated average IG effectiveness of 92%, can be qualified as moderately difficult, and is therefore a suitable sample space to investigate the impact of architectural variation.

[11] We have excluded the 100% effectiveness scores obtained on the trivial problem spaces with `zero utility = 100%`.

7.4.1 Where are the *really* difficult problems?

However, it is also important to know what the *really* difficult problems are, not only to uncover possible weak spots of IG, but also to determine whether the chosen set of problem parameters is adequate in the sense that these problems can be characterized in terms of the setting of these parameters.

We have already observed that sample spaces associated with highly eccentric tool bags tend to be difficult for IG. Another problem attribute that appears to correlate with low IG performance is a high (but not maximal!) percentage of useless tool types, in particular in combination with a range of alternative tool utilities that is narrow and close to maximal utility.

We did not vary all generation parameters simultaneously, so we may be able to witness an even lower average IG effectiveness if we combine the above settings of generation parameters. To this avail, we tried out the problem parameter setting of 75% `zero utility`, `low alternative` = 8, `high alternative` = 9 and a tool bag of maximal eccentricity. The IGRT output with this setting is shown in Table 7.6.

```
STATISTICS OVER 200 x 100 AUCTIONS (with various matrices):

opt. unconstrained community utility: 31.67-96.67 (avg=67.91; var= 1.1919)
opt. ind.rat. effectiveness: 91.67-100.00 (avg=99.97; var= 0.0008)
initial allocation effectiveness:  0.00-94.44 (avg=30.66; var= 1.9750)

final allocation effectiveness: 60.24-100.00 (avg=90.69; var= 0.5460)
perc. of zero-loss auctions: 95.00-100.00 (avg=99.10; var= 0.8760)
avg. norm. utility loss per loser: 80.00-90.00 (avg=82.51; var=14.4010)
highest norm. utility loss: 80.00-90.00 (avg=82.51; var=14.4010)
percentage of losers:  8.33- 8.33 (avg= 8.33; var= 0.0000)
perc. of zero-weariness auctions: 83.00-98.00 (avg=90.72; var= 8.5816)
max. norm. weariness concession:  0.00- 8.50 (avg= 1.13; var= 0.0525)

nr of bids: 36.00-596.00 (avg=66.23; var= 4.6182)
nr of rounds:  3.00-47.00 (avg= 5.93; var= 0.0620)
nr of phases:  1.00- 2.00 (avg= 2.00; var= 0.0000)
```

Table 7.6: IGRT statistics on $S(aaaaaaaabcde, 10, 3, 8, 9)$.

Contrary to our expectation, the average IG effectiveness associated with this sample space (90.69%) exceeds the average effectiveness obtained on the most difficult space that was encountered thus far (89.06%, with a tool bag of maximal eccentricity). Apparently, problem difficulty that stems from different generation parameters does not simply add up.

There is another figure in Table 7.6 that deserves our attention. The lowest effectiveness obtained on any problem in the sample appears to be 60.24%. This is so far below the lowest sample-space average (of 89.06%) we encountered that it raises doubts about the adequacy of the chosen problem-generation parameters as tools to characterize the difficult problems.

In our analysis of the low average IG effectiveness obtained with the sample space $\bar{S}(aaaaaaaabcde, 10, 1, 3, 9)$, we identified the high probability that tool type a will be the *only* oversupplied tool type (in most rounds of the auction) as a primary cause for IG's relatively low performance. Hence, a problem parameter that correlates with such concentrated oversupply could prove to be more adequate in capturing problem difficulty than our problem generation parameters. A promising candidate in this respect is the *tension* in the reassignment problem.

We have already defined the tension for an elementary composition problem in Chapter 3.2.7, as a measure of the discrepancy between tool type supply and demand. The tension in a reassignment problem is defined analogously.

Definition 7.1 (tension)
Let Γ and π denote the tool bag and the bag of first preferences of a reassignment problem $\langle e, U\rangle$ with initial assignment e and utility matrix U. Then the tension t of $\langle e, U\rangle$ equals

$$\frac{1}{2} \cdot \sum_{x \in R} |\Gamma(x) - \pi(x)| \tag{7.2}$$

△

The sum in (7.2) counts the total oversupply plus the total undersupply. Since these two entities are always the same, the tension can be defined in words as the total oversupply that will occur if every agent proposes to exchange its initial endowment for its first preference.

There are several reasons to expect that the tension may prove to be a good predictor for the effectiveness that can be obtained by IG. First, a problem instance with minimal (i.e., zero) tension amounts to an equilibrial market. Such problems are obviously trivial for IG, both in terms of effectiveness and in terms of auction duration. In contrast, if a problem instance features maximal tension, only few agents can be endowed with their first preference. In other words, many agents will have to seek for alternatives. It is plausible that the average effectiveness obtained by IG (or any other informationally decentralized mechanism) on such problems will be low, because many agents have the opportunity to take a wrong decision.

Second, there is a correlation between eccentric tool bags and the occurrence of problems with high tension, as expressed in Prop. 7.2 below.

Proposition 7.2
The highest tension that can occur in any viable problem instance with a tool bag Γ *of size* n *and multiplicity type* α *is*

$$n - 2 \cdot \min \alpha$$

Proof.
Let Γ, α, and n be defined as above. Define $R = \check{\Gamma}$. We assume that $|R| \geq 2$. Since $\sum_x \Gamma(x) = \sum_x \pi(x) = n$, the sum in (7.2) is maximal if we focus all of the demand π on a single tool type x_0, with minimal multiplicity in Γ. In other words, if

$$\Gamma(x_0) = \min_{x \in R} \Gamma(x)$$

then the expression (7.2) for the tension attains its maximum value (of $n - \Gamma(x_0)$), with the preference bag $\pi_0 \triangleq \{x_0 : n\}$.

However, in Prop. 4.44, we derived the following necessary and sufficient condition for the *existence* of a viable reassignment problem, in terms of the tool bag Γ, the bag of first preferences π, and the number of agents n.

$$(\forall x \in R) \quad \pi(x) + \Gamma(x) \leq n \tag{7.3}$$

Hence, since $\pi_0(x_0) + \Gamma(x_0) = n + \Gamma(x_0) > n$, π_0 conflicts with the datum that we are dealing with a *viable* problem instance. To comply with (7.3), we must ensure that $\pi(x_0) \leq n - \Gamma(x_0) = n - \min_x \Gamma(x)$. To do this, while keeping the demand as focused as possible, we pick an arbitrary $x_1 \in R \backslash \{x_0\}$, and define

$$\pi_1 \triangleq \{ x_0 : n - \Gamma(x_0) \ , \ x_1 : \Gamma(x_0) \}$$

Now we have

$$\begin{cases} \pi_1(x_0) + \Gamma(x_0) & = & n - \Gamma(x_0) + \Gamma(x_0) & = & n \\ \pi_1(x_1) + \Gamma(x_1) & = & \Gamma(x_0) + \Gamma(x_1) & \leq & n \\ \pi_1(x) + \Gamma(x) & = & \Gamma(x) & < & n, \quad \forall x \in R \backslash \{x_0, x_1\} \end{cases}$$

In other words, π_1 complies with (7.3). Hence, there exists a viable problem instance $\langle e, U \rangle$, compatible with Γ and π_1. The tension in such a problem instance equals

$$\begin{aligned} t &= \tfrac{1}{2} \cdot \sum_{x \in R} |\Gamma(x) - \pi_1(x)| = \\ &= |\Gamma(x_0) - \pi_1(x_0)| + |\Gamma(x_1) - \pi_1(x_1)| + \sum_{x \in R \backslash \{x_0, x_1\}} |\Gamma(x) - \pi_1(x)| \end{aligned}$$

Because $|R| \geq 2$, and $\Gamma(x_0) = \min_x \Gamma(x)$, we know that $\Gamma(x_0) \leq n - \Gamma(x_0)$. Hence,

$$\begin{aligned} t &= \tfrac{1}{2} \cdot \Big(\big(n - \Gamma(x_0) - \Gamma(x_0)\big) + \big(\Gamma(x_1) - \Gamma(x_0)\big) + \big(n - \Gamma(x_0) - \Gamma(x_1)\big) \Big) \\ &= \tfrac{1}{2} \cdot \big(2 \cdot n - 4 \cdot \Gamma(x_0) \big) = n - 2 \cdot \min_x \Gamma(x) = \\ &= n - 2 \cdot \min \alpha \end{aligned}$$

Of course, π_1 is not the only kind of first-preference bag that can result from moving $\Gamma(x_0)$ of π_0's preferences for x_0 to other tool types. We can also distribute these $\Gamma(x_0)$ preferences over *several* other tool types. Note, however, that if we adapt π_1 by moving k of the preferences for x_1 to some other tool type $x_2 \in R\backslash\{x_0, x_1\}$, this does not change the tension. The contribution $|\Gamma(x_1) - \pi_1(x_1)|$ of x_1 increases, but — since $(\forall x \in R) \quad k \leq \Gamma(x_0) \leq \Gamma(x)$ — that of x_2 decreases by the same amount k. Hence, there is no preference bag that renders a higher tension than (a bag of the form) π_1.

∎

All in all, it seems worthwhile to investigate to what extent high tension correlates with low IG effectiveness. We do this by comparing the effectiveness obtained by IG on problem samples with various, fixed values of tension. Because there is no problem generation parameter in the IGRT that can generate utility matrices with a prespecified tension, we have resorted to fixed-matrix simulation to obtain such samples. We picked a utility matrix that was generated by the IGRT with the default homogemic setting of the generation parameters, and exchanged agent utilities to obtain matrices with tension 1,2,..., 8.

$$U_1 = \begin{pmatrix} 3 & 10 & 0 & 9 & 0 \\ 7 & 10 & 4 & 0 & 0 \\ 0 & 5 & 10 & 0 & 8 \\ 10 & 6 & 0 & 8 & 0 \\ 8 & 0 & 8 & 10 & 0 \\ 0 & 0 & 8 & 10 & 5 \\ 0 & 8 & 0 & 7 & 10 \\ 0 & 0 & 5 & 10 & 9 \\ 8 & 0 & 10 & 0 & 3 \\ 8 & 10 & 0 & 0 & 9 \\ 10 & 0 & 6 & 0 & 7 \\ 10 & 0 & 7 & 6 & 0 \end{pmatrix} \qquad U_2 = \begin{pmatrix} 3 & 10 & 0 & 9 & 0 \\ 7 & 10 & 4 & 0 & 0 \\ 0 & 5 & 10 & 0 & 8 \\ 0 & 6 & 10 & 8 & 0 \\ 8 & 0 & 8 & 10 & 0 \\ 0 & 0 & 8 & 10 & 5 \\ 0 & 8 & 0 & 7 & 10 \\ 0 & 0 & 5 & 10 & 9 \\ 8 & 0 & 10 & 0 & 3 \\ 8 & 10 & 0 & 0 & 9 \\ 10 & 0 & 6 & 0 & 7 \\ 10 & 0 & 7 & 6 & 0 \end{pmatrix} \tag{7.4}$$

The matrices U_1 and U_2 in Eq. 7.4 are the first two of these eight matrices. Matrix U_2 was constructed from U_1 by exchanging the elements in columns 1 and 3 of row 4 (i.e., the utilities that agent 4 associates with tool types a and c). This changes the bag of preferences from $\pi_1 = aaabbbccddde$ for U_1 to $\pi_2 = aabbbcccddde$ for U_2. Because the tool bag is $aaabbbccddee$ in both cases, the tension is increased by one ($t = (|3-3| + |3-3| + |2-2| + |2-3| + |2-1|)/2 = 1$ for U_1, and $(|3-2| + |3-3| + |2-3| + |2-3| + |2-1|)/2 = 2$ for U_2. The other matrices are constructed in a similar fashion (U_3 from $U_2, \ldots$). The associated bags of first preferences are shown in Table 7.7.

Exhaustive fixed-matrix simulation entails the application of IG to all viable problems $\langle e, U \rangle$, for some fixed utility matrix U. Consequently, the performance-attribute

		a	b	c	d	e
π_1	=	3	3	2	3	1
π_2	=	2	3	3	3	1
π_3	=	2	2	4	3	1
π_4	=	2	1	4	4	1
π_5	=	1	1	5	4	1
π_6	=	0	1	5	5	1
π_7	=	0	0	6	5	1
π_8	=	0	0	6	6	0

Table 7.7: The first-preference bags associated with $U_1, \cdots, U_8$.

averages reported by IG (and plotted in Fig. 7.13) represent exact values, rather than estimates.

Even though the respective sample spaces are smaller than those we have investigated earlier, the samples (which *are* the sample spaces in this case) comprise more than the usual 20,000 problem instances. The number of different viable assignments for the matrices vary between 108,657 (for U_1) and 143,200 (for U_5). Because of this, and the fact that the matrices are all very similar except for their tension, the results constitute convincing proof that there is indeed a strong correlation between (high) tension and (low) IG effectiveness. From the first graph in Fig. 7.13, we conclude that the tension of a reassignment problem has a greater impact on IG effectiveness than any of the problem generation parameters investigated earlier. Moreover, high tension also appears to correlate positively with low performance in other respects (in terms of auction duration as well as the frequency of utility loss and weariness), and with high problem difficulty in general (in particular in terms of optimal community utility).

The matrix U_8 with *maximal* tension constitutes a somewhat surprising exception to the general pattern of decreasing performance with increasing tension. The sample space with $t = 8$ renders a *higher* average effectiveness than the sample space with $t = 7$. This suggest that the most difficult problems are those with *nearly* maximal tension. At this point, the evidence for such a hypothesis is rather scanty. I cannot exclude that a different choice of the preference bags π_i might have led to a performance low at $t = 8$ instead of $t = 7$.[12] However, there is at least one other finding that seems to support the hypothesis. In our analysis of the low average effectiveness obtained in the context of an eccentric tool bag, we identified the presence of a *single* oversupplied tool type (with high multiplicity) as a primary cause of failure.

[12] In particular, it may make a difference if we choose to focus the demand on a single tool type, thus working toward $\pi_8 = \{c\!:\!10, d\!:\!2\}$ instead of $\{c\!:\!6, d\!:\!6\}$.

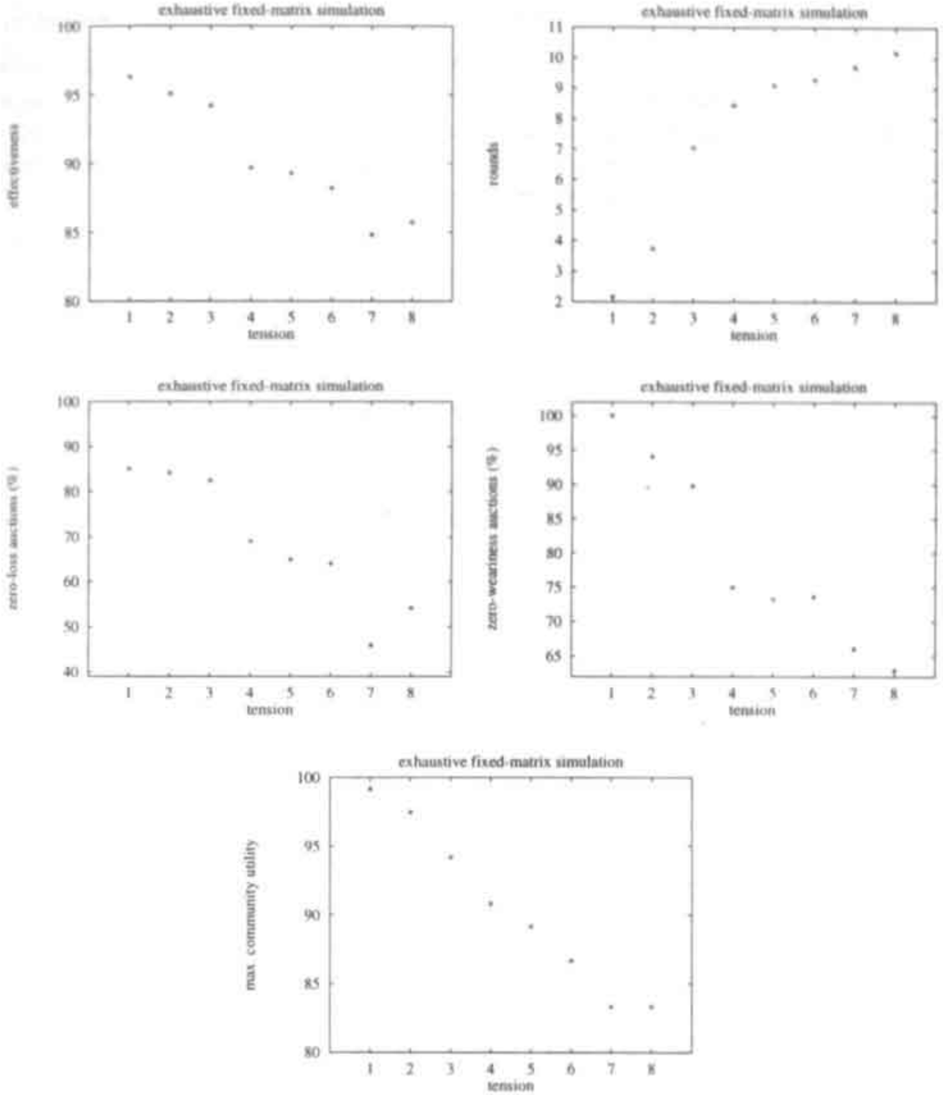


Figure 7.13: Influence of tension on performance and problem difficulty.

The argumentation used in this analysis allows for some generalization: It is likely that the relaxation incentives which IG's eligibility definition is intended to provide will work less well if *most* of the oversupply is confined to *fewer* tool types. This provides us with a plausible explanation as to why the lowest average effectiveness seems to be obtained with near-maximal tension: The oversupply is usually more focused in problems with moderate tension than in those with maximal tension.[13]

There is still a considerable gap between the lowest average effectiveness of problems with near-maximal tension (viz. 84.84% at $t = 7$) and the lowest effectiveness obtained on an individual problem instance (viz. 60.24% obtained with the sample space $S(aaaaaaaabcde, 10, 3, 8, 9)$.[14]). This suggests that the tension is not really adequate to characterize the most difficult problems either.

However, the gap is smaller than the figures 84.84 and 60.24 suggest. Because IG features nondeterministic decisions, its lowest performance ever is bound to involve some bad luck, which is not due to *any* property of the associated problem instance. Hence, upon testing the extent to which the tension can explain low IG performance, we should not compare the average performance in the $t = 7$-sample-space with the worst performance ever exhibited by IG, but with the *average* performance of *repeated* application of IG to the problem instance which led to the worst performance.

We have performed such repeated IG-invocation to two difficult problem instances: the instance which rendered the aforementioned effectiveness low of 60.24%, and another problem instance which rendered — in a second simulation run with the same sample space — an even lower effectiveness: 54.79%. The associated utility matrices are shown in Eq. 7.5 below. Before presenting the outcomes of repeated invocation, we trace the original invocation of IG to the two problem instances.

$$V_1 = \begin{pmatrix} \underline{0} & 10 & 0 & 0 & 9 \\ 0 & 10 & 0 & 9 & \underline{0} \\ \underline{0} & 0 & 10 & 8 & 0 \\ 9 & 0 & 10 & \underline{0} & 0 \\ 8 & \underline{0} & 0 & 0 & 10 \\ \underline{0} & 8 & 0 & 0 & 10 \\ \underline{9} & 0 & 0 & 10 & 0 \\ \underline{0} & 0 & 9 & 0 & 10 \\ \underline{0} & 10 & 0 & 0 & 9 \\ 8 & 10 & \underline{0} & 0 & 0 \\ \underline{0} & 0 & 10 & 8 & 0 \\ \underline{0} & 0 & 10 & 8 & 0 \end{pmatrix} \qquad V_2 = \begin{pmatrix} 9 & 0 & 10 & \underline{0} & 0 \\ \underline{0} & 10 & 0 & 9 & 0 \\ 8 & \underline{0} & 0 & 10 & 0 \\ \underline{9} & 10 & 0 & 0 & 0 \\ \underline{0} & 9 & 0 & 10 & 0 \\ \underline{9} & 0 & 0 & 0 & 10 \\ 0 & 9 & \underline{0} & 10 & 0 \\ \underline{0} & 10 & 0 & 0 & 9 \\ 10 & 0 & 0 & 9 & \underline{0} \\ \underline{0} & 8 & 0 & 10 & 0 \\ \underline{0} & 0 & 0 & 8 & 10 \\ \underline{0} & 0 & 8 & 0 & 10 \end{pmatrix} \tag{7.5}$$

When IG was first applied to the problem instance $\langle e_1, V_1 \rangle = \langle aeadbaaaacaa, V_1 \rangle$, with an initial effectiveness of 12.33%, it rendered the final assignment $aaaceadaabaa$. With this assignment, every agent obtains either 100% utility (viz. agents 4,5,7 and 10) or none whatsoever. The optimal community utility for this problem instance

[13] As an illustration, if $\xi(\Gamma) = \{3, 3, 2, 2, 2\}$, the highest tension values at which all initial over-supply in a viable problem instance can be confined to k tool types are $t = 3$ for $k = 1$ and $t = 6$ for $k = 2$.

[14] See Table 7.6.

is $\frac{73}{120} = 60.83\%$ (obtained, for example, by the assignment *adaaaeaabaac*). Hence, the effectiveness obtained by IG is $\frac{40}{73} = 54.79\%$.

Likewise, for the problem instance $\langle e_2, V_2\rangle = \langle dabaaacaeaaa, V_2\rangle$, with an initial effectiveness of 21.69%, and an optimal community utility of 69.17%, IG rendered the final allocation *cadbaeaaaaaa* with an effectiveness of 60.23%. Again, the final allocation is an all-or-nothing solution. Agents 1,3,4,6 and 9 obtain 100% utility, while the others obtain no utility at all.

With V_1, the bag of first preferences is $\{a{:}0, b{:}4, c{:}4, d{:}1, e{:}3\}$. Since the tool bag is $\{a{:}8, b{:}1, c{:}1, d{:}1, e{:}1\}$, the tension in $\langle e_1, V_1\rangle$ equals $0.5\cdot(8+3+3+0+2) = 8$. This is relatively high, but less than the highest tension that can occur with the tool bag $\Gamma = aaaaaaaabcde$ (which, by Prop. 7.2, equals $n - 2\cdot \min_x(\Gamma(x)) = 12 - 2 = 10$). For U_2, we find a bag of first preferences $\{a:1, b:3, c:1, d:4, e:3\}$, rendering a tension of $0.5\cdot(7+2+0+3+2) = 7$.

Thus, the two problem instances provide further support for the hypothesis — derived from Fig. 7.13 — that the most troublesome problem instances are those with high, but not maximal tension.

problem instance:	$\langle e_1, V_1\rangle$			$\langle e_2, V_2\rangle$		
outcomes:	final eff.	freq.	allocations	final eff.	freq.	allocations
	55%	201	*aaaceadaabaa*	60%	93	*cadbaeaaaaaa*
	64%	198	*aaaceaaaabad*	71%	317	*cadaaeabaaaa*
			aaaceaaaabda			*cadbaaaaaaea*
			aadceaaaabaa			*cbdaaeaaaaaa*
	66%	185	*abaceadaaaaa*	82%	190	*cadaaaabaaea*
	75%	216	*abaceaaaaaad*			*cbdaaaaaaaea*
			abaceaaaaada			
			abdceaaaaaaa			
Totals/averages:	65.25%	800		72.84%	600	

Table 7.8: Nondeterministic variation of IG performance.

We have applied tentative fixed-problem simulation (with a repetition factor of $k = 100$)[15] to the above problem instances $\langle e_1, V_1\rangle$ and $\langle e_2, V_2\rangle$ to see how much 'bad luck' was involved. The results are shown in Table 7.8. IG rendered eight different final allocations when applied to $\langle e_1, V_1\rangle$, with three different final effectiveness values.[16] Application of IG to $\langle e_2, V_2\rangle$ rendered six different solutions associated with three

[15]See the survey of simulation modes on page 248.

[16]The effectiveness values shown in the table are rounded to the nearest integer numbers. This is how they are rendered in the raw output file produced by the IGRT.

different effectiveness scores. Apparently, there was some bad luck involved in both cases. The minimal effectiveness of 55% (corresponding with the floating point value 54.79%) occurred in approximately 25% of the 800 invocations. The minimum of 60% (corresponding with 60.23%) on $\langle e_2, V_2 \rangle$ occurred even less frequently: in about 16% of 600 invocations.

The average effectiveness values of 65.25% on $\langle e_1, V_1 \rangle$, and 72.84% on $\langle e_2, V_2 \rangle$ are less extreme than the single-invocation values that we found for these problem instances in the grouped-sampling experiments, but they are still considerably lower than any sample space average we have encountered up to this point. This indicates that the tension is an important source of problem difficulty, but not the only one.

The sample spaces used for the tension experiments (of Fig. 7.13) differ in three respects from the sample spaces of the two difficult problem instances.

- The tool bag is less eccentric.
- The agents have more alternatives for their first preference.
- The average utility for the tool alternatives is lower.

The latter two of these differences comprise problem properties which do influence IG performance, but do not correlate with the tension. Hence, if we want to find out to what extent high tension is responsible for low IG effectiveness, we should compare the average effectiveness obtained with $t = 7$ with the average performance of repeated application of IG to the problem instance responsible for the lowest effectiveness in the *corresponding* (default homogemic) sample space. As noted earlier, the lowest observed effectiveness in a sample from this space was approximately 70%. Tentative fixed-problem simulation on the associated problem instance rendered an average IG effectiveness of 76.38% (over 900 invocations). This is fairly close to the average effectiveness of 84.84%, obtained with problems from the tension-7 problem space.

In view of the above analysis, we postulate that low IG performance is due to the following causes (some of which are interrelated).

1 frequent competition for tools;
this correlates with

1a high tension.

2 faltering eligibility heuristics, due to highly focused oversupply;
this correlates with

2a highly correlated agent preferences;

2b a highly eccentric community tool bag.

3 few tool alternatives (i.e., high `zero utility`)

4 high utilities for the tool alternatives (i.e., high `low alternative`)

7.4.2 An effectiveness of 92%: admirable or pityful?

When we discussed IG's performance on a sample from the default sample space, we remarked that it is not reasonable to expect that one can impose strong informational decentralization without incurring any detereoration of performance. We then qualified the average detereoration incurred by IG — from 100 to 92.45% — as "low". One can rightfully ask *on what grounds* such a label was attached. It is quite justifiable to speak of an effectiveness of 100% as "high", since it implies that a solution of maximal community utility was found. At first sight, it may also seem justifiable to label 99% effectiveness as "high", since the community utility of the rendered solution is very close to that of an optimal solution. Such a label can be misleading, however, because an assignment with an effectiveness of 99% *can* be the worst possible assignment. This would be the case, for example, if the problem that is solved involves a utility matrix with row maxima of 100, while all other entries are 99. In other words, except for effectiveness values of 0 or 100%, we need to know what the *problem* is, before we can evaluate the effectiveness that was obtained. To know whether 92% effectiveness is admirable or pityful, it is not enough to define a (problem-independent) ceiling of 100%, corresponding with an optimal assignment. We must also define a (problem-dependent) *floor*, some rock-bottom value designating the effectiveness which a decentralized algorithm should *at least* attain on the particular problem instance at hand.

As we noted earlier, the average initial effectiveness is too pessimistic to be a suitable reference level for the average final effectiveness to be obtained by IG, since it is computed from assignments in which no agent is fully satisfied. In this respect, the *tension* in the reassignment problem is a better candidate. Below, we define a reference level for the average effectiveness in a problem sample, based on the average tension in that sample.

If every agent proposes to exchange its current endowment for its first preference (as IG agents do in the first round of an auction), the tension equals the total undersupply. Hence, if the tension of a problem equals t, there exists an assignment in which $n - t$ agents possess their first preferences, while there is no assignment where more than $n - t$ agents attain maximal utility. This enables us to define a lower bound on the optimal community utility that is much sharper than the initial utility of a viable problem. The optimal community utility C^* for a problem $\langle e, U \rangle$ with tension t, and an agent community of size n is at least

$$C^{\perp} = \frac{n-t}{n} \cdot 100 \tag{7.6}$$

$C^{\perp}$ (pronounced as "C bottom") is a relatively *sharp* lower bound if the tension is low in comparison with the size of the agent population. Note, however, that — even with low tension — it is not always optimal to assign as many first preferences as possible.

As an example, consider the reassignment problem

$$\langle abc, U\rangle = \begin{pmatrix} \underline{0} & 9 & 10 \\ 10 & \underline{0} & 9 \\ 10 & 6 & \underline{0} \end{pmatrix} \tag{7.7}$$

The tension in this problem is 1, so it is possible to assign $3-1=2$ first preferences, either to agents 1 and 2, or to agents 1 and 3. This would lead to one of the assignments *cab* and *cba*, with respective community utilities of $\frac{26}{30} \approx 87\%$ and $\frac{20}{30} \approx 67\%$. In the optimal assignment *bca*, only agent 3 possesses its first preference, but the community utility is higher: $\frac{28}{30} \approx 93\%$. We have counted the number of problem instances in the default sample for which the computed optimal solution comprised less than $n-t$ completely satisfied agents. This appeared to be the case for about 40% (7,845 of 20,000) of the problem instances.

The quantity $C^{\perp}$ in Eq. 7.6 is not only a lower bound for the optimal community utility associated with a problem, but also a suitable reference value for the community utility that should *at least* be obtained by an informationally decentralized algorithm like IG. It expresses the portion of community utility that is due to agents who receive their first preference, if the number of such agents is as high as possible. Since agents in IG (as well as in other decentralized reassignment mechanisms, such as the Walrasian auction) convey their first preferences in the first round of bidding, it is quite easy to attain at least this community utility. All that needs to be done, is to grant as many of the first-round proposals as possible. For mechanisms which ensure individual rationality, this can be far less than $n-t$ (cf. Chapter 3), but for IG, it is always possible to satisfy $n-t$ proposals. Indeed, the bottom-level community utility $C^{\perp}$ is a lower bound for the community utility that will be attained by a *greedy* IG mechanism, such as one with zero `caution`. In the *default* IG mechanism, however, agents are risk-neutral instead of risk-insensitive. Hence, in this case, the community utility obtained by IG may turn out to be less than $C^{\perp}$ for some problems. This does not imply that the risk-insensitive variant of IG is superior, for the risk-neutral version performs much better on average. Because we are primarily concerned with the average performance, the *average* bottom-level community utility $\overline{C^{\perp}}$ in a sample space is a suitable rock-bottom reference for the *average* community utility that should be obtained by IG on that sample space.

Because the IGRT reports the average *effectiveness* attained by IG, we use the bottom-level community utility to obtain a reference level $E^{\perp}$ for the average effectiveness. Formally, if $C^{\perp}(s)$ and $C^{*}(s)$ denote the bottom-level community utility, and the optimal community utility associated with the problem instance s, the associated average bottom-level effectiveness $\overline{E^{\perp}}$ for a sample space S equals

$$\overline{E^{\perp}} = \frac{1}{|S|} \cdot \sum_{s \in S} \frac{C^{\perp}(s)}{C^{*}(s)}$$

Exact computation of $\overline{E^\perp}$ would require adaptation of the IGRT. Since the bottom-level community utility is a sharp lower bound in case of low tension only, this is not worth the effort. We therefore choose to compute a rough estimate of $\overline{E^\perp}$ instead. This *base-level effectiveness* $\widehat{E^\perp}$ for a sample space is defined in terms of the average tension $\bar{t}$ and the average optimal community utility $\overline{C^*}$ in the sample space.

$$\widehat{E^\perp} = \frac{\overline{C^\perp}}{\overline{C^*}} = \frac{\frac{n-\bar{t}}{n} \cdot 100}{\overline{C^*}} \tag{7.8}$$

In Fig. 7.13, one can observe an almost linear relation between the tension and the average optimal community utility in sample spaces associated with a *single* matrix. This suggests that $\widehat{E^\perp}$ is a reasonably accurate approximation of $\overline{E^\perp}$.

To compute the base-level effectiveness for the default sample space, we need to know the average optimal community utility and the average tension in that space. As apparent from Tables 7.2 and 7.6, the IGRT reports the average optimal community utility, but not the average tension. However, we can deduce the latter (for the default problem space) from the common endpoint of the C^*-curves in Fig. 7.1 on page 274. At 100% `zero utility`, agents can only contribute to the community utility if they obtain their first preference. Hence, in this case $C^* = C^\perp$. Thus, we can derive the average tension in the associated sample space from the value $C^* =$ 76.30% reported by the IGRT. This renders a value of 2.844 for the average tension in $S(\{3,3,2,2,2\},10,4,3,9)$. Since the IGRT chooses the first preferences of the agents independently from a uniform distribution on the tool types (cf. Sect. 6.5.4), the tension in a sample space depends *solely* on the multiplicity type of the tool bag. This multiplicity type is the same for all of the problem spaces associated with Fig. 7.1. Hence, the tension in the default problem space $\tilde{S}(\{3,3,2,2,2\},10,1,3,9)$ also equals 2.844.

Together with the reported value of 93.12% for C^* in the default problem space, this leads to a base level effectiveness of 81.94%. For the default homogemic sample space, which has a marginally higher average optimal community utility (93.27%), we arrive at an estimated base-level effectiveness of 81.81%.

Because it is always somewhat hazardous to rely on the randomness of pseudo-random number generator output, we have incorporated tension computation in a later version of the IGRT. This version reports average tension values for the default problem space of 2.85% (with rounding of the second decimal). From the raw output supplied by this IGRT version, we also estimated the distribution of tension in the default sample space, pictured in the leftmost graph of Fig. 7.14. In this case, we allowed the generation of (trivial) problem instances with zero tension, which are normally suppressed by the IGRT. It appears that such problems are rarely generated anyway. They comprised only 0.7% (144 of 20,000 problem instances) of the sample. The other graph in Fig. 7.14 pictures the tension distribution in the sample space $\tilde{S}(\{1:12\},10,1,3,9)$ of untyped problems. The average tension in this sample space is 4.22, considerably more than the average tension in the default (heterogemic and homogemic) spaces. The fact that the tension apparently increases if the number of tool types is increased (while the number of agents is not), confirms our earlier explanation why IG performs less well (in terms of effectiveness) on untyped problems than on typed ones (See Fig 7.6).

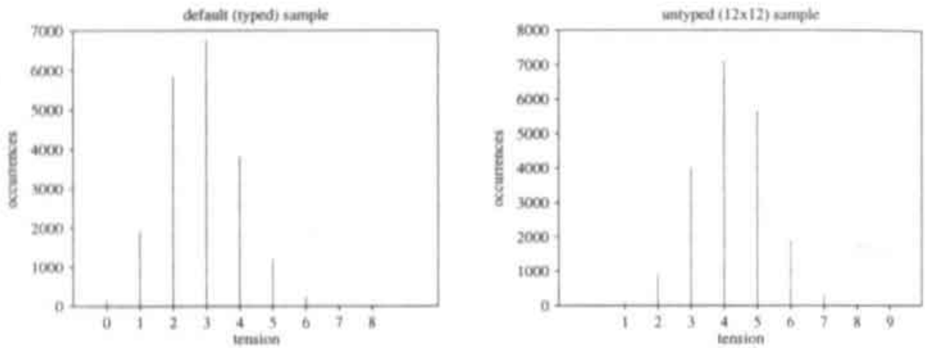


Figure 7.14: Distribution of tension in typed and untyped problem spaces.

The base-level effectiveness values of approximately 82% for the default (heterogemic and homogemic) sample spaces are useful for the discussion of the influences of agent and mechanism parameters in the next section. They allow us to conclude when an architectural variation brings about unacceptable detereoration of performance. Also, they give us some idea as to when performance differences between different variants of IG are negligible, and when they are not. Note, however, that, though base-level effectiveness is a much better reference for rock-bottom performance than the average effectiveness of viable assignments, it is by no means perfect. Because it reflects only the portion of community utility that can be contributed by agents that receive their favorite tool type, it can be overly pessimistic if there are few such agents (i.e., if the average tension is high), while there are many tool alternatives (i.e., if `zero utility` is low).

7.5 Effects of Varying Agent and Mechanism Parameters

In this section, we discuss the outcomes of the experiments performed to assess the influence of agent and mechanism parameters on IG performance. If not stated otherwise, performance statistics are obtained on (one and the same sample from) the default sample space $\tilde{S}(\alpha, 10, 1, 3, 9)$. This sample space contains reassignment problems with 12 agents, and 5 tool types. The tool bag is *aaabbbccdd*, and the utility matrices are heterogemic, with 1 or 3 zero entries per row, and one maximal entry (of 10) per row. The other entries are integer numbers between 3 and 9 (inclusive).

In most cases, we only vary one agent or mechanism parameter at a time. We make an exception for the boolean parameters `weariness` and `clearance`. These are varied in combination with the `deadline` parameter. Except in the histograms, confidence intervals are plotted around the data points in all of the plots. If a plot displays more than one set of data points, the points of each set are connected by lines to facilitate distinguishing between the sets.

7.5.1 The influence of `deadline` and `weariness`

Fig. 7.15 pictures the influence of the deadline and weariness parameters on the primary performance attributes `effectiveness` and `rounds`. All of the experiments that involve the deadline parameter comprise the settings `deadline` = 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, and 2000. With logarithmic scaling along the horizontal axis, these ordinates are approximately equidistant. The scaling along the vertical axis is usually linear. For brevity, we sometimes use D instead of `deadline`.

The first plot in the figure portrays the difference in the effectiveness obtained by IG with and without weariness. Moving along the two curves from right to left (i.e., from high deadlines to lower ones), we see that the curves run closely together at first, but split at $D = 20$. If we decrease the deadline further, this seems to incur only a slight decrease in the average effectiveness of IG without weariness, while the average effectiveness of IG with weariness drops dramatically toward about 31% at $D = 1$. In view of the base-level effectiveness of 82%, the value of 31% is dramatically low indeed. In fact, it equals the initial effectiveness in the default sample (cf. Table 7.2 on page 270). This is not a coincidence. With weariness incorporated, setting the deadline to 1 round has the effect that the weariness treshold is 100% from the start. In other words, no agent is ever permitted to deviate from its previous bid. Formally, there is no previous bid in round one. In the IGRT, however, the proposal profile is initialized to an array of empty proposals before the auction commences. Consequently, each agent is forced to submit an empty proposal in round one. This leads to a final assignment that equals the initial one.

With the deadline at 2 rounds, something similar occurs, though not to the same extreme. In round one, the weariness treshold is now 50%. This means that only those agents with an initial utility less than 50% are allowed to submit a nonempty proposal. Since the treshold reaches 100% in round 2, the other agents cannot improve on their initial utility. This leads to an average effectiveness which is higher than with $D = 1$, but still well below the base-level effectiveness of 82%, and hence, unacceptable.

Because of the extremely low value of 31%, the scaling of the first plot does not show

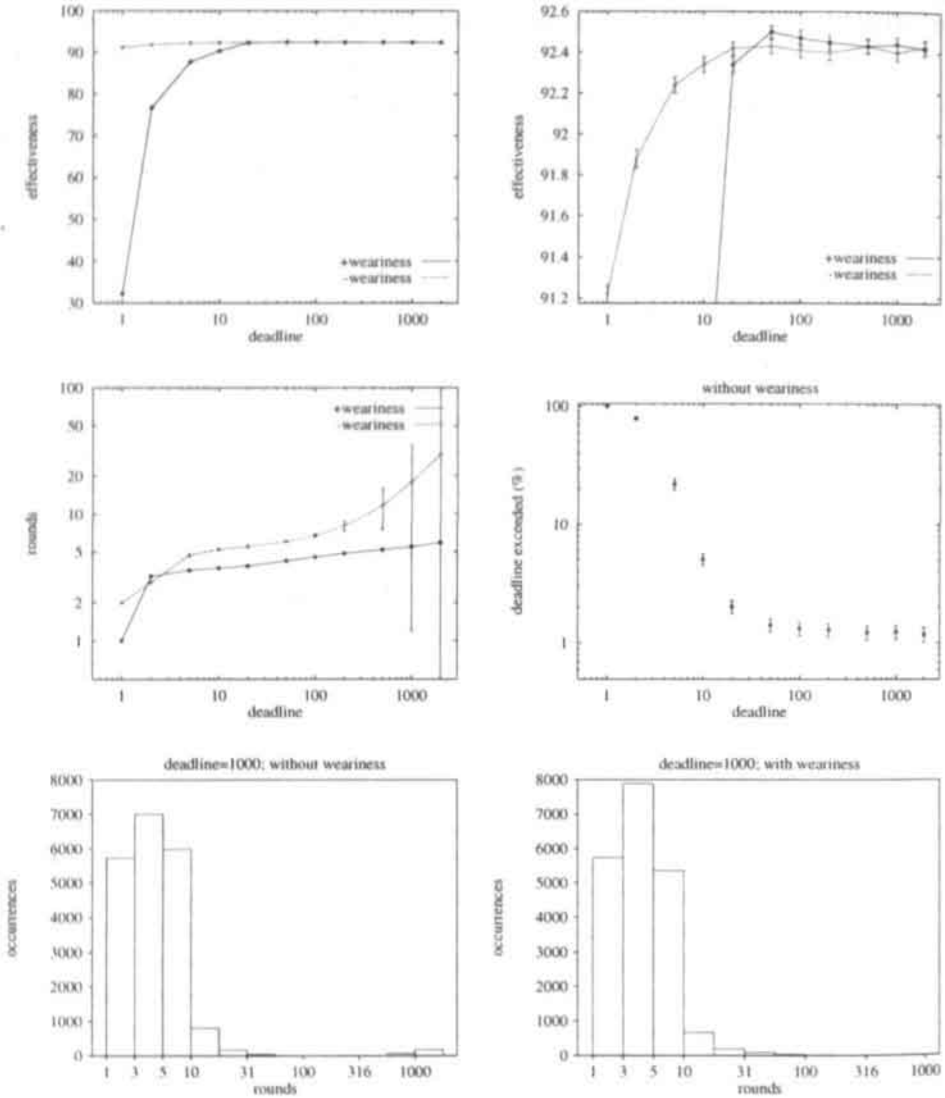


Figure 7.15: Influences of `deadline` and `weariness`.

clearly what happens at high values of D. The second (upper right) plot is the same as the first, except that it is scaled so as to provide a clearer view. It appears that, with a deadline of more than 20 rounds, the difference in obtained effectiveness of IG with and without weariness is marginal and not statistically significant. Also, once we have chosen a deadline of at least 50 rounds, there is little point in increasing it further, as far as the average effectiveness is concerned. However, the 'slight' decrease in average effectiveness that we observed in the leftmost plot (without weariness) when moving from a high deadline value toward $D = 1$ appears to be a considerable decrease after all.

From the third plot in Fig. 7.15, we conclude that, for large values of the deadline, the incorporation of weariness significantly reduces the average auction duration.[17] At `deadline` = 50 the reduction is still moderate (from 6 down to 4.2 rounds), but at `deadline` = 2000 it amounts to a factor 5 (from 30 rounds down to 6 rounds). However, the confidence intervals in the plot indicate that the averages computed for IG without weariness are very unreliable if the deadline is high. In fact, the confidence intervals themselves are unreliable. This is due to the fact that, without weariness, the deadline is sometimes reached. As the 4th plot shows, this does not happen very often, but if the deadline is high, it does imply that the distribution of the number of rounds — shown in the 5th plot of Fig. 7.15 — is too skew to satisfy the near-normality condition of our confidence limit computation (cf. Sect. 6.5.4, p. 256).

Histograms of the distributions underlying the data points at `deadline` = 1000 are shown in the 5th and 6th plot. The occurrence counts were performed on value categories between low (inclusive) and high (exclusive), as indicated along the horizontal axes. Thus, the first bar in the histogram indicates that there were approximately 5800 problem instances in the sample with an auction duration of 1 or 2 rounds, the second bar represents some 7000 instances which required 3 or 4 rounds, etc. Comparison of the 5th and the 6th plot shows that the difference in the average effectiveness between IG with and IG without weariness is primarily due to the fact that auction durations of 1000 or more rounds occur only if weariness is not incorporated.

Fig. 7.16 depicts the frequency and impact of weariness. As expected, weariness concessions occur frequently if the deadline is low. They also involve considerable concessions on the part of the agents in this case. If the deadline is set to a sufficiently high value, however, weariness seldom plays any role in the agents' decision making, and if it does, the associated utility concessions tend to be marginal.

Fig. 7.17 pictures the frequency and severity of utility loss. If we take into account the conclusion drawn from the effectiveness plot (Fig. 7.15), that the deadline should be 50 rounds or more to attain an acceptable average effectiveness, Fig. 7.17 is not very interesting. The percentage of zero-loss auctions at $D = 50$ differs only marginally (less than 2%) from any value at a higher deadline, and the same is true for the highest utility loss. Yet, the setting $D = 2$ without weariness is interesting,

[17] To interpret the curves properly, one should be aware that the scaling of this plot is logarithmic on *both* axes.

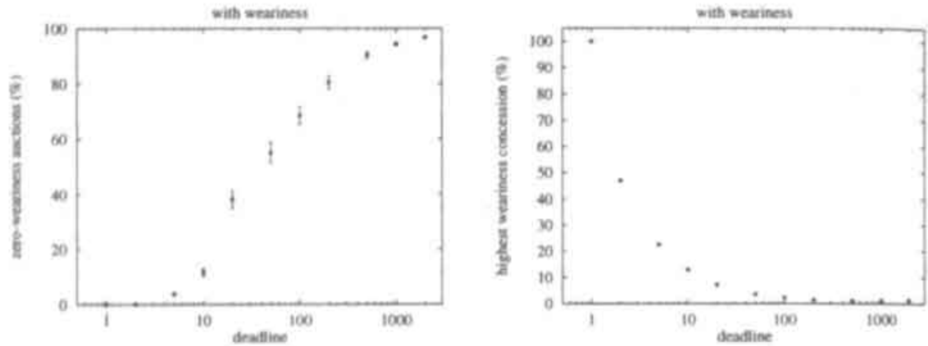


Figure 7.16: Frequency and severity of concessions due to weariness.

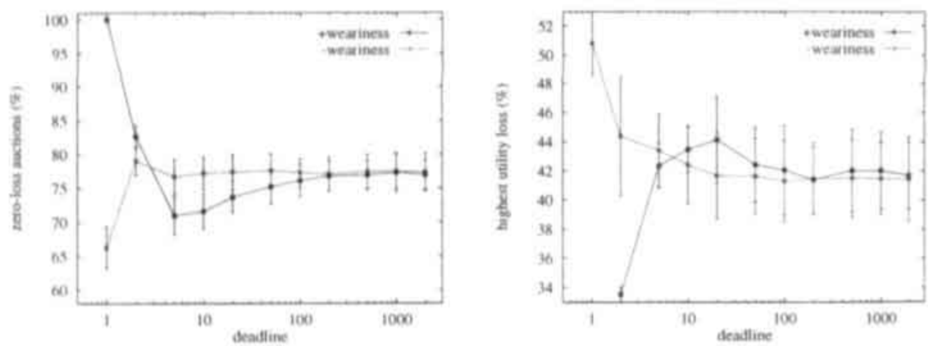


Figure 7.17: Frequency and severity of utility loss.

because the associated IG mechanism amounts to a reassignment version of delegated negotiation (See Sect. 5.5). This setting renders a slightly higher percentage of zero-loss auctions and a lower auction duration at the expense of some (0.6%) effectiveness.

setting		eff.	rounds	ZL	PL	P_l	HL
D	W						
500	TRUE	92.4	5.19	75.1	8.73	2.17	42
50	TRUE	92.5	4.22	76.9	8.80	2.03	42
2	FALSE	91.9	2.87	79.1	8.68	1.81	44

Table 7.9: Tradeoffs between `zero loss` and effectiveness.

Table 7.9 compares the relevant statistics of $D = 500$, $D = 50$, and $D = 2$ without weariness. W stands for weariness, and eff. for effectiveness. The semantics of the symbols ZL, PL, and P_l are *zero-loss*, *percentage of losers*, and *probability of loss* (cf. Table 7.5 on page 291). HL stands for *highest loss*.

It is conceivable that the scale of the problem has some bearing on the best value for **deadline**. One would, for example, expect that a relatively low value of **deadline** which still renders an acceptable solution quality for small-scale problems, may well prove too low to ensure the same on problems of a larger scale. To verify this, we have also investigated the influences of the **deadline** parameter on problems with 20 agents and 9 tool types. The results, shown in Fig. 7.18, point out that the above intuition is incorrect: With 20 agents and 9 tool types, the effectiveness curve levels off at the same deadline value (viz. $D = 20$) as with 12 agents and 5 tool types.

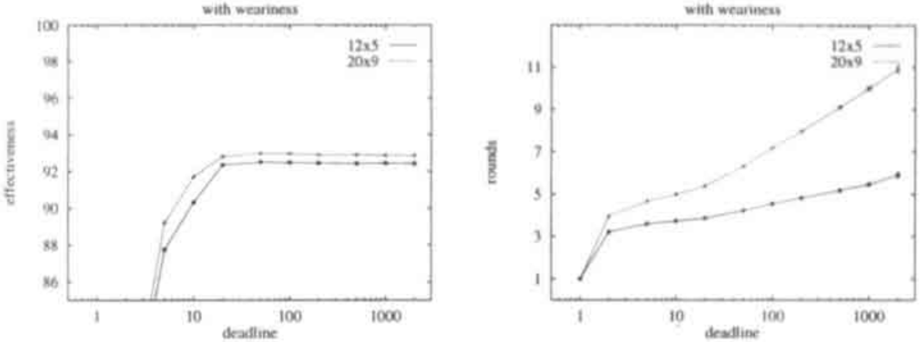


Figure 7.18: Deadline influence at different problem dimensions.

7.5.2 The influence of the `clearance` parameter

The motivation to consider market clearance as an additional sufficient condition for phase termination — next to proposal convergence and deadline excess — was the intuition that this could possibly increase the average effectiveness and decrease the average auction duration.

In Fig. 7.19, we see that the above intuition was principally correct, but the effects are marginal. The first curve shows no effectiveness differences at all. If we zoom in on the curves by rescaling from 92% upwards (in the third plot), it appears that the differences are virtually never[18] statistically significant, and amount to less than 0.1% for all values of D. The second curve shows a statistically significant, but equally marginal decrease of the average auction duration if market clearance is incorporated. The fourth curve explains why incorporating market clearance has little impact: It is seldom instrumental.

[18] At $D = 500$, one can observe a barely significant difference.

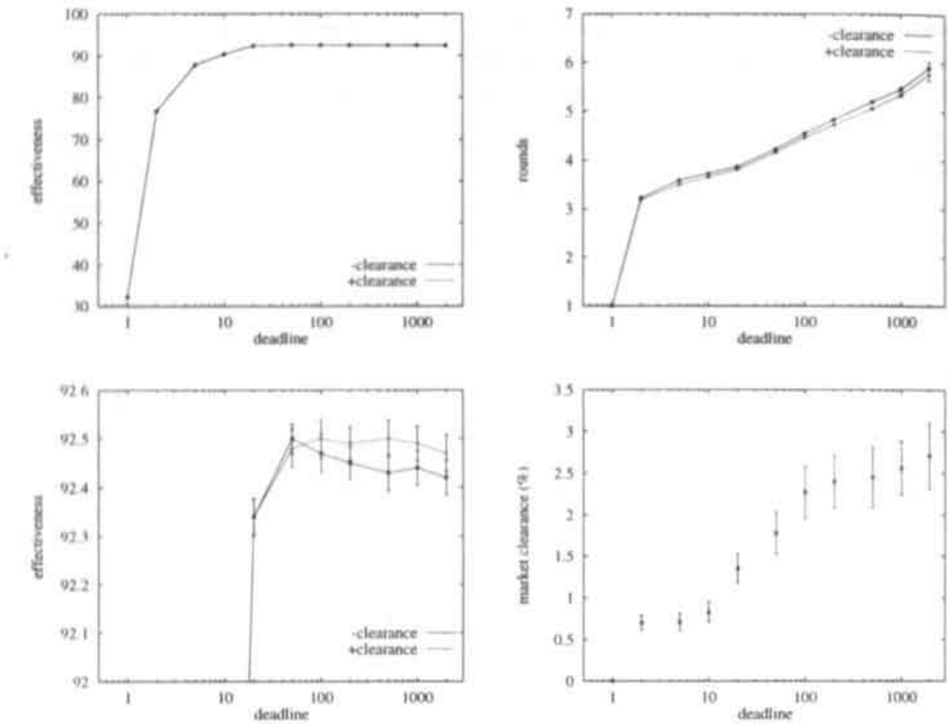


Figure 7.19: The impact of incorporating market clearance.

Because market clearance adds to the conceptual complexity of Informed Gambling, these findings strongly suggest to leave it out.

7.5.3 The influence of the `resolution` parameter

The resolution parameter determines whether the eligibility heuristic of IG features fine or course distinctions. High resolution implies that proposals differ in eligibility whenever the scarcities of the offered tool types differ. Low eligibility implies that it matters *whether* the offered tool type is scarce or oversupplied, but not *how* scarce or oversupplied it is. Mixed resolution implies that the degree of scarcity does matter if the tool type is scarce, but not if it is oversupplied.

In Sect. 6.2.3, we motivated the omission of the converse case (distinction between different degrees of oversupply, but not between different degrees of scarcity) by stating that, in a reassignment problem, an agent can adapt its demands in various ways while there are only two options for its supply: It either supplies its initial endowment or withdraws, not supplying anything. It is not immediately clear whether the provision of an incentive to an agent to change its demand from a tool type that is strongly undersupplied to one that is only slightly undersupplied can have a positive effect on mechanism performance, but in any case the agent is *capable* of such a relaxation. This is not the case for an incentive to relax its supply from a tool type that is strongly oversupplied to one that is slightly oversupplied, at least not in reassignment problems, where each agent can supply only one tool type.

In view of the above considerations, one would expect that the setting of the resolution parameter would have a considerable impact on IG performance, and that mixed resolution would lead to the highest performance. Fig. 7.20 shows that the latter is true, but the former is not.

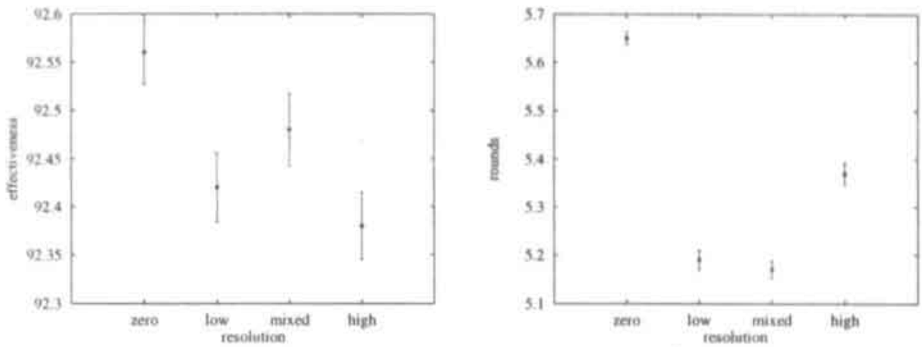


Figure 7.20: Influence of the `resolution` parameter.

The first plot in Fig. 7.20 shows that the difference between the effectiveness levels obtained with low, mixed and high resolution is marginal: in the order of magnitude of 0.1%. The second plot shows that there is hardly any difference between the average auction duration with low and mixed resolution, but high resolution incurs significantly longer auctions.

Next to low, mixed and high resolution, the plots in Fig. 7.20 also picture the average effectiveness and auction duration obtained with zero resolution. Zero resolution amounts to disabling the eligibility heuristic altogether. Originally, this setting was not in the value range of the resolution parameter. It was conceived as a refinement of the base-level effectiveness, which tends to be too low as a reference value in case of high tension, because it only counts the contribution of first preferences to the community utility. IG with zero caution and zero resolution constitutes an almost trivial mechanism, where the agents blindly strive for their first preference, and the auctioneer greedily satisfies as many proposals as possible. As such, I expected it to be an improvement of the base-level effectiveness, that would provide a *sharper* lower bound for the average effectiveness that an IG mechanism should attain at the very least.

It was a bit of a shock to see how sharp this 'lower bound' really was. Apparently, the eligibility heuristic has an *adverse* effect on the average effectiveness!

This does not imply that the eligibility heuristic has no virtue whatsoever. As the second plot in Fig. 7.20 shows, it incurs a decrease of the average auction duration (from 5.7 to 5.2 rounds), at the expense of a barely significant decrease of the average effectiveness (of about 0.1%). Furthermore, the heuristic does not always have an adverse effect on solution quality. For some other sample spaces (e.g. with a highly eccentric tool bag), the average IG effectiveness is higher with the heuristic than without it. We discuss such cases later, in Sect. 7.7.2.

7.5.4 The influence of the `asynchrony` parameter

In Sect. 5.7, we introduced asynchronous proposal processing as a first attempt to guarantee the termination of the adjustment process. An example showed that asynchronous processing, by itself, cannot guarantee proposal convergence in finite time. The incorporation of negotiation weariness can provide such a guarantee.

Since the incorporation of weariness guarantees proposal convergence at or before the deadline, irrespective of whether proposals are processed synchronously or not, the question arises whether asynchronous processing is still preferable over synchronous processing, once negotiation weariness is incorporated. Also, earlier research on Boltzmann neural networks (Lenting, 1992b) has pointed out that the information backlog inherent to unconstrained parallel processing can cause a considerable detereoration of solution quality if the deadline is low, and a formidable increase in the duration of the computation is the deadline is high. Because the functionality of weariness in IG is akin to that of the temperature in Boltzmann machines, information backlog may well play an important role in IG as well.

Fig. 7.21 pictures the influences of the `asynchrony` parameter. The percentages along the horizontal axis (starting at 0) correspond with market profiles being computed on the basis of 12, 11, ..., 2, and 1 submitted proposal(s). As such, the investigated parameter range covers all cases.

From the first plot in Fig. 7.21, we conclude that the asynchrony setting has no bearing on the average effectiveness, except that synchronous bidding lead to a marginally higher (+0.1%) effectiveness. However, this gain is obtained at the expense of a large increase of the average number of rounds (the second plot), accompanied by an equally pronounced decrease of the percentage of zero-weariness auctions. Apparently, negotiation weariness is unable to keep the average auction duration down to an acceptable level in case of synchronous bidding.

We have designated the number of rounds as the metric of choice for the auction duration. This is fine in most cases, since the number of rounds is usually proportional to the number of processed proposals. However, if we vary asynchrony, we vary the

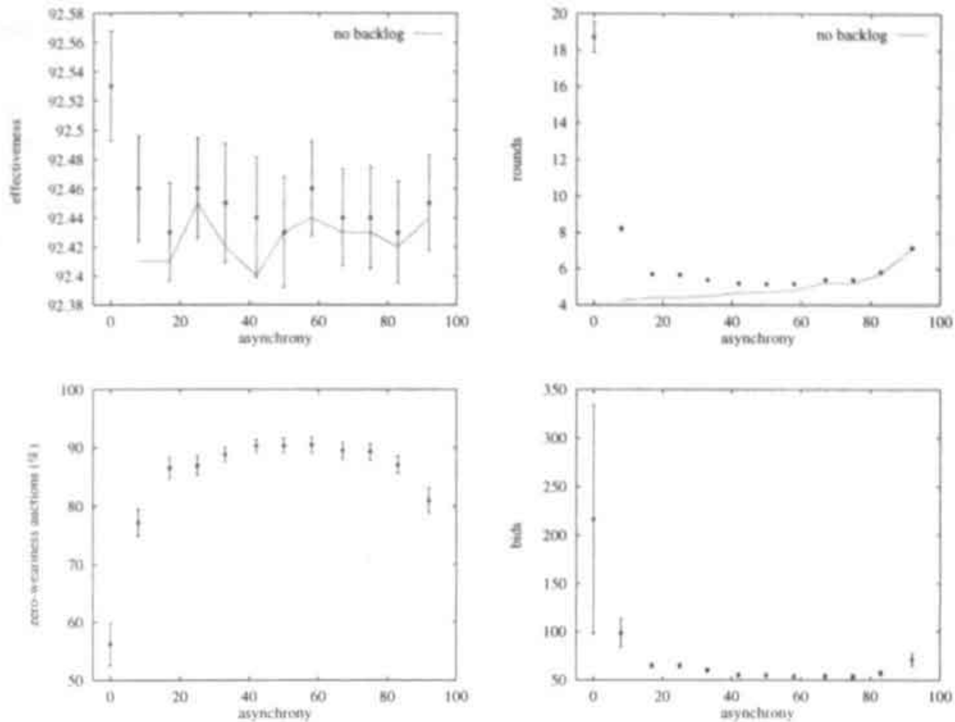


Figure 7.21: Influence of the **asynchrony** parameter.

number of processed proposals per round. The amount of *real time* consumed by an auction that takes ten rounds to complete depends on the number of processed bids per round, and hence, on the value of **asynchrony**. With **asynchrony** = 50% the auction can involve twice as much real time as with **asynchrony** = 75%, if the auctioneer is slow or if there is a large variation in the speed of agent response to a market profile.

Because of this, a fourth plot was added to Fig. 7.21, which pictures the auction duration in bids. The general pattern is in this plot is surprisingly similar to that in the rounds plot. In both cases, there is no significant difference between the averages obtained with different asynchrony settings, as long as the asynchrony is moderate (say, between 40 and 75%). These settings are also better than the more extreme values in both cases. A plausible explanation for the apparent similarity is that, though the required number of bids per round is low at high values of **asynchrony**, this does not necessarily mean that the effective number of bids is also low. Whenever the proposal profile in some round equals that in the previous round, the auctioneer requests all of the agents to submit a proposal to test whether the phase can be terminated because of proposal convergence. This is bound to happen much more often if there are only few responses initially.

Another explanation is provided by the no-backlog curves in the topmost two plots. These depict the average effectiveness and auction duration that would have been obtained in the absence of information backlog. This comes down to sequential bidding, except that proposals are submitted by a random subset of the agents, in a random order. As apparent from the second plot, information backlog has a considerable adverse effect on the auction duration. The first plot shows that the average effectiveness is not adversely affected by information backlog, however.

7.5.5 The influence of the `caution` parameter

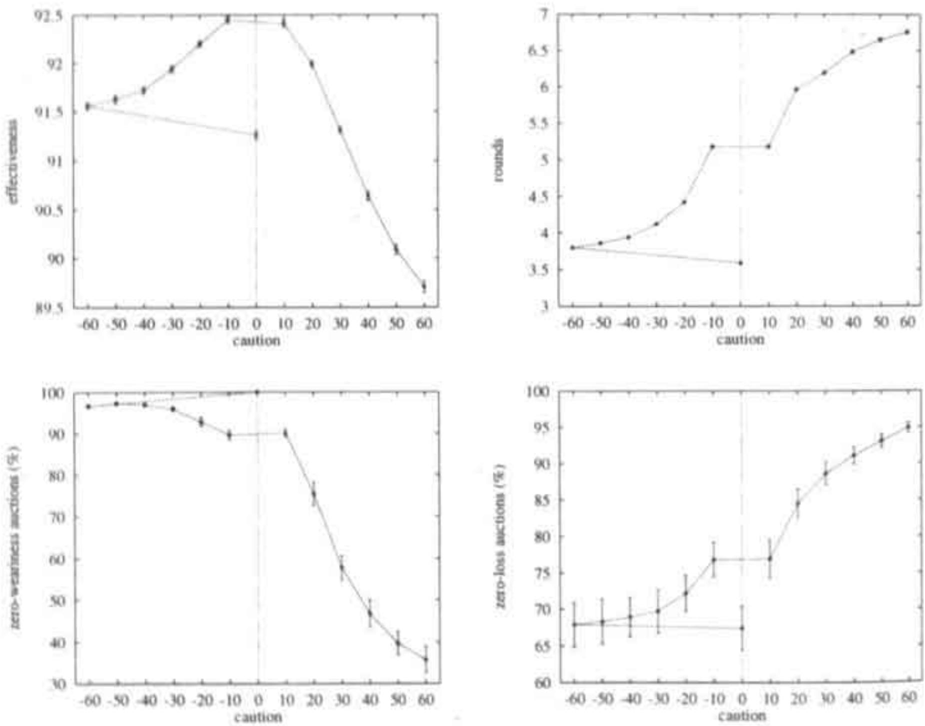


Figure 7.22: Influence of the `caution` parameter.

The influences of the caution parameter setting are shown in Fig. 7.22. The data points in the plots are connected by a line that indicates the conceptual order among caution values. If the line were omitted, the reader might be tempted to view the setting `caution` = 0 as an intermediate setting between `caution` = −10 and `caution`

= 10. This is technically correct, but conceptually wrong. The `caution` values −10 and 10 represent the same setting, and with zero caution, agents are *less cautious* than with `caution` = −60. Henceforth, we speak of "increasing caution" if the agents become *conceptually* more cautious. As such, changing `caution` from 0 to −60 will be referred to as *increasing* caution.

From the first plot in Fig. 7.22, it appears that, if we start at `caution` = 0 and move toward more cautious agents, the average effectiveness increases relatively slowly, attaining its maximum at the default setting of `caution` = 10. Thereafter, the effectiveness decreases quickly. The auction duration (in the second plot) appears to increase monotonically with increasing caution. So does the percentage of zero-loss auctions (in the fourth curve). The percentage of zero-weariness auctions decreases with increasing caution.

Upon introducing the caution parameter in Sect. 6.2.4, we defined the caution parameter in the testbed interface as the integral part of the actual, internal caution (the exponent C in Eq. 6.5 on page 241) *multiplied by a factor 10*. We motivated this detour by stating that the distortion of risk perception associated with $C = 2$ (i.e., caution = 20) is likely to be too gross for `caution` = 20 to be a profitable setting. A more modest distortion such as that associated with $C = 1.1$ seems more promising. In view of this argument, the experimental results pictured in Fig. 7.22 are insufficient to conclude that risk-neutral agents perform best. We need to experiment with caution on a finer scale. The outcomes of such experimentation is shown in Fig. 7.23, for caution values of 10, 11, 12, 13, 14, 15, 17, 20, and their negative counterparts.

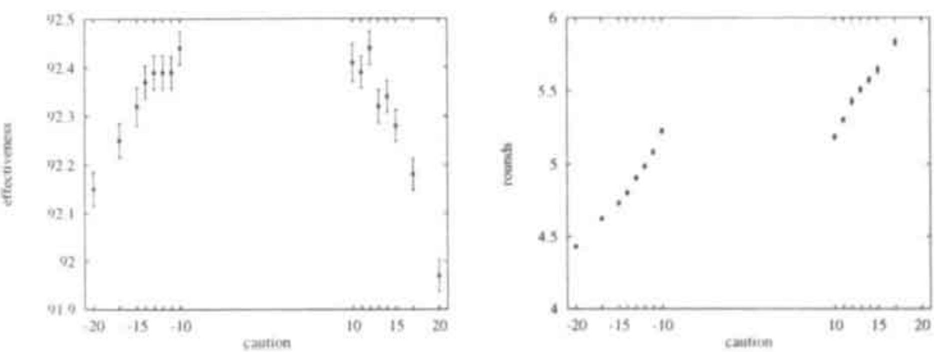


Figure 7.23: Fine tuning of the `caution` parameter.

Inspection of Fig. 7.23 shows that the preliminary conclusions, drawn on the basis of Fig. 7.22 are held upright: the highest average effectiveness is obtained with risk-neutral agents, and the average auction duration increases with increasing caution.

7.5.6 The influence of the maxloss parameter

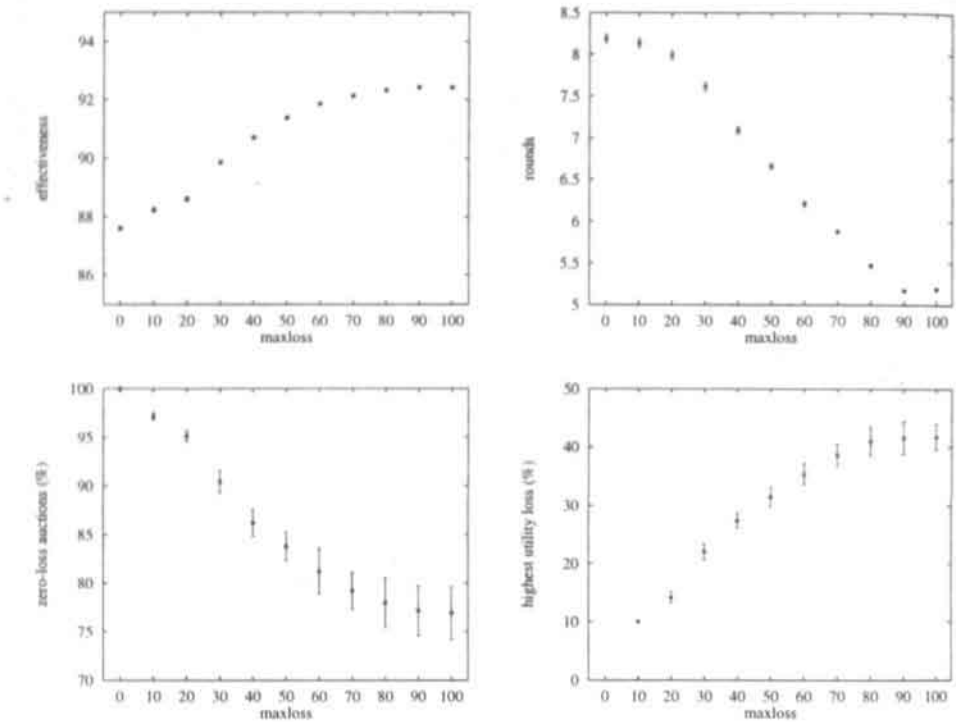


Figure 7.24: Influence of the **maxloss** parameter.

The maxloss parameter is related to the caution parameter in that both describe aspects of the agents' attitudes toward risk. As such, both parameters can be used to increase the percentage of zero-loss auctions. We have seen that we can attain a percentage of at least 95% zero loss, if we set the caution to 60. However, there is a tradeoff with the effectiveness of IG. Decreasing the probability of utility loss is accompanied by a considerable decrease of effectiveness.

As apparent from Fig. 7.24, the effectiveness decrease is even higher if we use the maxloss parameter to reduce the (frequency of) utility loss. On the other hand, the maxloss parameter offers the opportunity to *guarantee* that the reallocation is individually rational (by setting it to 0%).

If **maxloss** $= x > 0$, no agent can suffer a utility loss of more than x% (in terms of normalized agent utility). The fourth curve in Fig. 7.24 shows that, except for **maxloss** $= 10$, the highest utility loss is actually less than x.

7.5.7 The cost of demanding individual rationality

The `caution` and `maxloss` parameters constitute alternative means to reduce — or, in the case of zero `maxloss`, prevent — utility loss in IG. As we have seen in the previous two sections, this does involve some cost in terms of allocation effectiveness. The average decrease in effectiveness induced by demanding the solution to be individually rational (by setting `maxloss` to zero) is somewhat over 4%. It is interesting to compare this with the 'theoretical' minimum for this average cost: the average effectiveness difference between optimal individually rational allocations and optimal allocations. For the default sample space, this turns out to be as little as 0.36%![19]

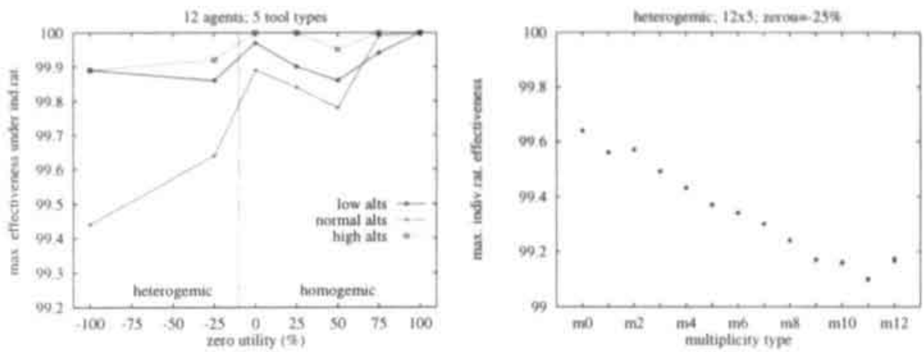


Figure 7.25: The cost of individual rationality *without* (strong) informational decentralization.

As Fig. 7.25 shows, this minimal cost of individual rationality is low in other sample spaces also: The average effectiveness of optimal individually rational allocations was never below 99% in any of our experiments. Apparently, individual rationality is a much more cumbersome constraint for IG than it is for centralized algorithms, or weakly decentralized ones such as a *monetary* Walrasian auction. We shall see later that the Walrasian *exchange* auction (which — like its monetary sibling — guarantees individually rational solutions, but — unlike the monetary auction — is *strongly* decentralized) attains an average effectiveness that is comparable to IG with the `maxloss`= 0 setting. This suggests that strong informational decentralization can be held responsible for the relatively high cost of individual rationality in IG. In other words, the cost of demanding individually rational solutions may well be high for *all* mechanisms featuring such decentralization.

[19] See line 2 of Table 7.2 on page 270.

7.5.8 The influence of the `volatility` parameter

The purpose of volatility is to lure the agents into *weakly altruistic behavior*. The motto is "Love thy neighbor if it does not cost thee much". Sometimes agents can do other agents a big favor, if they opt for a tool type with the *same* (or almost the same) utility as their current endowment, when they would normally withdraw from the auction because of lack of prospects of utility gain. The altruism brought about by positive volatility is *emergent* altruism: Positive volatility does not reduce the agents' self-centeredness. It works by distorting the agents' perception. The agents are made to perceive their endowment as less useful than it actually is. We pointed out earlier that such a distortion of agent perception is bound to be harmful to the community utility of the final assignment, if the depreciation of the endowment is large. It may lead to a higher community utility, however, if the depreciation is small.

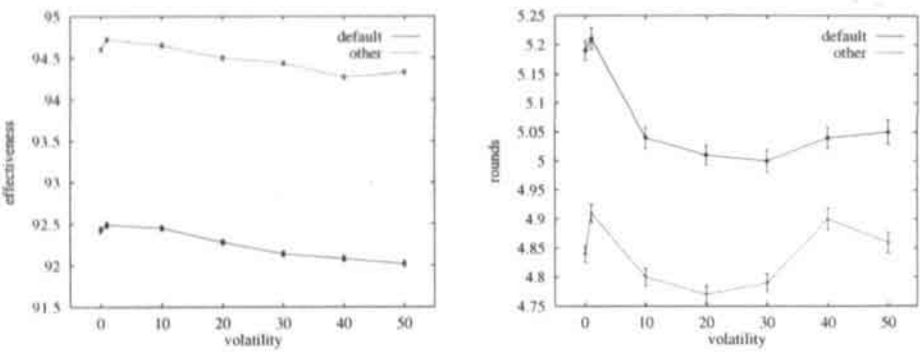


Figure 7.26: Influence of `volatility` on different sample spaces.

Fig. 7.26 shows that this intuition is correct, but the effectiveness gain incurred by a small, positive volatility is marginal. The two plots in Fig. 7.26 contain two curves, which correspond with different sample spaces. We discuss the default curves (i.e., those associated with the default sample space) first. For brevity, we use the symbol V for `volatility`.

The simulations are performed with volatility values of 0, 1, 10, 20, 30, 40, and 50%. Although the average effectiveness is indeed higher at $V = 1$ than at $V = 0$, the difference is not statistically significant. Moreover, the two effectiveness scores are very close together (92.43% at $V = 0$, and 92.49% at $V = 1$). So, in terms of effectiveness gain, the advantage — if any — of positive volatility is slim.

In the second plot, we observe a similarly minute difference between the average auction duration at $V = 0$ and $V = 1$. However, the average number of rounds drops considerably if we move to $V = 10$, while the average effectiveness at $V = 10$

is approximately the same (92.45%) as at $V = 0$. It appears that positive volatility does not significantly improve the quality of the solution, but it can reduce the average auction duration at no expense in terms of solution quality.

Because the default curve in the first plot reveals *some* gain in average effectiveness with small, positive volatility, it seemed worthwhile to investigate whether the gain is insignificant for *any* kind of reassignment problems. Obviously, it is best to do this on a problem space that offers good prospects. The curve labeled "other" in Fig. 7.26 corresponds with the sample space $S(\{3,3,2,2,2\},3,2,1,2)$. The parameter setting of this sample space is that of the default homogemic problem space, except that `utility range` = 3, `low alternative` = 1, and `high alternative` = 2. Since equal entries in the same row occur more frequently in this setting than in the default setting, there are generally more opportunities for agents to do some other agent a favor at no expense.

As the first plot shows, the difference between the average effectiveness at $V = 0$ and $V = 1$ is statistically significant with this other sample space. However, it is still marginal (94.6% at $V = 0$ and 94.7% at $V = 1$). As with the default sample space, the average auction duration can be reduced (at no expense) by setting V to 10%, but the reduction is less impressive than that obtained with the default sample space.

7.6 Performance Variation due to Nondeterminism

The analysis of the influence of the deadline parameter has made clear that IG performance can vary considerably between different problems in the same sample space (See Fig. 7.15). Hence, if a parameter setting X renders a better average performance than setting Y (in whatever respect) on some sample space, this does not imply that X leads to better performance than Y for *all* problem instances in that sample space.

The analysis in Sect. 7.4 has shown that there can even be considerable variation in performance, if IG is repeatedly invoked on the *same* problem instance, due to the nondeterministic nature of IG mechanisms. This is unpleasant, since it implies that low performance is not necessarily confined to difficult problems. It can simply be a matter of *bad luck*.

An example was provided in Table 7.8 on page 299. This table concerns the two problem instances associated with the lowest IG effectiveness values observed in any of the sample-space experiments: 55 and 60%. Repetitive application of IG rendered an *average* IG effectiveness for these problem instances of 65 and 73% respectively. This is still low, but considerably higher than the original performance lows.

Of course, repetitive application could be used to select the best solution in a sequence of trials. This would surely enhance the average IG performance. However, the associated mechanism is not informationally decentralized. Hence, picking the best solution from a series is only a viable option if informational decentralization is not a hard constraint. It could, for example, be imposed to tackle a constrained optimization problem which is too complex to be solved efficiently by means of a centralized algorithm. Obviously, this is never the case for reassignment problems. In this case, the only conceivable reason for informational decentralization is privacy, or some other *hard* constraint which precludes certain information from being gathered by a central system component.

Hence, we would like to know where the variance due to nondeterminism comes from, to see if we can reduce it (without adversely influencing the average performance). There are three potential sources for such performance variance.

the decisions taken by the auctioneer: In case of undersupplied tool types, the auctioneer employs the pseudo-composition protocol to decide which of the competing proposals will be satisfied. For an undersupplied tool type with supply k, the auctioneer's decision is deterministic iff, among the proposals *requesting the tool type, there are k proposals that are more eligible than* any of the others. Obviously, both the frequency of and the variance due to nondeterministic decisions depends on the value of the resolution parameter. With zero resolution, the decision is always nondeterministic, and any agent that has requested the tool type is a candidate to receive it. At the other end of the spectrum, high resolution will more often lead to deterministic solutions than mixed resolution, and if the solution is nondeterministic the number of candidates will often be smaller.

the decisions taken by the agents: If agents happen to have two or more options with maximal expected utility, they choose one of these at random. The frequency of such nondeterminism is governed primarily by the problem generations parameters `low alternative` and `high alternative`. If these are wide apart, the agent utilities for different (useful) tool types will all differ in the majority of generated problem instances. And if they are all different, nondeterministic agent decisions will be very rare.

asynchronous proposals processing: If the auctioneer employs asynchronous proposal processing, the identity of the agents that contribute to the update of the market profile varies nondeterministically. In case of synchronous proposal processing, this source of variance is absent. Hence, the asynchrony parameter is the relevant parameter in this case.

The prospects for reduction of the performance variance due to nondeterminism depend on the extent to which the variance stems from any of the above three sources. If the variance would appear to be mainly due to the nondeterministic decisions of the auctioneer, we could try to reduce the frequency of such decisions by adding a "very-high" resolution value. One possibility to achieve this would be to have the auctioneer keep track of the average demand and supply of tool types over the last k rounds, and use this as a tie break in case of equal eligibility. This would not reduce the frequency of nondeterministic decisions to zero,[20] but it would most likely reduce it, and have no undesirable side effects.

If the bulk of the variance is due to one or both of the other sources, the prospects are not as good. The only agent parameter that can be used to reduce the frequency of nondeterministic decisions by agents is the maxloss parameter, but lowering `maxloss` comes down to reducing the number of options. As we have seen in Fig. 7.24, this incurs a considerable decrease of the average effectiveness. The use of synchronous proposal processing to combat the third source of variance has a similar drawback. We know from Fig. 7.21 that this incurs a considerable increase of the average auction duration.

To assert the relative influence of the three sources of performance variation, we performed systematic fixed-instance simulation[21] on the default sample space, for various settings of the resolution and asynchrony parameters. We then computed, for each of these settings, the average number of different solutions per problem instance, and the range of incurred effectiveness variation. The outcomes are listed in Table 7.10. Because the simulation involved 200 invocations of IG to each of 100 different problem instances, the values in the table are the averages of only 100 numbers. Consequently, they are much less reliable than the statistics rendered in other experiments. This is illustrated by the fact that the reported average effectiveness, listed in the last column of the table, is slightly higher with the default IG mechanism (in the first row) than with resolution set to zero (in the fifth row), while we know from Fig. 7.20 that the converse is true for the sample space as a whole.

In view of the relative unreliability of the data, and the absence of confidence limits, we should be careful in drawing conclusions from Table 7.10. One can conclude, however, that there is not much difference between the degree of nondeterminism (the 4th column) and the incurred variation in effectiveness (the 6th column) between the first and the third, or the second and the fourth row. In other words, the choice between mixed and high resolution has little impact on the degree of nondeterminism and the incurred performance variance. Comparison of the relevant columns of the first and second rows, and those of the third and fourth shows that the asynchrony setting is more influential. The highest degrees of nondeterminism and effectiveness variance occur in the bottom two rows, which pertain to IG mechanisms that do not make use of any eligibility heuristic.

[20] Agents endowed with the same tool type will still be equally eligible.

[21] See the survey of simulation modes on page 248.

asynchrony	resolution	avg. nr.sols	max. nr.sols	avg.eff. range	max.eff. range	avg.eff
50%	mixed	5.0	36	5.8	20.0	92.8
0%	mixed	2.8	10	3.6	20.0	92.7
50%	high	4.9	24	5.6	20.0	92.9
0%	high	3.0	24	3.7	20.0	92.9
50%	zero	18.5	99	11.1	29.2	92.5
0%	zero	11.0	90	9.0	29.2	92.8

Table 7.10: Impact of asynchrony and resolution on nondeterminism.

All in all, the prospects of variance reduction do not seem to be very good. The variance reduction incurred by using high instead of mixed resolution is so small in comparison with that with using synchronous instead of asynchronous proposal processing that the addition of a "very-high" resolution (as suggested above) does not seem worthwhile.

7.7 Performance Comparison with Other Mechanisms

In this section, we compare IG with other strongly decentralized reassignment mechanisms, with respect to performance (in terms of allocation effectiveness), and also with respect to informational and operational decentralization.

In previous sections, we have discussed a number of mechanisms for decentralized reallocation. Some of these concern newly proposed approaches (e.g., delegated negotiation), while others are existing frameworks (viz. CDN (Sathi & Fox, 1989), and MOP (Wellman, 1994b)). These two frameworks are specifically aimed at reallocation problems that go beyond reassignment. Because we confine ourselves to reassignment, the performance comparison in this section does not involve these frameworks in their full capacity. We compare IG performance with *reassignment* mechanisms that reflect the ideas behind these general frameworks.

For MOP, this means that we investigate Walrasian *exchange* auctions instead of MOP mechanisms based on General Equilibrium Theory, which involve money, and production as well as consumption (such as Wellman's (1994b) WALRAS mechanism). For CDN (See Sect. 3.2.1), it means that we evaluate Top-Trading-Cycle (TTC) algorithms for elementary composition. Other components of CDN either do not make sense in the context of reassignment (e.g., reconfiguration), or are not described in sufficient detail in (Sathi & Fox, 1989) to produce an equivalent implementation (e.g., the interleaving of composition and relaxation).

Next to TTC mechanisms and the Walrasian exchange auction, the comparison also

mechanism	description
	Individually rational mechanisms:
WALRAS	computes a Walrasian assignment (if it exists)
$\underline{\text{T}}\text{TC}$	the default, nondeterministic Top-Trading-Cycles algorithm
$\underline{\text{T}}\text{TC}^{+}$	TTC + largest-cycle-first heuristic
$\underline{\text{T}}\text{TC}^{\bullet}$	TTC + largest-cycleset-first heuristic
$\underline{\text{T}}\text{TC}^{\forall}$	computes *all* TTC-solutions
	IG-related mechanisms:
IG	the default IG reassignment mechanism
$\underline{\text{I}}\text{G}_0$	IG with `caution = 0`
$\underline{\text{I}}\text{G}^{\nabla}$	IG without the eligibility heuristic
$\underline{\text{I}}\text{G}_0^{\nabla}$	IG^{∇} with `caution = 0`
DN	IG with `deadline = 2`, `weariness = FALSE`, and `asynchrony = 0`

Table 7.11: Reassignment mechanisms.

involves some mechanisms closely related to IG, such as delegated negotiation. A concise description of each of the mechanisms is provided in Table 7.11. In the table, we have separated the mechanisms that ensure individual rationality from those that do not.

WALRAS Our WALRAS algorithm computes a Walrasian assignment (if it exists) by means of stratification, as described in Sect. 4.2 and 4.4. Note that this is not the WALRAS framework described in (Wellman, 1994b), nor a reassignment version thereof.

TTC The basic Top-Trading-Cycles algorithm is described in Sect. 3.3.5 (near page 73). A decentralized version of this TTC algorithm comprises the following sequence of instructions (to be performed by the auctioneer).

1. Ask the agents to specify their endowment and their first preference, and compute tool type supply.
2. Communicate the current tool type supply to the agents in the audience.
3. Ask these agents to specify their first preference *in the current supply*.
4. If there is no TTC that involves at least two agents, finish.
5. Pick a TTC that involves at least two agents, perform a (partial) reassignment according to this TTC, remove the associated agents from the audience, and update the supply.
6. If some tool type has vanished from the supply due to the most recent reassignment, go to step 2. Otherwise, go to step 4.

TTC^+ This algorithm is the same as the above TTC algorithm, except that a TTC of maximal size[22] is chosen in step 5. Note that TTC^+ involves much more effort on the part of the auctioneer than plain TTC, which only needs to compute one top-trading cycle.

$\mathrm{TTC}^\bullet$ This algorithm is the same as TTC, except that, in step 5, the auctioneer searches for a maximal*set* of TTCs that can be satisfied simultaneously. At first sight, computing a maximal set of TTCs may seem to involve a higher computational cost than computing a single maximal TTC. However, because $\mathrm{TTC}^\bullet$ makes use of the elementary composition algorithm described in Sect. 3.2.9, it is, in fact, less expensive than TTC^+.

$\mathrm{TTC}^\vee$ This is a computationally costly backtracking algorithm based on the balancing algorithm described in Sect. 3.2.7. It proceeds similarly to $\mathrm{TTC}^\bullet$, except that it computes all TTC sets that cannot be expanded further, and explores all of these alternatives. Also, it is not informationally decentralized: The mediator needs to know the preferences of all agents (up to their initial endowments).

IG^∇, IG_0^∇ These mechanisms differ from IG and IG_0 in that they do not employ any eligibility heuristic. All competing proposals are equally eligible. Hence, if we stick to the letter of Def. 5.2 of IG mechanism, they are not IG mechanisms. However, they can be selected in the IGRT, and are obviously closely related to IG.

DN We described delegated negotiation in Sect. 5.5 as a single-shot reallocation mechanism, where each agent has one opportunity to submit an exchange proposal which it considers optimal in view of the tool type supply and demand conveyed by the auctioneer. Because we did not provide a formal definition of agent rationality in DN, this description does not define a DN reassignment mechanism unambiguously. However, if we assume that the agents in DN exhibit fictitious rationality, as they do in IG, the above description implies that a DN reassignment mechanism is equivalent to IG with 0% asynchrony, no weariness, and a deadline of 2 rounds per phase.

Before we discuss the differences in average effectiveness obtained by (some of) these mechanisms, we summarize the differences in the respective designs. This is done in Table 7.12, in terms of informational and operational decentralization.

As apparent from Table 7.12, $\mathrm{TTC}^\vee$ is not a serious contender among these reassignment mechanisms, because it is not informationally decentralized, and has a very low degree of operational decentralization. Moreover, it is not tractable. Nevertheless, it is included in the effectiveness comparisons, because it provides an upper bound

[22] The size of a TTC is the number of agents involved in it.

mechanism	inform. decentr.?	agent workload	auctioneer workload
IG	yes	high	medium
IG_0	yes	low	medium
IG^{∇}	yes	high	low
IG_0^{∇}	yes	low	low
DN	yes	high	medium
TTC	yes	low	medium
TTC^{+}	yes	low	high
$TTC^{\bullet}$	yes	low	high
$TTC^{\forall}$	no	low	very high
WALRAS	yes	low	medium

Table 7.12: Mechanism characteristics.

for the effectiveness of TTC mechanisms. The TTC mechanisms TTC^{+} and $TTC^{\bullet}$ have in common that they reassign tools incrementally, satisfying the exchange proposals of agents in one or more top-trading cycles in each step. They differ in the heuristic that is employed to determine in which order top-trading cycles are processed. TTC^{+} employs the Largest-Cluster-First (LCF) heuristic, while $TTC^{\bullet}$ uses the Largest-Bag-First (LBF) heuristic.[23] We cannot imagine a better heuristic than LBF at present, but this, of course, does not mean that a better one does not exist. Hence, it is interesting to know how much improvement such a heuristic could bring about. $TTC^{\forall}$ answers this question by providing an upper bound for the effectiveness that can be obtained by a TTC mechanism.

In the following, $\underline{T}TC^{\clubsuit}$ denotes the TTC oracle, the *hypothetical*, informationally decentralized mechanism that renders a TTC solution of maximal effectiveness whenever it is invoked. Except as a means to compute $TTC^{\clubsuit}$'s effectiveness, we also use $TTC^{\forall}$ to evaluate the performance of TTC. This is possible, because the IGRT keeps track of the averages of the minimal, maximal, and average TTC solution effectiveness over a sample of problems, if $TTC^{\forall}$ is activated. Since a sample average of the average TTC solution effectiveness reported by $TTC^{\forall}$ is an unbiased estimate of the average effectiveness obtained by TTC on the associated sample space, we can use $TTC^{\forall}$ to evaluate TTC. In fact, this is preferable to the more direct approach of implementing TTC itself as an algorithm that greedily satisfies the proposals associated with any top-trading cycle it encounters, because the effectiveness obtained by such an algorithm will depend on implementational details (in particular, *how* it searches for a top-trading cycle).

Because Walrasian equilibria are rare in sample spaces of typed reassignment problems, and the IGRT has no provisions to generate only Walrasian markets, we would

[23] cf. Sect. 3.2.3.

have to generate very many problem instances to come up with statistically significant differences in average effectiveness between IG and the Walrasian auction. Furthermore, even if we would do this, the comparison between IG and the Walrasian auction is bound to be of little value. As we pointed out in Sect. 4.6.4, there are indications that a large proportion of the typed reassignment problems that do possess a Walrasian equilibrium are trivial problem instances (viz. equilibrial markets).

In view of this, the comparison between IG and WALRAS is performed on sample spaces of *untyped* reassignment problems with weakly row-monotonic utility matrices. In such spaces, each problem is a Walrasian market, and if the utility matrix is row-monotonic, all of the TTC algorithms coincide with WALRAS, that is, they render the (unique) Walrasian assignment. If the utility matrix is not row-monotonic, there are generally multiple TTC solutions, but each of these corresponds with the Walrasian allocation of some market in the SPAM enumeration of the problem. This implies that the average performance of WALRAS on a sample from an untyped problem space is indicative for the average performance that can be expected of *any* of the individually rational mechanisms in Table 7.11.

It is computationally costly to obtain statistically significant findings on large problems with the $\text{TTC}^{\forall}$ algorithm. To solve one problem with 10 agents and 4 tool types, $\text{TTC}^{\forall}$ requires approximately 100 ms (within the IGRT on a 486 PC). Solving a problem with 12 agents and 5 tool types takes almost one second. Apparently, adding 2 agents and one tool type incurs a tenfold increase in computation time. If we extrapolate this finding exponentially[24] to problems with 20 agents and 9 tool types, processing a standard-size sample (i.e., one of 20,000 problem instances) would take more than 6 years. Hence, we only apply $\text{TTC}^{\forall}$ to sample spaces with 12 agents and 5 tool types.

I have not implemented TTC^{+}, for two reasons. First, it is very unlikely that this algorithm will ever outperform TTC^{*} in terms of average solution quality. The LCF heuristic in TTC^{+} is based on the hypothesis that it is profitable to satisfy as many agents as possible with their (current) first preference. If this hypothesis is correct, the LCF heuristic is never better than the LBF heuristic of TTC^{*}, while we have seen (in Fig. 3.6 on page 59) that there are cases where LBF leads to much more satisfied agents than LCF. Second, our implementation of TTC^{*} is fast, because it makes use of the same (polynomial) algorithm we use to compute the optimal community utility of a problem instance (Jonker & Volgenant, 1987). It is not obvious that a similarly fast algorithm for TTC^{+} exists.

[24]Exponential extrapolation amounts to the assumption that the addition of 2 agents always incurs a tenfold increase.

7.7.1 IG versus the Walrasian auction

We compare the effectiveness obtained by IG with that obtained by a Walrasian auction on sample spaces with various problem sizes. It has become clear in Chapter 4 that Walrasian equilibria seldom exist for *typed* reassignment problems, especially if one excludes (trivial) zero-tension problems, as the IGRT does.[25] Hence, the solution quality comparison is performed on problem spaces with untyped reassignment problems, for which a Walrasian equilibrium always exists. Except for the tool bag, we use the default setting of problem generation parameters. In other words, for all of the investigated sample spaces, `utility range` = 10, `low alternative` = 3, and `high alternative` = 9, while `zero utility` equals −25 in the heterogemic case, and 50 in the homogemic one. Thus, the sample spaces associated with the data points in each of the graphs differ only in the multiplicity types of the associated tool bags ($\{1 : n\}$, where $n = 10, 12, 14, 16, 18, 20$). In the following, we denote these sample spaces by S_n if they are homogemic, and by $\bar{S}_n$ if they are heterogemic. The base-level effectiveness values of each of the sample spaces are listed in Table 7.13.

nr of agents (n)	10	12	14	16	18	20
avg. tension in S_n	3.49	4.22	4.97	5.69	6.42	7.17
avg. C^* in S_n	90.3	91.3	92.0	92.6	93.1	93.5
$\widehat{E^\perp}$ of S_n	72.1	72.0	70.0	69.5	69.1	68.7
avg. tension in $\bar{S}_n$	3.48	4.22	4.97	5.70	6.43	7.16
avg. C^* in $\bar{S}_n$	90.6	91.5	92.0	92.6	92.9	93.4
$\widehat{E^\perp}$ of $\bar{S}_n$	72.0	71.8	70.1	69.5	69.2	68.8

Table 7.13: Base-level effectiveness ($\widehat{E^\perp}$) of the sample spaces.

The base-level effectiveness appears to vary roughly[26] between 72% for the small problems and roughly 70% for the large ones. Note, however, that the tension in a problem space with large problems (e.g., S_{20}) is considerably higher than the tension in the default space. Hence, the associated base-level effectiveness is an overly pessimistic estimate of the effectiveness that an unsophisticated mechanism can obtain. Consequently, it is not permissable to conclude from the decreasing $\widehat{E^\perp}$ with increasing problem size that larger problems are more difficult. In fact, the rising effectiveness curves in Fig. 7.27 strongly suggest that larger problems are easier.

[25] See the remarks about Walrasian subspaces of SPAM$^*(12, 4)$ near the end of Sect. 4.6.4.

[26] The differences between the average tension reported in the homogemic and heterogemic samples indicate that the third digit of the computed $\widehat{E^\perp}$ values is not entirely reliable (since the average tension in the sample *spaces* is the same in both cases).

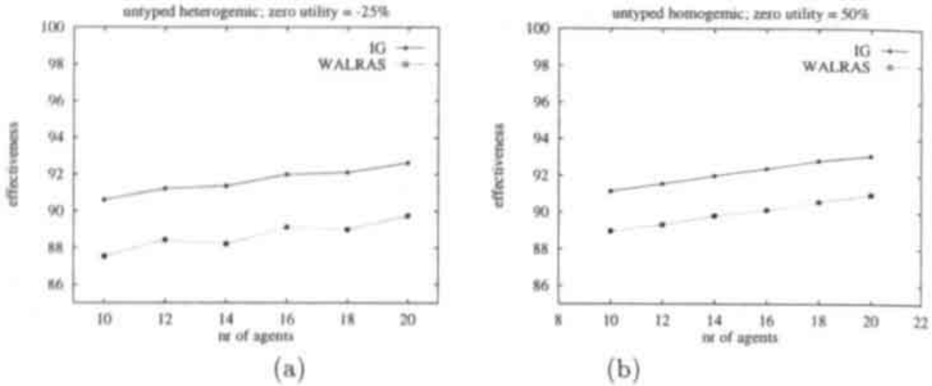


Figure 7.27: Performance comparison of IG and Walrasian exchange
(a) on heterogemic problems.
(b) on homogemic problems.

As apparent from Fig. 7.27, IG outperforms the Walrasian auction, both on heterogemic and on homogemic problems. The performance differences appear to be somewhat larger in the heterogemic case. This is probably due to the fact that the cost of ensuring individual rationality tends to be higher for heterogemic problems than for homogemic ones. In both cases, the differences in average effectiveness are statistically significant. They are also considerable in comparison with the variations observed between different IG mechanisms in the previous section.

To get a better idea of the differences between the quality of the solutions rendered by the two mechanisms, we do not only compare the *average* solution quality, but also inspect the respective distributions. Fig. 7.28 portrays the distribution of the effectiveness obtained with IG and that of the Walrasian assignments of the 20,000 sample problems from which the data points at `agents` = 12 in the leftmost (heterogemic) graph of Fig. 7.27 were computed.

As Fig. 7.28 shows, there is more variation in the effectiveness of Walrasian equilibria than in that of IG solutions. Furthermore, the distribution curve of the Walrasian auction lies entirely above that of IG for effectiveness values below 90%, and entirely below it for higher effectiveness values. As such, there is a consistent pattern of more frequent high performance and less frequent low performance of IG, in comparison with the Walrasian auction.

This does not imply, however, that IG performs consistently better than a Walrasian auction on each of the sample problems. If we plot the distribution of the difference between the effectiveness values on the same problem instances, the effectiveness obtained by IG appears to range from 25 percent below to 35 percent above the effectiveness of the Walrasian equilibria. This differential distribution is shown in

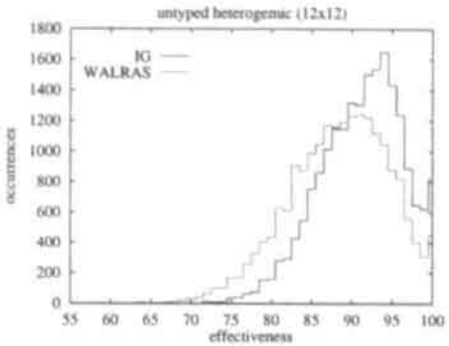


Figure 7.28: IG and Walrasian effectiveness distributions.

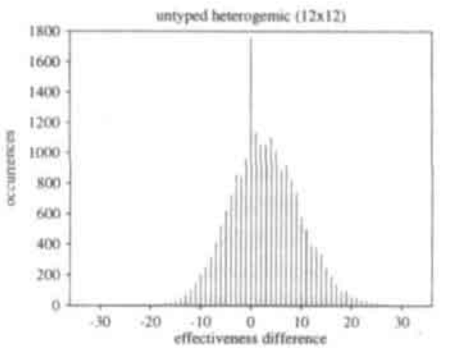


Figure 7.29: Differential distribution (IG minus Walrasian effectiveness).

Fig. 7.29.

Conceptually, IG differs from the Walrasian auction in two respects.

1. The reassignments rendered by IG are not always individually rational, while those rendered by a Walrasian auction are.

2. The distinction between "slightly better" and "much better" influences the decisions of agents in an IG auction, while it does not play any role in a Walrasian (exchange!) auction.

This prompts the question which of these differences is the most important one. In other words, what is the *primary* cause of IG's superior performance? To test this, we have repeated the simulations underlying Fig. 7.27, 7.28, and 7.29 on heterogemic sample spaces with `utility range` = 99, and `low alternative` = `high alternative` = 98. With this parameter setting, the generated utility matrices are very nearly binary, all agent-level utilities equalling either 0, or 1, or (98/99 =)

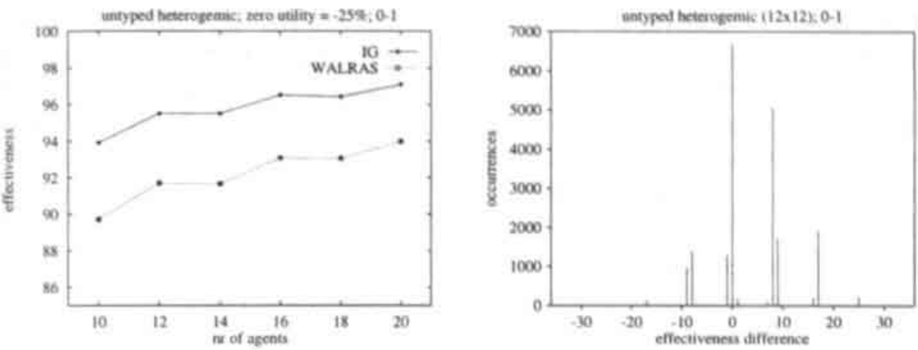
0.9899. In other words, agents are either (almost) fully satisfied with an endowment, or utterly dissatisfied.

At first sight, one might be prone to conclude that cardinal distinctions — which do not play *any* role in binary problems — play a relatively *minor* role in near-binary ones. The converse is true, however. In the context of the near-binary utility matrices, a final endowment that is "somewhat better" for the agent than its initial endowment involves an improvement in agent-level utility of about 1% (from 98.99% to 100%), while a "much better" final endowment can only be attained by agents with zero initial utility, and constitutes an improvement in normalized agent utility of virtually 100% (from 0% to either 98.99% or 100%). As such, the difference between "somewhat better" and "much better" in this sample space is about as large as it can get.

The other aspect in which IG differs from WALRAS is the absence of an individual-rationality constraint. As for the relative impact of this difference on allocation effectiveness in the case of near-binary matrices, it is difficult to draw conclusions on theoretical grounds. However, the outcomes of our experiments with the near-binary problems rendered two indications that the absence or presence of an individual-rationality constraint has relatively little impact on the final effectiveness that can be obtained. First, the *lowest* observed effectiveness values of optimal individually rational reassignments were 99.99% for $n = 10$, 99.91% for $n = 12$, and 100% for the other population sizes. Second, IG solutions were virtually always individually rational: the percentage of zero-loss auctions was 99.99% in the samples with population sizes of 14 and 20, and 100% in the other samples.

Summarizing, with near-binary problems, IG's ability to reckon with cardinal utility differences is bound to have a greater impact on allocation effectiveness (than with problems drawn from the default sample space), while the absence of an individual-rationality constraint is of relatively minor importance. Hence, if IG outperforms WALRAS to a *greater* extent on the near-binary problems than it did on problems generated drawn from the default sample space, we can conclude that the most important difference between IG and WALRAS is IG's ability to take cardinal utility differences into account. If, on the other hand, the effectiveness difference between IG and WALRAS is much smaller with the near-binary problems, this would indicate that the absence of an individual-rationality constraint is the primary cause of IG's superior performance.

The outcomes of our experiments with the near-binary problems are pictured in Fig. 7.30. The first plot shows the average effectiveness obtained by the two mechanisms on samples of near-binary problems, with population sizes ranging from 10 to 20 agents. The second plot shows the distribution of the extra effectiveness obtained by IG on problems with 12 agents (i.e., problems drawn from $\tilde{S}(\{1:12\}, 99, 3, 98, 98)$).



(a) (b)

Figure 7.30: Effectiveness comparison on near-binary problems.
(a) Effectiveness of IG and WALRAS for various n.
(b) Distribution of the extra effectiveness ($E_{\text{IG}} - E_{\text{WALRAS}}$) obtained by IG over WALRAS with $n = 12$.

It appears that IG outperforms the Walrasian auction on these sample spaces also. Moreover, comparison of the first plot with Fig. 7.27 reveals that the differences in average effectiveness are larger than with the untyped default problem spaces, and comparison of the second plot with Fig. 7.29 shows that the superiority of IG is also more consistent on the near-binary problems. Hence, we conclude that the most important advantage of IG over WALRAS is its emergent capability to take cardinal differences between the utility gains of different agents into account.

7.7.2 IG versus TTC mechanisms

The Top-Trading-Cycle mechanisms are not plagued by the absence of equilibria. Hence, we measure their effectiveness on sample spaces of typed reassignment problems. The sample spaces used are listed in Table 7.14.

space	deviation from the default parameter setting
$\tilde{S}$	none
S	`zero utility` $= 50\%$
S_{zz}	`zero utility` $= 0\%$
$\tilde{S}_{ecc}$	$\alpha = \{8, 1, 1, 1\}$
$S_{20\times 9}$	$\alpha = \{3, 3, 2, 2, 2, 2, 2, 2, 2\}$

Table 7.14: The sample spaces used for mechanism evaluation.

The comparison involves IG, the related mechanisms IG_0, IG^∇, IG_0^∇, and DN, and the Top-Trading-Cycle mechanisms TTC, $TTC^\bullet$, and $TTC^\clubsuit$. We employ $TTC^\triangledown$ to emulate $TTC^\clubsuit$ as well as TTC, using the reported maximum and average TTC effectiveness respectively. In view of the computational complexity of $TTC^\triangledown$ (cf. page 326), we did not attempt to apply it to $S_{20\times 9}$. The attempt to apply $TTC^\triangledown$ to $\tilde{S}_{ecc}$ failed for the same reason. Consequently, there are no experimental data on the solution quality obtainable with TTC and $TTC^\clubsuit$ on $S_{20\times 9}$ or $\tilde{S}_{ecc}$.

The numerical outcomes of the other experiments are shown in Table 7.15. A schematic display of these outcomes is presented in Table 7.15.

sample space	avg. effectiveness obtained by									% W
	IG	IG_0	IG^∇	IG_0^∇	DN	TTC	$TTC^\bullet$	$TTC^\clubsuit$	W	
$\tilde{S}$	92.4	91.2	92.6	91.5	91.8	89.6	89.9	96.2	89.9?	1.6
S	93.0	92.2	92.9	92.2	92.1	91.2	91.5	97.1	91.3?	1.4
S_{zz}	96.3	95.9	96.1	95.9	96.0	95.0	95.4	98.3	95.4?	1.6
$\tilde{S}_{ecc}$	89.1	89.4	87.8	89.1	88.4	-	86.8	-	82.9?	3.1
$S_{20\times 9}$	92.8	91.6	93.6	91.6	91.1	-	89.9	-	92.2??	0.07

Table 7.15: Numerical performance comparison of IG- and TTC-like mechanisms (in terms of the average effectiveness obtained on the sample).

The precision of the data in Table 7.15 generally reflects the level of statistical significance.[27] This is not the case, however, for the last two columns. The data in these columns represent the measured average effectiveness of Walrasian assignments, and the percentage of the sample problems for which such an assignment turned out to exist. From the percentages in the last column, it follows that the averages in the 'W' column are based on a number of problem instances, ranging between 14 for $S_{20\times 9}$ and 620 for $\tilde{S}_{ecc}$. Hence, none of the entries in the 'W' column are reliable, and the bottom entry constitutes little more than a wild guess.

The reliable data in Table 7.15 are reflected in Table 7.16, in the form of an *ordinal* comparison of the performance of the mechanisms. In each column of the table, the mechanisms are listed in (top-down) decreasing order of average effectiveness for the associated sample space. The presence of a horizontal line between two mechanisms in the same column indicates that the performance difference between these two mechanisms is statistically significant.[28] Note that a statistically significant performance difference can still be marginal. This is the case, for example, for

[27] The size of the confidence intervals varies between 0.006 and 0.07.

[28] Here, "statistically significant" means that the (95%) confidence intervals around the respective estimates of the average effectiveness do not overlap.

$\bar{S}$	S	S_{zz}	$\bar{S}_{ecc}$	$S_{20\times 9}$
TTC$^{\clubsuit}$	TTC$^{\clubsuit}$	TTC$^{\clubsuit}$	?	?
IG$^{\nabla}$	IG	IG	IG$_0$	IG$^{\nabla}$
IG	IG$^{\nabla}$	IG$^{\nabla}$	IG$_0^{\nabla}$	IG
DN	IG$_0$	DN	IG	IG$_0^{\nabla}$
IG$_0^{\nabla}$	IG$_0^{\nabla}$	IG$_0^{\nabla}$	DN	IG$_0$
IG$_0$	DN	IG$_0$	IG$^{\nabla}$	DN
TTC$^{\bullet}$	TTC$^{\bullet}$	TTC$^{\bullet}$	TTC$^{\bullet}$	TTC$^{\bullet}$
TTC	TTC	TTC	?	?

Table 7.16: Ordinal performance comparison of IG and TTC mechanisms (in terms of top-down decreasing effectiveness).

the difference in average effectiveness obtained by IG$^{\nabla}$ and IG on $\bar{S}$, which — as inspection of Table 7.15 reveals — equals approximately 0.2%.

The question marks represent conjectures on the positions of TTC and TTC$^{\clubsuit}$ in the ranking for the sample spaces $\bar{S}_{ecc}$ and $S_{20\times 9}$, on which the application of TTC$^{\vee}$ was not feasible.

From the table, we conclude that the average effectiveness of the Top-Trading-Cycle mechanisms TTC and TTC$^{\bullet}$ is below that of *each* of the IG-related mechanisms on *all* of the investigated sample spaces.[29] Inspection of the table reveals that the difference between the performance of TTC$^{\bullet}$ and the worst performance of any IG-related mechanism is always statistically significant, though sometimes marginal: Numerically (cf. Table 7.15), the performance differences range from 0.2% (with $S_{20\times 9}$) to 1.0% (with $\bar{S}_{ecc}$).

In general, the IG mechanisms with reckless agents (i.e., IG$_0$ and IG$_0^{\nabla}$) obtain a lower average effectiveness than those in which the agents perform risk estimation. However, this is not the case on $\bar{S}_{ecc}$. Here, reckless agents appear to have an advantage.

To explain this phenomenon, we recall our analysis of the relatively low IG performance on problems with eccentric tool bags in Sect. 7.5. Here, we mentioned that, in the context of an eccentric tool bag, the most numerous tool type is often the only oversupplied one. This causes the agents endowed with this tool type to *behave* recklessly, even though they *are* not reckless (in the sense that they do perform risk estimation). This can lead to outcomes of low community utility, because these 'auction hogs' tend to *intimidate* the other agents. With this in mind, it is not surprising that it appears to be advantageous to refrain from risk estimation in the

[29] As in the case of the Walrasian auction, this does not imply that the IG-related mechanisms perform better than the TTC mechanisms on every individual problem instance.

context of an eccentric tool bag: Reckless agents are not susceptible to intimidation.

While IG_0 usually provides solutions of lower quality than IG, it performs better than IG if the tool bag is very eccentric. This indicates that the effects of parameters in the context of the default sample space are not necessarily indicative for the effects in other sample spaces. In particular, the observed superiority of IG_0 prompts the question what the *best* setting for the caution parameter is, if the tool bag is eccentric. To answer this question, we performed the experiments with the caution parameter again, on the sample space $\tilde{S}_{ecc}$.[30] The results are shown in Fig. 7.31.

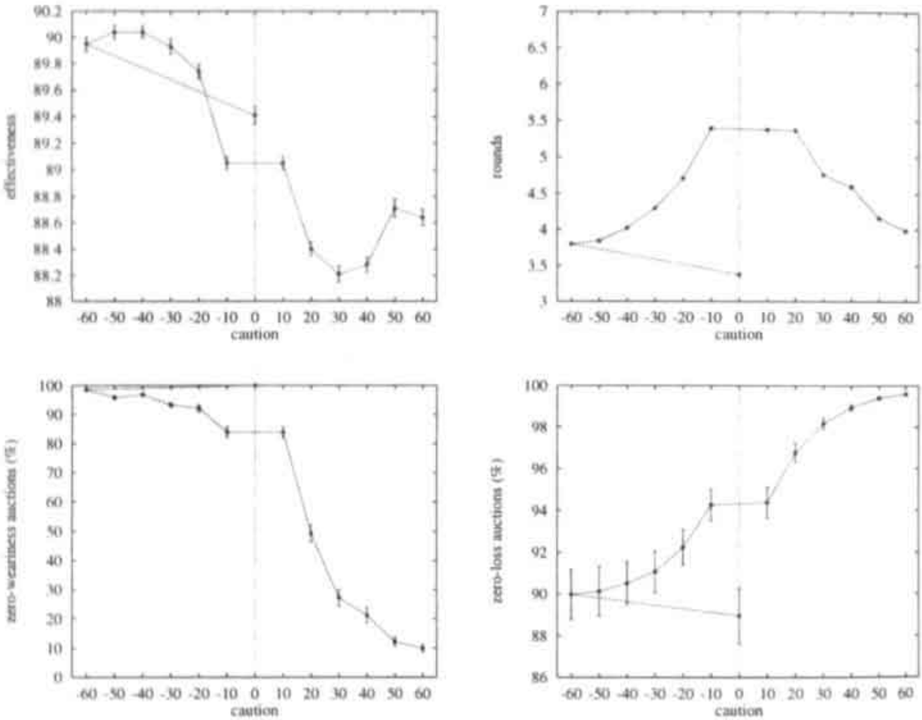


Figure 7.31: Influence of the `caution` parameter on $\tilde{S}_{ecc}$.

The influences of `caution` in Fig 7.31, in the context of the sample space $\tilde{S}_{ecc}$ appear to be quite different from those exhibited in the context of the default sample space, shown in Fig. 7.22 on page 314. This is especially so for the influence of caution on effectiveness. While `caution` = 10 renders maximal average effectiveness for problems in the default sample space, the best setting for problems in $\tilde{S}_{ecc}$ lies somewhere near `caution` = −50. As the second plot in Fig. 7.31 shows, this is also a good setting in terms of the average auction duration. Furthermore, the difference

[30]cf. Table 7.14 on page 331.

between the average effectiveness at this value and that at the default caution value of 10 is considerable (approximately 2%).

The second plot also differs considerably from the rounds plot in Fig 7.22. In the context of $\bar{S}_{ecc}$, the default setting of `caution` = 10 appears to be the very *worst* setting (together with `caution` = 20), as far as auction duration is concerned.

The most important lesson from this experiment is that there is no single best attitude toward risk. What the most rewarding attitude is, depends on problem characteristics, even within a relatively confined problem domain such as reassignment.

The average effectiveness of the *best* TTC solution, rendered by the hypothetical TTC$^\clubsuit$ mechanism, is considerably higher than the average effectiveness obtained by any of the IG-related or (real) TTC mechanisms. While this shows that there is *room* for improvement on TTC$^\bullet$'s LBF heuristic, it does not imply that improvement — in the sense of finding a superior *heuristic* — is feasible.

In fact, we claim that such a heuristic does not exist for the general case. In the absence of any information other than the bag of submitted proposals, the LBF heuristic is obviously optimal: It maximizes the conditional expectation of the community utility under the condition that the *cardinal* utilities associated with proposals are distributed uniformly and independently. A superior heuristic only exists if there is information to the contrary (on these probability distributions). In the present context, the only such information is the preference information embodied in proposals that were communicated earlier. In general, it is very difficult to translate this information into hypotheses on the underlying cardinal utilities. Hence, any algorithm that makes use of such information is are bound to be complex if it is to constitute a significant improvement over TTC$^\bullet$ (let alone IG). The term 'heuristic' would be inappropriate for such a complex reasoning scheme.

7.8 Chapter Summary and Conclusions

7.8.1 Experiment-categories

The conducted experiments fall into five categories:

1. performance evaluation of the default IG mechanism on various sample spaces (selected by using different settings of problem-generation parameters);

2. characterization of the problems that are difficult for IG;

3. performance evaluation of various IG mechanisms (selected by using different settings of agent- and mechanism-parameters) on the default problem space;

4. performance comparison of IG mechanisms, Walrasian exchange, and Top-Trading-Cycle mechanisms;

5. investigation of the impact and sources of nondeterminism in IG.

Below, we summarize the most important findings in each of these categories, together with some concepts that were conceived in the process of analyzing the experimental results. Unless stated otherwise, the conclusions about IG performance pertain to the primary performance attributes allocation effectiveness and auction duration, with the emphasis on the former.

7.8.2 Sample-space selection

The experiments in category 1 revealed that our default sample space (which was used in the majority of experiments) is adequate in the sense that the generated problems are sufficiently 'average' without being too easy.

Variation of the setting of problem-generation parameters — usually involving variation of only one or two parameters with all others at their default setting — led to the following observations. Variation of the number of tool types revealed that IG performs better on (highly) typed problems than on (nearly) untyped ones. Also, IG performance tends to be higher on problems with homogeneous tool bags than on problems with highly eccentric ones. A high `zero utility` (i.e., percentage of useless tool types) is detrimental for IG performance. Finally, the utility range of alternatives for the agents' first preferences — defined by the parameters `low alternative` and `high alternative` — appears to influence the impact of the `zero utility` on IG performance. The variation of performance with `zero utility` is most pronounced if the utilities of tool alternatives lie in a narrow band near 100%.

7.8.3 The impact of problem scale

Our analysis reveals that there is no unambiguous answer to the question how IG performance scales up, because it is not immediately clear what a scale increase amounts to for *typed* reassignment problems. If one is solely interested in the performance aspect of auction duration, this is not much of an issue, since the auction duration increases with increasing scale, whichever (sensible) definition of scale we employ. The effects of problem scale on solution quality, however, differ substantially for different interpretations of 'scale increase'.

It depends on the nature of the (real-life) problem context whether the average effectiveness obtained by IG increases, decreases, or is approximately constant with increasing scale. In a domain where a scale increase is synonymous with an increase of the population, the performance of IG increases with scale. In a domain where an increase of the population is usually accompanied by an increase of the number of tool types, IG performance is either approximately constant, or it decreases with scale. It depends on the number of useful tool types per agent which of these two cases applies. In domains where the number of useful tool types tends to be proportional to the total number of tool types, the average IG effectiveness is not significantly affected by the problem scale. If the number of useful tool types per agent is approximately constant, that is, independent of the total number of tool types, the average IG effectiveness decreases with increasing scale.

Neither of these scaling effects are special features of IG; in our analysis, we argued why it is to be expected that they also pertain to other strongly decentralized mechanisms: They can be explained as a consequence of the nature of reassignment problems in combination with the fact that we use an measure of solution quality that is an average of individual agent satisfaction.

7.8.4 Characterization of difficult problems

Most of the above problem generation parameters appear to have a considerable impact on IG effectiveness, even if we confine ourselves to single-parameter variation. However, we observed that there is a considerable gap between the average effectiveness of IG on the most difficult sample spaces and the lowest effectiveness obtained on individual problem instances.

This suggests that the chosen set of problem-generation parameters is not adequate to characterize the hard problems. We identified the *tension* (i.e., the discrepancy between initial supply and demand) of a reassignment problem as a quantity that is a good measure of problem difficulty, and showed experimentally that problems of *near*-maximal tension are among the most difficult problems. Yet, the average tension in a sample space is not the sole determinant of IG's performance on that space. Another important problem property in this respect is the degree to which the oversupply is spread over different tool types. Our analysis of experimental outcomes revealed that IG's eligibility heuristic falters if there is only one oversupplied tool type, because the agents endowed with this tool type cannot possibly lose any utility, and therefore behave recklessly. Such a focus of oversupply on one (or very few) tool type(s) correlates with high tension, but it cannot occur in (viable) reassignment problems of maximal tension. This is a plausible explanation why problems with near-maximal tension tend to be more difficult for IG than problems with maximal tension.

The tension itself is not a suitable problem-generation parameter, because *efficient* generation of problem instances with a prespecified tension is feasible only for sample spaces associated with one specific utility matrix.

However, we identified the *correlation* between agent preferences as a candidate problem-generation parameter that correlates with the tension, and does allow for efficient sample-space selection. The incorporation of a preference-correlation parameter would also offer the opportunity to improve the extent to which the generated problems can be qualified as realistic. In retrospect, we concluded that our assumption of zero average correlation between the preferences of different agents is a shortcoming in this respect.

We also used the tension to define a (problem-dependent) reference level for the average effectiveness that a decentralized mechanism should at least attain. Knowledge of this ***base-level effectiveness*** is required to judge whether a measured average effectiveness of less than 100% qualifies as admirable or pityful performance. The base-level effectiveness of a single problem instance equals the amount of effectiveness stemming from first preferences, if one assigns as many first preferences as possible. Hence, it is a lowerbound on the effectiveness that can be obtained by a trivial variant of IG, where all proposals are equally eligible, and agents simply stick to their first preferences, without paying any attention to the risk of failure.

7.8.5 Impact of agent- and mechanism-parameters

The outcomes of our experiments indicate that negotiation weariness should definitely be incorporated in IG, and that the market clearance condition for phase termination is best omitted: It is hardly ever invoked, and has hardly any impact on the average performance.

Volatility is probably best left out also. Although our expectation that a small positive setting would lead to better performance than the default setting of zero volatility appeared to be principally correct, the improvement is so marginal that it does not outweigh the disadvantage that a positive volatility constitutes irrational behavior.

As for the best setting of the other agent and mechanism parameters, there are often tradeoffs between solution quality and auction duration, or between different aspects of solution quality (such as effectiveness and zero-loss). If we regard the effectiveness as (by far) the most important performance attribute, the following conclusions can be drawn.

1. The default setting of the agent and mechanism parameters appeared to be optimal, or nearly so, on the default sample space, with respect to the primary performance attributes (the average effectiveness and the average auction duration in rounds).

2. The resolution parameter, which determines the courseness of the eligibility ordering employed in the pseudo-composition protocol, appears to have only a marginal effect on IG effectiveness. Within the original value range

(low,mixed,high), mixed resolution comes out first but the differences with respect to average effectiveness are minute, and not statistically significant. It turns out, however, that the value `eligibility` = `zero`, that is, the IG mechanism in which eligibility plays no role at all, performs (marginally, but significantly) better than IG with mixed resolution in terms of effectiveness, and only slightly worse in terms of auction duration. In other words, for reassignment problems, the eligibility heuristic is of little value. The original idea behind this heuristic was to incent agents to release *scarce* tools so as to improve their chances of obtaining other scarce tools. Because an agent in a reassignment problem only possesses a single tool, while an agent in a full-fledged reallocation problem can possess tools of many different types, the merits of the eligibility heuristic may well be greater in the context of full-fledged reallocation.

3. Asynchrony appears to have a considerable effect on auction duration, but its influence on effectiveness is marginal. Also, if one opts for asynchronous parallelism (i.e., for `asynchrony` > 0), the exact setting is not very important. Grossly, moderate asynchrony (say, between 40 and 75%) is preferable over more extreme settings. With weariness incorporated, employing synchronous parallelism becomes a viable option: It leads to a higher average auction duration, but also to a (marginally) higher average effectiveness.

4. If weariness is incorporated in IG, the deadline is an agent parameter, quantifying the negotiation stamina exhibited by the agents. In this case, the default setting of the deadline parameter (to 500 rounds) appears to be unnecessarily high. A `deadline` value of about 30 rounds is high enough for the default sample space. Smaller values (e.g., 5 or 10) render a much lower average effectiveness, but the curve that pictures the influence of the deadline on effectiveness levels off at `deadline` = 20. There are no statistically significant differences between the effectiveness values obtained with higher deadlines. This observation pertains to the default sample space of problems with 12 agents and 5 tool types, but scale does not seem very relevant here: the curve for the sample space of problems with 20 agents and 9 tool types levels off at `deadline` = 20 as well.

5. The remaining agent parameters, caution and maxloss, are alternative means to reduce the frequency and severity of utility loss. Utility loss can only be prevented completely by setting maxloss to zero, but this tends to be costly in terms of decreased effectiveness (incurring decreases like from 92.5% to 87.9%). If utility loss is considered acceptable, provided that it does not occur very often, the caution parameter is a preferable handle to reduce the frequency of utility loss. A reduction of the percentage of auctions featuring utility loss from 22 to 9% costs about 2% in effectiveness decrease with `caution` as a

handle, and about 2.5% with `maxloss`. These effectiveness decreases are much larger than the difference between the effectiveness of an optimal allocation and an optimal, individually rational allocation (viz. 0.39% on average for the default sample space).

Furthermore, our experiments point out that Walrasian exchange and TTC mechanisms are also relatively costly means to produce individually rational solutions: The average effectiveness of (individually rational) Walrasian allocations is approximately 2% below the average effectiveness obtained by IG on the same samples.

Together, these observations suggest that the cost of preventing or reducing utility loss is considerably higher in strongly informationally decentralized mechanisms (such as IG and Walrasian exchange) than in mechanisms without such decentralization (such as the monetary Walrasian auction).

6. The percentage of zero-loss auctions can vary considerably for different values of a parameter, but this is often compensated by the percentage of agents that loses utility. As an example, the percentage of zero-loss auctions tends to decrease with increasing problem scale, but this is compensated by a decrease of the average percentage of agents that lose utility. As a consequence, the probability that an agent which takes part in an IG auction will lose utility is not significantly dependent on problem scale. This probability equals 2% with the default parameter setting for all of the investigated problem sizes.

7. It turns out that the optimal setting for the caution parameter depends heavily on the kind of reassignment problems IG is confronted with. Risk-neutral neutral agents perform best (in terms of effectiveness) on the default problem space, but risk-insensitive agents do better in the context of a community tool bag that is very eccentric. In this case, the caution value of 50 is optimal, both in terms of average effectiveness and average auction duration. In both cases, excessive pessimism is more damaging (to effectiveness as well as auction duration) than excessive optimism.

7.8.6 Performance comparison of IG and other mechanisms

On all investigated sample spaces (which involve only *untyped* reassignment problems in the case of the Walrasian auction), all of the major variants of IG appear to outperform the Walrasian auction as well as decentralized TTC mechanisms in terms of average effectiveness.

7.8.7 Nondeterminism

Our experiments have revealed that the variation in performance within a problem sample tend to be much larger than the differences between the average values of performance attributes over the samples. In other words, if mechanism X outperforms mechanism Y, the superiority of X over Y is all but consistent. This is partly due to the fact that IG mechanisms tend to involve many nondeterministic decisions. As a consequence, an observed on-average superiority of IG mechanism X over mechanism Y is not even guaranteed to be consistent in repeated application of X and Y to the *same* problem instance.

This prompted us to investigate to what extent the agent- and mechanism-parameters can be used to reduce the variation in performance due to nondeterminism. There are three candidate parameters in this respect. The resolution parameter affects the frequency of nondeterministic decisions by the auctioneer, the maxloss parameter affects the frequency of such decisions by the agents, and the asynchrony parameter affects the amount of nondeterminism due to variations in agent response speed and communication delays. Maxloss is a less suitable handle, because it reduces the amount of nondeterminism by narrowing instead of sharpening the agents' eyesight. Thus, reduction of nondeterminism by means of maxloss is relatively costly in terms of effectiveness decrease. Hence, we confined our experiments to the asynchrony and resolution parameters.

It turns out that these two parameters, in combination, exert a considerable influence on performance variation due to nondeterminism. The average number of different solutions rendered for one and the same problem varied between 18.5 (with 50% asynchrony and zero resolution) and 2.8 (with 0% asynchrony and mixed resolution), with associated variations in average effectiveness of 11.1% and 3.6%.

However, the prospects of a further reduction of the effectiveness range (to less than 3.6%) do not seem very bright: The amount of nondeterminism due to asynchrony is already at its minimum with `asynchrony` = 0%, and a higher-than-high resolution is technically possible, but unlikely to have the desired effect, in view of the observation that changing the resolution from mixed to high does not decrease the amount of performance variation at all.

Chapter 8

Conclusions, Future Research, and Reflection

8.1 Chapter Overview

This last chapter comprises three parts. In Sect. 8.2, we recapitulate the most important conclusions, and put these into perspective. In Sect. 8.3, we suggest some topics for future research on Informed Gambling. Finally, in Sect. 8.4, we reflect on our work, and discuss how and why it differs from most other MAS research.

8.2 Conclusions

In Chapter 1, we announced that the thesis would address two questions.

1. How do agent characteristics and coordination rules influence the performance of a multi-agent system?
2. How can we solve TR problems with multi-agent systems?

The second question played a dominant role in the first five chapters, and was ultimately answered with the presentation of the Informed-Gambling framework. We then used IG in chapters 6 and 7 to address the first, more fundamental question.

Conclusions on the second question

The conclusions drawn in chapters 1 to 5 with respect to the second question, the quest for an adequate multi-agent approach to TR, are based on five demands imposed on multi-agent mechanisms. These demands are recapitulated below.

1. The mechanism should be operationally decentralized.

2. The mechanism should be informationally decentralized.
3. The adequacy of the mechanism should not hinge on the assumption of transferable utility.
4. The mechanism should be productive.
5. The mechanism should render high-quality solutions on average, where the solution quality is defined as the average agent satisfaction.

The main conclusions, drawn in view of these demands, are as follows.

Chapter 3:

- We concluded that the TR problem domain calls for a multi-agent approach which features at least some central coordination. In principle, a fully decentralized approach is possible, but it would be very inefficient, since, in the context of TR, the desirability of a local transaction cannot be determined without global knowledge of the problem.
- To ensure operational decentralization, it is vital that the messages which the agents sent to the coordination module are *simple*. Composite reallocation proposals appear to incur an unacceptable workload for the coordination module, even if one employs a straightforward composition/revision scheme.
- The incentive relaxation scheme employed in the Walrasian auction, and related approaches such as MOP, tends to be an adequate approach for problems featuring divisible goods. Moreover, they are capable of ensuring optimality in the context of transferable utility. However, if the goods are indivisible, and utility is not transferable, the Walrasian auction has two shortcomings.
  1. The relaxation process may never end, since equilibria are generally not guaranteed to exist.
  2. If an equilibrium is found, the associated solution is not guaranteed to be optimal; in fact, the solution quality can be arbitrarily low.

Chapter 4:

- The Walrasian density (i.e., the probability that a randomly picked problem features a Walrasian equilibrium) is low for tool reassignment problems, except for problem spaces with very small-scale problems. The density, which is 1 for untyped reassignment, decreases exponentially as the typedness (the ratio of tools to tool types) increases. In the problem space of reassignment problems with a 3-2-2-1 distribution of 8 tools over 4 tool types, it is already as low as 6%.

- We did not *measure* Walrasian densities in problem spaces of full-fledged tool reallocation problems. However, our theoretical analysis points out that one can expect similar observations here: The Walrasian density is bound to decrease with increasing typedness, and will approach zero as the ratio of agents over tool types approaches infinity.

Chapter 5:

- An Informed-Gambling mechanism is a combination of the two approaches studied in Chapter 3. Incentive relaxation is combined with a heuristic variant of composition/revision. In this combination, commitment in the face of uncertainty plays an essential role. Such commitment is capable of preventing the trust formation which can lead to low-quality solutions in mechanisms based on Walrasian exchange.

- IG agents exhibit behavior that is reminiscent of human agents who grow weary of lengthy negotiations: They become progressively less fussy with time. If negotiation weariness is not incorporated, the stationary states of the relaxation process correspond with the game-theoretic notion of correlated equilibrium. In that case, however, the timely termination of the relaxation process cannot be guaranteed, even if such a correlated equilibrium does exist.

- Agent weariness conflicts with the usual golden standard of perfect rationality as the only admissible model for unknown external agents. Yet, we claim that IG agent models are adequate for open systems, provided that the deadline is changed from a deterministic into a stochastic value.

Conclusions on the first question

With respect to our first question, how agent characteristics and coordination rules influence the performance of a multi-agent system, the most important conclusions, drawn from experiments with the IG reassignment testbed, are the following.

Chapter 7:

- On average tool reassignment problems, the least sophisticated coordination rules appear to work best. IG's eligibility heuristic seems to have no positive impact on IG performance in this case. On difficult, tightly constrained problems, the default IG mechanism (i.e., IG with the eligibility heuristic) tends to perform better.

- It is hazardous to extrapolate the above observation to full-fledged reallocation. The eligibility heuristic is likely to be more effective in this case, because agents can adjust their tool *supply* as well as their demand.

- On average problems, risk-neutral agents tend to perform best. However, overly optimistic agents tend to render considerably better outcomes than overly pessimistic ones. On difficult problems, overly optimistic agents even outperform risk-neutral agents. This observation is pleasant for IG's application to real-life problems: as demonstrated in Sect. 5.9.1, the bounded rationality proposed for such problems results in estimates for proposal success probabilities that are usually optimistic, and never pessimistic.

- The frequency and severity of utility concessions due to negotiation weariness tends to be marginal. This observation supports our conjecture that negotiation weariness is an admissible characteristic of agent behavior, even in open systems. Furthermore, IG mechanisms with weariness tend to render much better solutions than mechanisms without weariness in cases of very low deadline values. For high deadline values, the average solution quality is approximately the same. Thus, weariness allows for IG to be used as an anytime algorithm.

- All variants of Informed Gambling outperform all variants of Walrasian exchange on all investigated problem spaces in terms of the *average* solution quality in a problem space.

- IG does not outperform Walrasian exchange on every problem instance. This is partly due to nondeterminism. If IG is repeatedly applied to the *same* problem instance, the quality of the rendered solutions can vary considerably.

- Individual rationality appears to be much more costly in strongly informationally decentralized algorithms than in centralized ones. If IG agents do not take any risk, the solution is individually rational, but in this case, the average solution quality drops considerably, to the level of other mechanisms that guarantee individual rationality, such as Walrasian exchange.

8.3 Future Research

We envision two areas of useful future research: a further investigation of the admissibility of IG agent models in open systems, and the development of IG mechanisms for real-life optimization problems.

Admissibility of fictitious rationality in open systems

In Sect. 5.9.2, we claimed that IG agent models are admissible in open systems, because the prospects of gaining utility by deviating from the behavior specified in the model are sufficiently discouraging. This statement is not backed up by hard numbers, and hence, open to future research.

A useful and feasible endeavor in this direction would be to determine, by means of testbed simulations, how much utility gain can be obtained by an agent with perfect knowledge of the last proposal profile, within a community of agents who only know the last market profile (cf. (Sandholm & Ygge, 1997)).

In view of the discussion in Sect. 8.4 and 5.9.2, investigation of the gain obtainable from *manipulation* seems neither feasible nor necessary: the transformation of the deadline into a stochastic variable will most likely frustrate any attempts in this direction. However, the effects of the transformation itself on IG performance do deserve further investigation.

Application of IG to real-life optimization problems

In Chapter 7, it has become clear that, on tool reassignment problems, IG compares favorably with Walrasian exchange in terms of allocation effectiveness. Of course, it is not certain that this is indicative for IG's potential with respect to constrained optimization, since this typically involves full-fledged reallocation instead of reassignment. However, the strategy-space expansion which we proposed in Sect. 5.9.1 as a feasible approach for such problems still involves fictitious rationality as a basic reasoning schema. Consequently, commitment in the face of uncertainty plays a prominent role in IG reallocation mechanisms as well. Since this type of commitment is instrumental in IG's superiority over Walrasian exchange in the reassignment domain, an investigation whether IG is also superior in domains involving full-fledged reallocation is called for.

For such an investigation to be worthwhile, a problem domain should have the following characteristics.

soft constraints: The coordination module of IG has no provisions for constraint satisfaction. Consequently, the satisfaction of hard constraints is the responsibility of the agents. Since IG agents generally run some risk that their proposals fail, and may lose valuable tools if this happens, IG is less suited for problems with a tight set of hard constraints.

discreteness: For problem domains that can be tackled by IG as well as by MOP, MOP has the distinctive advantage that its producer agents incur an emergent problem decomposition. Hence, problem domains which are not discrete can probably be tackled more satisfactorily by MOP.

no transferable utility: If the assumption of transferable utility is admissible, a GET-based approach such as MOP should be able to ensure optimality. Unless there are other impediments to MOP application (such as discreteness), there is little purpose in developing an IG mechanism for such domains.

separability: To translate an optimization problem into a (re)allocation problem, the optimization criterion should be expressible as a function of agent utilities which do not depend on the allocations to specific other agents. Since an IG agent is only informed of the collective response behavior of other agents, an agent whose satisfaction depends on the final endowment of *specific* other agent(s) is unable to determine the expected utility of a proposal, even if it is guaranteed to succeed.

We remark that separability is less restrictive than the above description suggests. It does not require that an agent's utility is unaffected by the allocation of goods to others, but only that it should not matter to *which* agents these other goods are allocated. An example of a domain where this distinction is important is air traffic control. If we define agents as airplanes, and tools as reservations of approach corridors during time intervals, the assignment of tools to agents must comply with the safety rules on separation between airplanes (F. Mulder, 1996, personal communication). Hence, for an airplane, the allocation of corridors adjacent — in space or time — to its own corridor is relevant, but it it is not relevant to whom these tools are allocated. Consequently, air traffic control problems can be reformulated as allocation problems in the same manner as the 8-queens problem (Ex. 2.2 on page 22).

8.4 Reflection

We conclude with a reflection on our work. This thesis differs from most other research endeavors in MAS by its nature. It is fundamental, but contrary to most fundamental research, its basic assumptions are grounded in considerations of applicability to a specific — albeit broad — problem domain.

As a consequence, some of the conclusions are more tentative than is common in fundamental research. This pertains, for instance, to the existence of Walrasian equilibria in TR\SPAM, in Sect. 4.9. Our applicability considerations also led to some definitions which are *semi*-formal. The definition of near-perfect rationality (Def. 5.13 on page 215) is a good example.

Another consequence of the emphasis on applicability is that the thesis uses only a few established MAS concepts. Most existing concepts proved inadequate for our purposes. Notions of agent rationality proposed in MAS literature invariably incurred excessive computational complexity in the context of TR, or were based on inappropriate assumptions.[1] Game-theoretic concepts suffered from similar deficiencies, or were at odds with our design requirements of operational and informational decentralization.

However, the difficulties experienced in the search for adequate basic concepts are not due to the nature of our research *per se*. A more fundamental cause is the discrepancy between the the *requirements* of MAT and the *purpose* for which fundamental notions of MAS, game theory, and micro-economics have been conceived. This is particularly true for multi-agent mechanism design. As a case in point, we discuss the deficiencies of three game-theoretic notions in the context of MAT: Pareto optimality, strategy proofness, and incentive compatibility.

Finally, we explain why mechanism design calls for another research paradigm than agent design. This is important for a proper understanding of our suggestions (cf. Sect. 5.9.2) on how to deal with speculative agent behavior in open systems.

Requirements for MAM, CMAT, and OMAT

In Sect. 1.3.2, we proposed a subdivision of the MAS research field into the research areas of MAM and MAT, and a further subdivision of MAT in CMAT and OMAT. The three areas MAM, CMAT, and OMAT call for different research attitudes, and different basic notions, because their respective purposes differ.

These differences are pictured schematically in Table 8.1. In view of the current popularity of game theory and micro-economics as concept repositories for MAS, these research fields are included in the table as well.

The table represents our own view on these research areas. The entries in column 2 (*nature*) indicate whether the research area is descriptive or normative. Most areas contain elements of both. In that case, the entry reflects the primary inclination. The entries in column 3 (*norm*) indicate what the (primary) norm is. Column 4 (*description target*) does the same for descriptive fields. Columns 5 and 6 indicate, for the MAS areas, to what extent descriptive accuracy, and computational efficiency are relevant. The entries for game theory and micro-economics in these columns indicate how much attention *is* actually paid. Here, (>) indicates that researchers in the area have argued that more attention is required.

The entries for the MAT research areas in the last two columns indicate that OMAT is inherently more difficult than CMAT, because it imposes demands on computational adequacy as well as descriptive accuracy. Comparison of these entries with the associated entries for game theory and micro-economics point out where problems can be expected if one attempt to use these fields as concept repositories for OMAT:

[1] An elaborate account of the inadequacy of DAI rationality notions for TR mechanisms can be found in (Lenting, 1999b).

research area	nature	norm	description target	relevance of descriptive accuracy	relevance of computational efficiency
MAM	descriptive	symbolic paradigm	human agents	major	minor
CMAT	normative	computational adequacy	—	minor	major
OMAT	normative	computational adequacy	computational agents	major	major
game theory	normative	mathematic soundness	human agents (?)	minor (>)	minor
micro-economics	descriptive	mathematic soundness	economic agents	major (>)	minor

Table 8.1: Differences between MAS subfields, game theory, and micro-economics

computational adequacy may prove to be a problem.

Deficiencies of some game-theoretic notions

In the past decade, many MAS researchers have employed notions from game theory (Rosenschein *et al.*, 1988; Ephrati & Rosenschein, 1991; Zlotkin & Rosenschein, 1992; Durfee *et al.*, 1993; Kraus *et al.*, 1995) and micro-economics (Wellman, 1994b; Huberman & Clearwater, 1995; Ygge, 1998). Below, we point out that many such notions have serious deficiencies in the context of MAT, due to the different nature and requirements of the respective fields.

Pareto optimality

At first sight, Pareto optimality seems perfectly suited as a target notion for mechanism design in MAT. Since it is formulated in ordinal, rather than cardinal terms, it is also applicable in contexts where cardinal expressibility is a problem. And how could it be wrong to require that an optimal solution be such that no agent can do better, without some other agent being worse off?

It is true that a solution which is not Pareto optimal is not optimal in the usual utilitarian sense either. However, in OMAT, optimality is a bit too much to ask. Especially for *large* open systems, such as national economies, or the Internet, Pareto optimality is far too strong a demand.

Curiously, it is also too weak, in small-scale as well as large-scale open systems. This becomes apparent if one considers an easy method to attain a Pareto-optimal allocation: allocating everything to a single agent. This allocation is Pareto optimal,[2] while it is obviously *far* from optimal in the relative-utilitarian sense.

Economic mechanisms such as the Walrasian auction attain more than just Pareto optimality. Their *core* solutions are significantly better than Pareto-optimal solutions: the above 'monocratic' allocation is not a core solution, and if the assumption of transferable utility is admissible, a core solution is *optimal* in the utilitarian sense.

Unfortunately, in open systems, it is often not realistic to assume transferable utility. Hence, in such systems, properties like Pareto optimality and core membership can only be used as *heuristics* to attain an acceptable level of utilitarian solution quality. Our research reveals that they are not very proficient in this respect. Even core membership appears to be a *mediocre* heuristic: On reassignment problems, the average solution quality of the core solutions rendered by the Walrasian exchange auction is below the average solution quality obtained with IG_0^∇. Since IG_0^∇ is among the least sophisticated reassignment mechanisms one can think of,[3] there seems to be little justification for the present-day prominence of Pareto optimality in MAS literature.

Strategy proofness

A mechanism is strategy proof if it is always best for each agent to be truthful in its communication, irrespective of how the other agents behave. As such, strategy proofness is, in principle, a useful concept for mechanism design in OMAT.

However, the demand for strategy proofness can conflict with other desirable outcome properties, such as Pareto optimality (Muller & Satterthwaite, 1985). An example in (Lenting, 1999a) reveals that there can also be repercussions in terms of utility loss, if one employs a relative-utilitarian solution quality criterion, and the possibility of mistakes, signal noise, or foul play cannot be excluded.

Incentive compatibility

In micro-economics, strategy proofness is less prominent than in social choice theory. The key notion in micro-economics is *incentive compatibility* (Hurwicz, 1986; Myerson, 1985). A mechanism is *incentive compatible* if rational responses of the agents

[2]Formally, the Pareto optimality of this allocation hinges on the assumption of monotonic preferences (i.e., "more is always better", but this is a very common assumption.

[3]In IG_0^∇, the agents simply submit the proposals which reflect their first preferences, without paying any attention to their chances of success, and the rules of the composition protocol are equally indiscriminate: For any tool type that is scarce, all contenders have equal chances to obtain the tool.

always lead to a final outcome with certain desirable properties, such as Pareto optimality and individual rationality. Incentive compatibility is similar, but slightly weaker than strategy proofness. Where the property of strategy proofness can be described concisely as "Truth is a dominant equilibrium", the essence of incentive compatibility is "Truth is a Nash equilibrium".

Incentive-compatibility is defined in the context of a mechanism with a central mediator. The mediator asks the agents to reveal their private information. It then computes a global solution that it deems acceptable in face of its (now complete, but not necessarily correct) knowledge of the problem, and finally suggests actions to the agents that will bring about the desired solution. To obtain incentive compatibility, the suggestions of the mediator must be such that none of the agents have an incentive to lie about their private information, or to deviate from the suggested action. This implies that the mediator must perform considerably *more* computation than would be required to solve the whole problem in a centralized fashion (Myerson, 1985, p.245). Again, we have stumbled on an aspect of a game-theoretical notion which is not a problem in the descriptive context of micro-economics, but a serious deficiency in MAT.

Agent design versus mechanism design

Differences in research requirements exist even within MAT. In particular, the requirements of mechanism design differ from those of agent design. In Sect. 5.9.2, we argued that the IG agent model of fictitiously rational agents with weariness can be turned into an admissible model for external agents in open systems by a suitable definition of the (stochastic) deadline. The idea behind this conjecture deserves some attention.

The main idea of mechanism design for OMAT is to ensure that autonomous, self-centered agents behave in a globally desirable manner by providing suitable incentives. In IG, the main incentive is the uncertainty which the agents face if they take part in an IG auction. The fact that they must make commitments under uncertainty appears to suppress certain undesirable behavior (viz. trust formation).

Our proposal to employ a stochastic deadline to prevent agents from exhibiting behavior that deviates from the IG agent model is based on the same idea. It entails using added uncertainty as a roadblock for agents which might otherwise be inclined to speculate on the proposal profile, or even on the utilities of their fellow agents, to improve on their expected utility. This idea is similar to that of encryption: There exists no encryption code that cannot be broken, but one can come up with algorithms that guarantee deciphering to be extremely costly on average (Herschberg, 1998).

Speculation is worth preventing. This is not only true for IG, but also for MAT mechanism design in general. From the (global) point of view of a mechanism designer, speculation amounts to wasting computational resources, since it is unlikely to render any *global* utility gain. Moreover, if agents speculate with the purpose to *manipulate* other agents, a course of events similar to an arms race may unfold, in which agents are drawn ever deeper into the swamp of reciprocal reasoning, to prevent other agents from pulling their leg.

Our conviction that speculation should be avoided is not shared by all MAS researchers. Indeed, from the point of view of agent design, speculation is an opportunity, rather than a problem. In computer chess, *opponent modeling* involves speculation on the depth to which the opponent searches the game tree (Gao *et al.*, 1999). The potential gains are obvious. Within MAS, speculation is welcomed as an opportunity with even greater enthusiasm. The Recursive Modeling Method (Gmytrasiewicz & Durfee, 1995; Vidal & Durfee, 1996) does not merely strive for insight into the potential gain of speculation on agent behavior, but also aims to investigate the gain of deeper-level reciprocal reasoning (cf. Sect. 5.8.1). At present, one has not gone beyond level-2 speculations (Vidal & Durfee, 1996), studying the gain a single agent can obtain if it has a perfect model of the other agents, who are only able to learn from observations. As such, there are many more levels to go...

Even though the ultimate goal of RMM is unclear, studying the gains that can be obtained from reciprocal reasoning at various depths is, in principle, a viable course of research if one is engaged in agent design. However, as mechanism designers, we hope to obviate such research in the near future.

The essence of IG

As some readers may have noticed, the above description of the game-theoretic notion of incentive compatibility is very similar to our earlier account on Myerson's interpretation of correlated equilibria (on page 209).

In our characterization of IG,[4] we described its equilibria as correlated equilibria, except that the strategy recommendations of the fully informed mediator are replaced by the probabilistic assertions which the agents derive from their observation of a shared aggregation device (viz. the IG auctioneer). Where the mediator messages are the glue from which the correlated equilibrium is formed, proposal success probabilities are the seeds of correlated agent behavior in IG.

A similar comparison can be drawn between the above mediation scenario for incentive compatibility and our proposals on how one should stimulate agent designers to endow their agents with fictitious rationality and weariness. As in IG itself, we sought the solution in confronting the agents with (more) uncertainty, in the form

[4]This pertains to the original conception of IG, in which the agents do not exhibit negotiation weariness.

of a stochastified deadline parameter.

Hence, the essence of Informed Gambling can be described as "an alternative to the game-theoretic implementation of incentive compatibility, based on the use of uncertainty as an incentive". The main advantage of IG-style incentive compatibility over the game-theoretic version, is that is does not conflict with the MAT requirements of informational and operational decentralization.

Appendix: Distributed Pseudo-composition Protocols

Response Protocol for Buyers

```
CASE m.type OF

"election":
  r := m.tooltype
  IF items-left[ r ] = 0 THEN
    send-to( m.sender, ( my-address, retraction, m.option-id ) )
  ELSE
    send-to( m.sender, ( my-address, commitment, m.option-id ) )
    decrement( items-left[ r ] )
  FI

"rejection":
/* NO RESPONSE REQUIRED */

ESAC
```

Response Protocol for Sellers

```
CASE m.type OF

"option":
 r := m.tooltype
 IF items-left[ r ] = 0 THEN
  send-to( m.sender, ( my-address, rejection, m.option-id ) )
 ELSE
  IF scarcity[ r ] <= 0 THEN
   IF election-counter[ r ] < items-left[ r ] THEN
    send-to( m.sender, ( my-address, election, m.option-id ) )
    increment( election-counter[ r ] )
   ELSE
    push( option-queue[ r ], m )
   FI
  ELSE /* SCARCE RESOURCE */
   push( option-queue[ r ], m )
   increment( option-counter[ r ] )
   IF option-counter[ r ] = demand[ r ] THEN
    WHILE election-counter[ r ] < items-left[ r ] DO
     m2 := best-option( option-queue[ r ] )
     remove-from( option-queue[ r ], m2 )
     send-to( m2.sender, ( my-address, election, m2.option-id ) )
     increment( election-counter[ r ] )
    OD
   FI
  FI
 FI

"commitment":
 r = m.tooltype
 decrement( items-left[ r ] )

"retraction": see next page
```

Response Protocol for Sellers (cont.)

```
"retraction":
  r := m.tooltype
  IF empty( option-queue[ r ] ) THEN
    decrement( election-counter[ r ] )
  ELSE
    IF scarcity[ r ] > 0 THEN /* SCARCE RESOURCE */
      m2 := best-option( option-queue[ r ] )
      remove-from( option-queue[ r ], m2 )
    ELSE
      m2 := pop( option-queue[ r ] )
    FI
    send-to( m2.sender, ( my-address, election, m2.option-id ) )
  FI
ESAC
```

References

Aarts, E. & Korst, J. 1989. *Simulated Annealing and Boltzmann Machines.* John Wiley & Sons.

Aliprantis, C.D., Brown, D.J. & Burkinshaw, O. 1989. *Existence and Optimality of Competitive Equilibria.* Springer Verlag.

Allen, B. & Hellwig, M. 1989. The Approximation of Competitive Equilibria by Bertrand-Edgeworth Equilibria in Large Markets. *Journal of Mathematical Economics,* **18**, 103–127.

Artstein, Z. & Wets, R. J-B. 1989. Decentralized Allocation of Resources among Many Producers. *Journal of Mathematical Economics,* **18**, 303–324.

Aumann, R.J. 1974. Subjectivity and Correlation in Randomized Strategies. *Journal of Mathematical Economics,* **1**, 67–96.

Aumann, R.J. 1987. Correlated Equilibrium as an Expression of Bayesian Rationality. *Econometrica,* **55**(1), 1–18.

Banâtre, J-P. & Le Métayer, D. 1993. Programming by Multiset Transformation. *Communications of the ACM,* **36**(1), 98–110.

Baye, M. & Kovenock, D. 1993. Rigging the Lobbying Process: An Apllication of the All-Pay Auction. *American Economic Review,* **83**, 289–294.

Bertsekas, D.P. & Tsitsiklis, J.N. 1988. Distributed Asynchronous Algorithms. *Pages 591–593 of: Proceedings of the 1988 IEEE Conference on Systems, Man, and Cybernetics,*, vol. 1.

Bond, A.H. & Gasser, L. 1988a. An Analysis of Problems and Research in DAI. *Pages 3–35 of:* Bond, A.H. & L.Gasser (eds), *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann.

Bond, A.H. & Gasser, L. (eds). 1988b. *Readings in Distributed AI.* Morgan Kaufmann.

Brown, G.W. 1951. Iterative Solutions of Games by Fictitious Play. *In:* Koopmans, T.C. (ed), *Activity Analysis of Production and Allocation.* Wiley & Sons, New York.

Castelfranchi, C. 1997. Modeling Social Action for AI Agents. *Pages 1567–1576 of: Proceedings IJCAI-97.* Morgan-Kaufmann.

Cohen, P.R. & Levesque, H.J. 1987. Persistence, Intention and Commitment. *In:* Georgeff, M.P. & Lansky, A.L. (eds), *Reasoning about Actions and Plans, Proceedings of the 1986 Workshop.* Morgan-Kaufmann.

Damme, E. 1991. *Stability and Perfection of Nash Equilibria.* Springer Verlag.

Decker, K., Sycara, K. & Williamson, M. 1997. Middle Agents for the Internet. *Pages 578–583 of: Proceedings IJCAI-97.*

Derks, J. & Kuipers, J. 1996. *Implementing the simplex method for computing the prenucleolus of tranferable-utility games.* Tech. rept. M96-06. Dept. of Mathematics, University of Maastricht, Netherlands.

Durfee, E.H. 1988. *Coordination of Distributed Problem Solvers.* Kluwer Acadamic Publishers.

Durfee, E.H. 1995. Blissful ignorance: knowing just enough to coordinate well. *Pages 406–413 of: Proceedings ICMAS-95.* AAAI Press.

Durfee, E.H., Lee, J. & Gmytrasiewics, P.J. 1993. Overeager Reciprocal Rationality and Mixed Strategy Equilibria. *Pages 109–114 of: Proceedings of the 12th Int. Workshop on Distributed Artificial Intelligence.*

Ephrati, E. & Rosenschein, J.S. 1991. The Clarke Tax as a Consensus Mechanism Among Automated Agents. *Pages 173–178 of: Proceedings AAAI-91.*

Finin, T., Labrou, Y. & Mayfiled, J. 1996. *KQML as an agent communication language.* MIT Press.

Gale, D. 1984. Equilibrium in a Discrete Exchange Economy with Money. *International Journal of Game Theory,* **13**(1), 61–64.

Gao, X., Iida, H., Uiterwijk, J.W.H.M., Herik & H.J. 1999. A Speculative Strategy. *In: Proceedings of the First Int. Conf. on Computers and Games (CG'98).* Electrotechnical Laboratory, Tsukuba, Japan.

Gasser, L. & Huhns, M. (eds). 1989. *Distributed Artificial Intelligence.* Pitman, London.

Genesereth, M.R. & Nilsson, N.J. 1987. *Logical foundations of artificial intelligence.* Kaufmann, Palo Alto, CA.

Gmytrasiewicz, P.J. & Durfee, E.H. 1995. A Rigorous, Operational Formalization of Recursive Modeling. *Pages 125–132 of: Proceedings ICMAS-95.* AAAI Press.

Grimaldi, R. P. 1985. *Discrete and Combinatorial Mathematics.* Addison-Wesley.

Harsanyani, J.C. 1968. Games with Incomplete Information Played by Bayesian Players. *Management Science*, **14**, 159–189.

Herschberg, I.S. 1998. Al Goed. *Pages 201–216 of:* van den Herik, H.J. & Lindgren, E. Roos (eds), *Bewaar me*. Van Spijk Publ., Venlo, the Netherlands.

Hewitt, C. 1991. Open Information Systems Semantics for Distributed Artificial Intelligence. *Artificial Intelligence*, **47**, 79–106.

Hildenbrand, W. & Kirman, A.P. 1976. *Introduction to Equilibrium Analysis*. Vol. 6. North-Holland Publishers, Amsterdam.

Hildenbrand, W. & Kirman, A.P. 1988. *Equilibrium Analysis*. Vol. 28. Elsevier Science Publishers, Amsterdam.

Hoel, P.G. 1984. *Introduction to Mathematical Statistics*. Wiley & Sons, New York.

Huberman, B.A. & Clearwater, S. 1995. A multi-agent system for controlling building environments. *Pages 171–176 of: Proceedings ICMAS-95*. AAAI Press/MIT Press.

Huhns, M.N. (ed). 1987. *Distributed Artificial Intelligence*. Morgan Kaufmann.

Hurwicz, L. 1986. Incentive Aspects of Decentralization. *In:* Arrow, K.J. & Intriligator, M.D. (eds), *Handbook of Mathematical Economics*, vol. 3. Elsevier Science Publishers, Amsterdam.

Jonker, R. & Volgenant, A. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, **38**, 325–340.

Joosten, R. 1996. *Dynamics, Equilibria, and Values*. Ph.D. thesis, University of Maastricht, The Netherlands.

Jordan, J.S. 1993. Three Problems in Learning Mixed-Strategy Equilibria. *Games and Economic Behavior*, **5**, 368–386.

Kaneko, M. 1982. The Central Assignment Game and the Assignment Market. *Journal of Mathematical Economics*, **10**, 205–232.

Kelly, J.S. 1978. *Arrow Impossibility Theorems*. Academic Press, New York.

Knuth, D.E. 1997. *Fundamental Algorithms*. Vol. 1. Addison-Wesley.

Kraus, S., Wilkenfeld, J. & Zlotkin, G. 1995. Multiagent Negotiation Under Time Constraints. *Artificial Intelligence*, **75**(2), 297–346.

Kreyszig, E. 1970. *Introductory Mathematical Statistics*. Wiley & Sons, New York.

Kurose, J.F. & Simha, R. 1989. A Microeconomic Approach to Optimal Resource Allocation in Distributed Computer Systems. *IEEE Transactions on Computers*, **38**(5), 705–717.

Kuwabara, K. & Ishida, T. 1992. Symbiotic Approach to Distributed Resource Allocation: Toward Coordinated Balancing. *In:* Cesta, A., Conte, R. & M., Miceli (eds), *Proceedings MAAMAW'92*. National Research Council, Institute of Psychology, Rome, Italy.

Lenting, J.H.J. 1992a. Impediments to composability in low-precision simulation. *Pages 179–183 of:* Luker, P. (ed), *Proceedings of the Summer Computer Simulation Conference 1992*. Reno, Nevada: Society for Computer Simulation, San Diego, CA.

Lenting, J.H.J. 1992b. On the virtue of unlimited parallelism in a neural network. *Pages 278–284 of:* Krug, W. & Lehmann, A. (eds), *Proceedings of the European Simulation Symposium ESS'92*. Dresden, Germany: Simulation Councils, Inc.

Lenting, J.H.J. 1995. Representation Issues in Boltzmann Machines. *Pages 131–144 of:* Braspenning, P. J., Thuysman, F., & Weijters, T. (eds), *Artificial Neural Networks, An Introduction to ANN Theory and Practice*. Lecture Notes in Computer Science, no. 931. Springer Verlag.

Lenting, J.H.J. 1999a. *Drawbacks of Pareto optimality, strategy proofness, and incentive compatibility in the context of MAS mechanism design*. in preparation. Dept. of Comp. Sc., Univ. of Maastricht, the Netherlands.

Lenting, J.H.J. 1999b. *On the inadequacy of DAI notions of rationality for multiagent technology*. in preparation. Dept. of Comp. Sc., Univ. of Maastricht, the Netherlands.

Lenting, J.H.J. & Braspenning, P.J. 1993. Delegated Negotiation for Resource Re-Allocation. *Pages 299–312 of: Proceedings of the 16th German AI Conference (GWAI-92)*. Lecture Notes in Artificial Intelligence, no. 671. Springer Verlag.

Lenting, J.H.J. & Braspenning, P.J. 1994. An all-pay auction approach to distributed reallocation. *Pages 259–263 of: Proceedings ECAI'94*. Wiley & Sons, New York.

Lumer, E. & Huberman, B.A. 1990. *Dynamics of Resource Allocation in Distributed Systems*. Tech. rept. SSL-90-05. Systems Sciences Laboratory, Xerox PARC.

Ma, J. 1994. Strategy-Proofness and the Strict Core in a Market with Indivisibilities. *International Journal of Game Theory*, **23**, 75–83.

Maskin, E.S. 1985. The Theory of Implementation in Nash Equilibrium: A Survey. *Pages 173–204 of:* Hurwicz, L., Schmeidler, D. & Sonnenschein, H. (eds), *Social Goals and Social Organization*. Cambridge University Press.

Mertens, J-F. & Zamir, S. 1985. Formalization of Bayesian Analysis for Games with Incomplete Information. *International Journal of Game Theory*, **14**, 1–29.

Miao, X.Y., Lu, P.B. & Kleinman, D.L. 1988. Distributed Stochastic Resource Allocation in Teams. *Pages 445–448 of: Proceedings of the 1988 IEEE Conference on Systems, Man, and Cybernetics.*, vol. 1.

Milgrom, P.R. 1985. The Economics of Competitive Bidding: A Selective Survey. *Pages 261–289 of:* Hurwicz, L., Schmeidler, D. & Sonnenschein, H. (eds), *Social Goals and Social Organization.* Cambridge University Press.

Moret, B.M. 1998. *The theory of Computation.* Addison Wesley.

Mukherji, A. 1990. *Walrasian and Non-Walrasian Equilibria.* Clarendon Press, Oxford.

Muller, E. & Satterthwaite, M.A. 1985. Strategy-Proofness: The Existence of Dominant-Strategy Mechanisms. *Pages 131–171 of:* Hurwicz, L., Schmeidler, D. & Sonnenschein, H. (eds), *Social Goals and Social Organization.* Cambridge University Press.

Myerson, R.B. 1985. Bayesian Equilibrium and Incentive Compatibility: An Introduction. *Pages 229–259 of:* Hurwicz, L., Schmeidler, D. & Sonnenschein, H. (eds), *Social Goals and Social Organization.* Cambridge University Press.

Neumann, J. von & Morgenstern, O. 1947. *Theory of Games and Economic Behavior.* Princeton University Press.

Papadimitriou, C. H. & Steiglitz, K. 1982. *Combinatorial Optimization.* Prentice Hall.

Plaat, A. 1996. *Research, Re:search & re-search.* Ph.D. thesis, Erasmus University Rotterdam, The Netherlands.

Postlewaite, A. 1985. Implementation via Nash Equilibria in Economic Environments. *Pages 205–228 of:* Hurwicz, L., Schmeidler, D. & Sonnenschein, H. (eds), *Social Goals and Social Organization.* Cambridge University Press.

Quinzii, M. 1984. Core and Competitive Equilibria with Indivisibilities. *International Journal of Game Theory*, **13**(1), 41–60.

Robinson, J. 1951. An Iterative Method of Solving a Game. *Annals of Mathematics*, **54**, 296–301.

Rosenschein, J.S. 1993. Negotiation Mechanisms for Multi-Agent Systems. *Pages 792–799 of: Proceedings IJCAI'93.*

Rosenschein, J.S., & Genesereth, M.R. 1988. Deals Among Rational Agents. *In:* Huberman, B. (ed), *The Ecology of Computation.* Elsevier Science Publishers, Amsterdam.

Roth, A.E. & Postlewaite, A. 1977. Weak versus Strong Domination in a Market with Indivisible Goods. *Journal of Mathematical Economics*, **4**, 131–137.

Sandholm, T. & Ygge, F. 1997. On the Gains and Losses of Speculation in Equilibrium Markets. *Pages 2–1 of: Proceedings IJCAI-97.*

Sathi, A. & Fox, M. S. 1989. Constraint-Directed Negotiation of Resource Reallocations. *In:* Gasser, L. & Huhns, M. (eds), *Distributed Artificial Intelligence*, vol. 2. Pitman, London.

Scarf, H. 1973. *The Computation of Economic Equilibria.* Yale University Press.

Schmidhuber, J.H. 1996. A general method for incremental self-improvement and multi-agent reinforcement learning in unrestricted environments. *In:* Yao, X. (ed), *Evolutionary Computation: Theory and Applications.* Scientific Publ. Co., Singapore.

Sen, A. 1986. Social Choice Theory. *In:* Arrow, K.J. & Intriligator, M.D. (eds), *Handbook of Mathematical Economics*, vol. 3. Elsevier Science Publishers, Amsterdam.

Shapley, L. & Scarf, H. 1974. On Cores and Indivisibility. *Journal of Mathematical Economics*, **1**, 23–37.

Shoham, Y. 1993. Agent-Oriented Programming. *Artificial Intelligence*, **60**(1), 51–92.

Simon, H.A. 1955. A Behavioral Model of Rational Choice. *Quarterly Journal of Economics*, **59**, 99–118.

Singh, M.P. 1998. The Intentions of Teams: Team Structure, Endodeixis, and Exodeixis. *Pages 303–307 of: Proceedings ECAI-98.* Wiley & Sons.

Smith, R.G. 1980. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, 1104–1113.

Sycara, K.P., Roth, S.F., Sadeh, N. & Fox, M.S. 1991. Resource Allocation in Distributed Factory Scheduling. *IEEE Expert*, February.

Tadenuma, K. & Thomson, W. 1991. No-Envy and Consistency in Economies with Indivisible Goods. *Econometrica*, **59**(6), 1755–1767.

Takayama, A. 1985. *Mathematical Economics.* Cambridge University Press.

Vidal, J.M. & Durfee, E.H. 1996. The Impact of Nested Agent Models in an Information Economy. *Pages 377–384 of: Proceedings ICMAS-96.* AAAI Press.

Wako, J. 1984. A Note on the Strong Core of a Market with Indivisible Goods. *Journal of Mathematical Economics*, **13**, 189–194.

Wako, J. 1987 (May). *Strong Core and Competitive Equilibria of an Exchange Market with Indivisible Goods.* Tech. rept. Faculty of Economics, Tokyo Metropolitan University.

Walras, L. 1874. *Elements d'Économie Politique Pure.* Corbaz, Lausanne.

Wang, Q. & Parlar, M. 1989. Static Game Theory Models and their Application in Management Science. *European Journal of Operational Research*, **42**, 1–21.

Weber, R.J. 1985. Auctions and competitive bidding. *Pages 143–170 of:* Young, H.P (ed), *Fair Allocation: Proceedings of Symposia in Applied Mathematics*, vol. 33.

Wellman, M.P. 1992. A General-Equilibrium Approach to Distributed Transportation Planning. *Pages 282–290 of: Proceedings AAAI-92.*

Wellman, M.P. 1994a. A Computational Market Model for Distributed Configuration Design. *In: Proceedings AAAI'94.*

Wellman, M.P. 1994b. A Market-Oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems. *Journal of Artificial Intelligence Research*, **1**, 1–23.

Werner, E. 1989. Cooperating Agents: A Unified Theory of Communication and Social Structure. *Pages 3–36 of:* Gasser, L. & Huhns, M. (eds), *Distributed Artificial Intelligence*, vol. 2. Pitman, London.

Williams, C. & Hogg, T. 1992. Using Deep Structure to Locate Hard Problems. *Pages 472–477 of: Proceedings AAAI-92.*

Ygge, F. 1998. *Market-Oriented Programming and its Application to Power Load Management.* Ph.D. thesis, Lund University, Sweden.

Zlotkin, G. & Rosenschein, E.S. 1992. A Domain Theory for Task-Oriented Negotiation. *In:* Cesta, A., Conte, R. & M., Miceli (eds), *Proceedings MAAMAW'92.* National Research Council, Institute of Psychology, Rome, Italy.

Definition Index

Summary

This thesis reports on research in the area of Multi-Agent Systems (MAS). Its subject is the design of *multi-agent mechanisms*, i.e., computer programs in which computational agents solve a problem collectively, by interacting with a mediating agent or a coordination device. More specifically, the thesis addresses two research questions:

1. How can we solve a certain class of problems with a multi-agent mechanism?
2. How do coordination rules and agent behavior characteristics influence the performance of a multi-agent mechanism?

To answer these questions, we use the following approach. We select a broad problem domain of practical relevance, and develop a mechanism design framework for this domain. We then use the framework to design mechanisms for a subdomain that is sufficiently simple to allow for a thorough systematic investigation of the influence of agent behavior characteristics and coordination rules on mechanism performance.

The overall problem domain for which we develop the Informed Gambling (IG) framework is *tool reallocation* (TR). In this context, the concept 'tool' is a generalization of the tools we use in everyday life, such as hammers, toothbrushes, and computers. Such goods typically have the following two properties. First, they are *typed* goods: unless your hammer is an archaeological find, there exist different, but completely equivalent hammers in the store where you bought it, or in other people's homes. Second, tools are *indivisible* goods. Tool exchanges between users usually involve few — often only one — tools of the same type. Consequently, the trade volumes in tool exchanges can generally not be cut into a number of equal portions. This distinguishes tools like hammers and computers from divisible goods, such as water and gasoline. In the context of TR, we label *any* good that is indivisible and typed as a tool. Thus, next to reallocation of everyday tools, TR also covers problems like the reallocation of reservations on trains or airplanes. Grace to this generalization, the TR domain has considerable practical relevance: many discrete optimization problems can be reformulated as TR problems. As an example, transportation in networks with finite-capacity links can be reformulated as a TR problem where each tool is a unit of transportation capacity across some link.

Our IG framework for TR mechanism design is based on an economic metaphor. Agents in an IG mechanism act as entrepreneurs in a tool exchange market. They submit tool exchange proposals to an auctioneer, who will ultimately employ a

fixed decision protocol to deal with conflicting proposals. The decision rules of this protocol resolve conflicts between proposals by defining an eligibility order on proposals. The order is based on the scarcity (i.e., demand minus supply) of the tool type offered in a proposal. Invocation of the protocol results in acceptance of the most eligible proposals, and adaptation or rejection of the remaining ones (if any).

Before the protocol is invoked, the agents get multiple opportunities to adjust their proposals so as to reduce the risk of unfavorable adaptation or rejection. To arrive at sensible proposal adjustments, the agents can use their knowledge of the protocol, and a survey of tool supply and demand, broadcasted by the auctioneer. The agents use this information to estimate acceptance probabilities for candidate proposals. This enables them to select proposals which maximize or enhance[5] the expected utility of the outcome (i.e., their expectation of their satisfaction with their final endowment). These two steps (proposal submission and broadcast of supply/demand information) are repeated until all agents stick to their last proposals. The decision protocol is then invoked to compute and execute a feasible reallocation. If some agents are dissatisfied with the outcome, the entire process is repeated with the dissatisfied agents.

Proposal adaptation by the auctioneer entails that the agent receives, in exchange for the tool offered in its proposal, a tool that is selected randomly from the oversupplied tools. This implies that agents often run at least some risk of ending up with a tool endowment that is *less* valuable to them than their initial endowment. In this respect, IG differs fundamentally from other economically inspired multi-agent approaches, such as Market-Oriented Programming (MOP).

MOP tends to perform very well on optimization problems with continuous variables, but its application to discrete problems is problematic. In the face of discreteness — that is, in economies with indivisible goods — equilibria often do not exist, and even if they do, it tends to be difficult to steer the artificial economy toward an equilibrium state. Furthermore, in economies with equilibria, the associated allocations need not correspond with optimal solutions. In fact, the allocation quality — defined as the average satisfaction of agents with their endowments — can be arbitrarily low, unless real or artificial money can be used as a filler to turn the discrete problem into a continuous one. To ensure that money is effective in this respect, the economy must have a property known as "transferable utility". This property entails that (1) every agent is willing to sell any good if the price is sufficiently high, and (2) every agent is sufficiently rich to buy any endowment of goods which she values more than any other agent. Obviously, transferable utility is never a problem in closed systems, where the system designer can simply specify how an agent should behave. However, in open systems (such as mechanisms to coordinate *existing* computational agents), utility is often not transferable.

[5] In complex reallocation problems agents strive for a high, but not necessarily maximal utility.

One advantage of the IG framework is that it does not require transferable utility, because IG employs uncertainty where MOP employs money. The functionality of money in MOP is to allow for comparison of the utilities attached to some tool by different agents, without *communicating* any such utilities. The idea is that an agent who attaches a relatively high utility to some tool is willing to pay a relatively high *price* for it. In IG, the basic idea is that an agent who attributes a relatively high utility to some tool is willing to take a relatively high *risk* to obtain it.

The IG framework aims to be suited for closed as well as open mechanisms. The design of an open mechanism is more difficult than that of comparable closed mechanisms. Because the designer of an open mechanism has no control over the behavior of the external agents, she faces the additional constraint that any assumptions on agent behavior made to further computational or design efficiency must be *realistic*.

In fundamental MAS research, one usually translates this constraint to the assumption that the behavior of an existing agent is *perfectly rational*, that is, completely determined by the strife to obtain an endowment of maximal utility (to itself), without regard for the effort required to achieve this. I postulate that a greater regard for the computational cost of decision making, coupled with a shift of attention from the rationality of computational agents to the rationality of *human agent designers*, can lead to the conception of open negotiation mechanisms with a better overall performance. This postulate is the background of my claim that the IG agent model for open systems is preferable to the standard model of perfectly rational agents.

The agent behavior specified in the IG agent model for simple TR problems is a combination of negotiation weariness and fictitious rationality. Agent weariness amounts to an indifference among agents toward small utility gains that is growing with time. Its purpose is to ensure timely termination of the negotiation process. Fictitious rationality differs from perfect rationality in that a fictitiously rational agent (1) ignores information that is principally there, but difficult to use, and (2) does not contemplate the possibility of attempting to manipulate the other agents. The rationale for these differences is that, with the kind of uncertainty which IG agents are facing, perfect rationality incurs excessive design and computational complexity, while the reward of using perfect instead of fictitious rationality (in terms of extra utility) is often low, possibly nil, and typically difficult to estimate in advance.

In the IG framework, we turn this rationale into a design principle for open systems: a key activity in designing open IG mechanisms is the conception of ways to *increase* the design complexity and the reward uncertainty associated with perfect rationality, so as to keep agent designers from developing agent behavior schemes which deviate from the relatively simple scheme of the IG agent model. Thus, uncertainty is used to turn agent behavior that may well be irrational from the agent's point of view into a behavior specification that is rational from the viewpoint of an agent *designer*. This is antithetical to the ruling paradigm that the rationality of autonomous computational agents is a datum that can and should not be tampered with.

Applicability considerations have played an important role in the conception of IG. Yet, our research is *fundamental* rather than applied. Instead of focusing on a single, challenging real-life problem, we perform a systematic statistical analysis of mechanism performance on various sample spaces of TR problems. For such an analysis to be feasible, the complexity of computing an optimal solution for a problem instance must be low. This is why most of our theoretical, and all of our experimental research is confined to tool *reassignment*, the subdomain of tool reallocation where each agent possesses and desires only one tool. Contrary to full-fledged TR, tool reassignment is tractable.

Our experiments on the influence of agent characteristics and coordination rules on mechanism performance were carried out on a testbed with a parametrized IG reassignment mechanism as its core element. The investigated agent behavior characteristics are negotiation weariness and various aspects of the agents' attitudes toward risk. The investigated coordination rules are the criterion for proposal eligibility, the deadline for (i.e., the duration of) proposal relaxation, and the kind of parallelism employed in the relaxation process. The main conclusions drawn from these experiments are the following.

Conclusions with respect to research question 1:

1. Uncertainty can serve as an alternative to money, to reduce the solution quality decrease incurred by informational decentralization: IG performs significantly better than TR mechanisms that feature neither money nor uncertainty, such as Walrasian exchange.
2. Uncertainty is, however, less proficient than money in the context of transferable utility: in most cases, IG cannot ensure optimality.

Conclusions with respect to research question 2:

1. Agent weariness is a valuable characteristic. Both the frequency and the severity of utility concessions due to negotiation weariness tend to be marginal. This observation supports our conjecture that agent weariness is an admissible characteristic of agent behavior, even in open systems. Furthermore, IG mechanisms with weariness tend to render much better solutions than IG mechanisms without weariness in cases of very low deadline values. For high deadline values, the average solution quality is approximately the same. Thus, weariness allows for IG to be used as an anytime algorithm.

2. The kind of parallellism employed in the relaxation process has little influence on mechanism performance. If negotiation weariness is incorporated, even synchronously parallel proposal updating — which normally leads to severe convergence problems — is a feasible option.
3. It pays for agents to perform risk estimation. A community of agents who estimate proposal acceptance probabilities, and select a proposal which maximizes their *expected* utility tends to obtain better allocations than a community of totally reckless agents, who simply *assume* that the proposal describing their preferred exchange will be accepted.
4. In general, a pessimistic agent attitude toward the probability of proposal acceptance is more detrimental to allocation quality than an optimistic attitude, but it depends on the difficulty of the problem (more specifically, on the initial discrepancy between tool demand and supply) what the best attitude toward risk is. For average problems, with only moderate discrepancy between supply and demand, a risk-neutral attitude (i.e., unbiased estimation of proposal acceptance probabilities) appears to be best, but for problems with large discrepancies, a considerably optimistic attitude is profitable.
5. Demanding that reallocations must be individually rational (i.e., demanding that agents should never end up with a final endowment that is *less* satisfactory than their initial one. appears to be much more costly in strongly informationally decentralized algorithms than in centralized ones. If IG agents do not take any risk, the solution is individually rational, but in this case, the *average* agent satisfaction drops considerably, to the level of other mechanisms that guarantee individual rationality, such as Walrasian exchange. In contrast, imposing the constraint of individual rationality in a *centralized* allocation algorithm incurs hardly any decrease of the average agent satisfaction.
6. The proposal eligibility criterion has little influence on mechanism performance. Contrary to our expectation, it matters very little which proposal selection heuristic is employed to decide between conflicting proposals. Even more surprising is the observation that, on average problems, the highest average allocation quality is obtained by using no heuristic at all (i.e., by choosing randomly between conflicting proposals). In problems with large discrepancies between tool supply and demand, the use of heuristics does lead to better allocations. In all of these cases, however, the differences in allocation quality are marginal.

As for future research, the conclusions of the thesis call for additional fundamental as well as applied research. Additional fundamental research is required on the use of uncertainty in mechanism design for open systems. Future applied research is required on the actual application of IG to real-life discrete optimization problems.

Samenvatting

Dit proefschrift beschrijft onderzoek naar het ontwerpen van multi-agent mechanismen. Dit zijn computerprogramma's waarin computationele agenten[6] gezamelijk een vraagstuk oplossen door *indirecte* onderlinge communicatie, via een bemiddelende agent of een coördinerend apparaat. De kerngedachte achter multi-agent mechanismen is dat complexe vraagstukken vaak gemakkelijker kunnen worden opgelost door er niet één, maar meerdere programma's op te zetten, die elk slechts een bepaald aspect voor hun rekening nemen. Een nadeel van een dergelijke aanpak is dat de agenten, als gevolg van hun beperkte blikveld en hun onafhankelijkheid, geneigd zijn met deeloplossingen aan te komen die onderling onverenigbaar zijn. Het is de kunst om coördinatieregels te vinden die dit kunnen voorkomen zonder het onafhankelijk opereren van de agenten te belemmeren.

In het proefschrift wordt in het bijzonder aandacht besteed aan de volgende twee onderzoeksvragen.

1. Hoe kan een bepaald type problemen worden opgelost met behulp van multi-agent mechanismen?

2. Wat is de invloed van coördinatieregels en agentgedrag-karakteristieken op de prestatie van een multi-agent mechanisme?

Onze aanpak om deze vragen te beantwoorden is de volgende. We kiezen een breed probleemgebied met praktische relevantie, en ontwikkelen een raamwerk voor het ontwerpen van multi-agent mechanismen voor dit gebied. Vervolgens gebruiken we dit raamwerk om mechanismen te ontwerpen voor een deelgebied van problemen die voldoende eenvoudig zijn om de tweede onderzoeksvraag rigoreus te onderzoeken.

Het brede probleemgebied waarvoor we de Informed-Gambling (IG) ontwerpmethodiek ontwikkelen betreft het (her)verdelen van ondeelbare goederen onder agenten. In het Engels duiden we dit gebied aan met de term "tool reallocation" (TR). Naast het toedelen van tastbare objecten aan menselijke agenten (bv. computers aan werknemers) omvat TR ook logistieke problemen waarin de 'goederen' niet tastbaar zijn en/of de agenten niet menselijk. Voorbeelden hiervan zijn het toewijzen van routes aan transporten en het aanpassen van werkroosters bij ziekmeldingen.

[6] Een computationele agent is een (onderdeel van een) computerprogramma dat, met een grote mate van onafhankelijkheid, in samenwerking of competitie met andere agenten, een eigen doel nastreeft.

Ons IG-raamwerk voor het ontwerp van TR-mechanismen is gebaseerd op de economische metafoor van ondernemers die opereren in een ruilmarkt, gecoördineerd door een veilingmeester. In een IG-mechanisme zijn de ondernemers computationele *agenten*. De veilingmeester daarentegen heeft meer weg van een *apparaat*, dat volgens vaste gedragsregels — die niet zijn ingegeven door eigenbelang — reageert op de biedingen van de ondernemers.

In IG dienen de agenten ruilvoorstellen in bij de veilingmeester, die reageert met een overzicht van vraag en aanbod van de verschillende typen goederen. Op basis van dit overzicht kunnen de agenten een inschatting maken van de kans dat een ingediend voorstel gehonoreerd zal worden en het eventueel vervangen door een ander voorstel, dat meer kans van slagen heeft. Dit proces van aanpassen van voorstellen en leveren van marktoverzichten gaat door totdat alle agenten vasthouden aan hun voorstel, of het aantal biedrondes een bepaalde bovengrens bereikt. Ook in de eindtoestand zijn doorgaans niet alle voorstellen onderling verenigbaar. De veilingmeester beslist dan, volgens vaste, bij de agenten bekende regels, welke van de voorstellen gehonoreerd zullen worden. Deze beslissingsregels komen neer op het definiëren van een volgorde van verkiesbaarheid, gebaseerd op de *schaarste* (d.i. vraag min aanbod) van het in een ruilvoorstel aangebodene. Een agent kan dus de kans om een schaars goed te bemachtigen vergroten door een ander schaars goed in ruil aan te bieden. Hierbij loopt hij echter wel een risico. Als de verkiesbaarheid van zijn ruilvoorstel onvoldoende hoog is (doordat mededingers nog schaarsere goederen aanbieden in hun voorstellen), is hij verplicht om het aangebodene te leveren, terwijl hij in plaats van het gevraagde iets ontvangt waarvan het aanbod de vraag overstijgt. Dit impliceert dat een agent die vasthoudt aan een voorstel om schaarse goederen te ruilen het risico loopt dat zijn eindsituatie *minder* bevredigend is dan zijn situatie vóór reallocatie. In dit opzicht verschilt IG fundamenteel van bestaande economisch geïnspireerde multi-agent raamwerken als Market-Oriented Programming (MOP).

MOP is een krachtige methode voor het gedecentraliseerd oplossen van veel logistieke problemen, maar het werkt niet altijd. IG richt zich met name op problemen waarvoor MOP niet geschikt is: problemen van discrete aard (d.w.z. met geheeltallige variabelen) waarin de aanname van *overdraagbare utiliteit* niet gemaakt kan worden. Deze aanname houdt in dat het logistieke probleem vertaald kan worden naar een markt waarin (1) iedere agent bereid is welk goed dan ook te verkopen als de prijs maar hoog genoeg is, en (2) iedere agent voldoende geld bezit om goederen te kopen die voor hem waardevoller zijn dan voor alle andere agenten. Overdraagbaarheid van utiliteit is nimmer een probleem in gesloten systemen, waar de systeemontwerper domweg kan specificeren hoe agenten zich hebben te gedragen. In open systemen is het echter vaak een te zware eis.

Overdraagbaarheid van utiliteit is geen vereiste in IG, omdat IG gebruik maakt van onzekerheid waar MOP gebruik maakt van geld. In MOP bewerkstelligt de aanwezigheid van geld dat de verschillen tussen de utiliteiten (d.w.z. de waarde) die verschillende agenten aan eenzelfde goed hechten tot uiting komen in de uiteindelijke herverdeling, *zonder* dat deze utiliteiten worden gecommuniceerd: Een agent die veel waarde hecht aan bepaalde goederen zal bereid zijn hiervoor een relatief hoge *prijs* te betalen. In IG wordt hetzelfde bereikt doordat een agent die veel waarde hecht aan een object bereid is een relatief groot *risico* te nemen om het object te verkrijgen.

Het IG-raamwerk is zowel geschikt voor open als voor gesloten systemen. Het ontwerp van mechanismen voor open systemen is moeilijker, omdat de ontwerper in dit geval geen controle heeft over het gedrag van de agenten. Zij moet er dus voor zorgen dat iedere wenselijke[7] aanname over agentgedrag ook een *realistische* aanname is.

De huidige trend in fundamenteel onderzoek naar multi-agent systemen is om deze extra randvoorwaarde te vertalen in de aanname dat het gedrag van een bestaande agent altijd *volkomen rationeel* is. Dit houdt in dat een agent altijd kiest voor een ruilvoorstel waarvan het te verwachten resultaat optimaal is, gezien de hem beschikbare informatie. Hierbij wordt de computationele complexiteit van het selecteren van een dergelijk ruilvoorstel genegeerd. Ik postuleer dat men tot betere mechanismen voor open systemen kan komen door zich te richten op de rationaliteit van de *ontwerpers* van agenten, in plaats van die van de agenten zelf. Als leidraad moet dienen dat het niet lonend mag zijn voor een ontwerper/programmeur om een agent te ontwerpen waarvan het gedrag afwijkt van dat van het IG-agentmodel.

Het gedrag beschreven in dit agentmodel is een combinatie van onderhandelingsmoeheid en iets dat ik "fictieve rationaliteit" noem. De onderhandelingsmoeheid houdt in dat IG-agenten in toenemende mate onverschillig worden met betrekking tot kleine verschillen tussen de te verwachten utiliteiten van ruilvoorstellen. Dit is van belang voor een tijdige beëindiging van het proces van aanpassing van de voorstellen. Fictieve rationaliteit houdt in dat IG-agenten (1) informatie die moeilijk te benutten is negeren, en (2) geen poging doen om andere agenten te manipuleren. Het motief voor deze gedragsaannames is dat volkomen rationaliteit, door de onzekerheid waarmee IG-agenten geconfronteerd worden, extreem arbeidsintensieve berekeningen vergt, terwijl de resulterende extra utiliteit voor de agent veelal laag, mogelijk nihil, en doorgaans niet goed vooraf in te schatten is.

In het IG-raamwerk wordt dit motief verheven tot een ontwerp-*beginsel* voor open systemen: Een hoofdaktiviteit in het ontwerpen van open IG-mechanismen is het scheppen van extra onzekerheden om te voorkomen dat ontwerpers agenten ontwerpen waarvan het gedrag afwijkt van het IG-agentmodel. Door extra onzekerheden in te bouwen in een IG-mechanisme kan zowel de ontwerp-complexiteit van (meer) volkomen rationaliteit als de onzekerheid omtrent de hieraan verbonden utiliteitswinst verhoogd worden. Dit bewerkstelligt dat het voor een agentontwerper

[7] wenselijk met het oog op het prestatienivo van het mechanisme of de eenvoud van het ontwerp

niet lonend — en dus irrationeel — is om het gedragsrepertoire van hun agenten af te laten wijken van de gedragsspecificatie van het IG-agentmodel, *ondanks* het feit dat het resulterende agentgedrag in sommige situaties irrationeel is vanuit het oogpunt van de agent. Deze ontwerpstrategie staat haaks op de heersende zienswijze dat de rationaliteit van autonome computationele agenten een gegeven is waaraan niet getornd mag of kan worden.

Toepasbaarheid is een belangrijke richtlijn geweest in de conceptie van IG. Het in dit proefschrift beschreven onderzoek is echter van fundamentele aard. Het richt zich niet op een enkel concreet praktijkprobleem, maar omvat een systematisch statistisch onderzoek naar de prestatie van IG-mechanismen op een klasse van meer abstracte TR-problemen. Om een dergelijke statistische prestatie-analyse mogelijk te maken is het noodzakelijk dat de onderzochte probleemklasse uit problemen bestaat waarvan de optimale oplossingen relatief gemakkelijk kunnen worden bepaald. Dit is de reden dat het merendeel van het theoretische en al het experimentele onderzoek beperkt is tot de klasse van herverdelingsproblemen waarin iedere agent slechts één object bezit, en dat hij zou willen ruilen tegen één andersoortig object.

De experimenten om de invloed van coördinatieregels en agentgedrag op de prestatie van mechanismen te onderzoeken zijn uitgevoerd met behulp van een testprogramma waarin een geparametriseerd IG-mechanisme de centrale component is. De onderzochte agentgedrag-parameters hebben betrekking op de onderhandelingsmoeheid en op verschillende aspecten van het omgaan met risico's. De parameters die betrekking hebben op de coördinatie door de veilingmeester omvatten het criterium voor de verkiesbaarheid van ruilvoorstellen, de maximale duur (in biedrondes) van het biedproces, en de aard van het parallellisme dat in dit proces wordt gebruikt.

De belangrijkste conclusies die uit de resultaten kunnen worden getrokken zijn de volgende.

Conclusies m.b.t. de eerste onderzoeksvraag:

1. Onzekerheid kan, in plaats van geld, gebruikt worden om het verlies van oplossingskwaliteit als gevolg van informationele decentralisatie te beperken. IG-mechanismen presteren in dit opzicht beter dan TR-mechanismen waarin geld noch onzekerheid een rol spelen.
2. Onzekerheid is echter minder adequaat dan geld in de context van overdraagbaarheid van utiliteit: In de meeste gevallen zijn de door IG opgeleverde allocaties goed, maar niet optimaal.

Conclusies m.b.t. de tweede onderzoeksvraag:

1. Onderhandelingsmoeheid is een waardevolle component van IG. Het blijkt dat een geringe dosis onderhandelingsmoeheid in de regel voldoende is

om een tijdige beëindiging van het proces van aanpassing van de ruilvoorstellen te bewerkstelligen, zonder dat de kwaliteit van de oplossingen hieronder lijdt. Een snel toenemende onderhandelingsmoeheid levert een zeer snelle beëindiging van het proces op — na pakweg 3 tot 5 ronden — met een oplossingskwaliteit die weliswaar lager is dan normaal, maar veel hoger dan wanneer men *zonder* onderhandelingsmoeheid zo'n kort aanpassingsproces gebruikt.

2. De aard van het parallellisme dat in het aanpassingsproces gebruikt wordt heeft geen noemenswaardige invloed op de prestatie van IG-mechanismen. In IG-mechanismen met onderhandelingsmoeheid levert zelfs *synchroon* parallelle verwerking van de ruilvoorstellen geen problemen op, terwijl dit in mechanismen zonder onderhandelingsmoeheid het bereiken van een stationaire toestand dikwijls onmogelijk maakt.
3. Risico-inschatting loont voor IG-agenten. In IG-mechanismen waarin agenten een poging doen om het risico van aanpassing of verwerping van hun voorstellen in te schatten is de kwaliteit van de oplossingen doorgaans aanzienlijk hoger dan in mechanismen waarin de agenten domweg aannemen dat hun voorstellen zonder aanpassing zullen worden geaccepteerd.
4. Een pessimistische inschatting van de acceptatiekansen van ruilvoorstellen is veel schadelijker voor de kwaliteit van de eindallocaties dan een optimistische inschatting. Het hangt echter af van de moeilijkheidsgraad van een herverdelingsprobleem (en met name van de initiële discrepantie tussen vraag en aanbod) wat de meest profijtelijke houding jegens risico is. Voor doorsnee problemen (d.w.z. problemen met enige, doch geen al te grote discrepantie tussen vraag en aanbod) blijkt risico-neutraal[8] agentgedrag de beste resultaten op te leveren. Voor relatief moeilijke problemen (met grote discrepantie) leidt een flinke dosis optimisme tot een aanzienlijk hogere gemiddelde oplossingskwaliteit dan risico-neutraal gedrag.
5. Het opleggen van de eis dat de herverdeling individueel-rationeel[9] dient te zijn blijkt aanzienlijk kostbaarder in de context van sterke decentralisatie van informatie[10] dan in gecentraliseerde systemen. Men kan in IG individuele rationaliteit garanderen door de agenten geen enkel risico te laten nemen. De kwaliteit van de oplossingen is met een dergelijk agentgedrag

[8]Risico-neutraal gedrag houdt in dat de agenten een ruilvoorstel kiezen zodat de zuivere verwachtingswaarde van de hieraan verbonden utiliteit maximaal is.

[9]In een individueel-rationele herverdeling gaat geen enkele agent erop achteruit.

[10]Sterke informatie-decentralisatie houdt in de context van herverdelingsproblemen in dat (1) informatie over de utiliteit die een agent hecht aan een pakket goederen niet gecommuniceerd kan of mag worden aan andere agenten, en (2) dat geen gebruik kan worden gemaakt van geld om ondanks deze informatiebeperking toch een afweging mogelijk te maken van de utiliteiten die verschillende agenten hechten aan eenzelfde pakket goederen.

echter aanzienlijk lager dan met het standaard (risico-neutrale) agentgedrag. Zij is bij benadering gelijk aan de oplossingskwaliteit die verkregen kan worden met andere mechanismen die individuele rationaliteit waarborgen, zoals een Walrasiaanse ruilmarkt. Dit verlies van oplossingskwaliteit staat in scherp contrast met het zeer marginale verlies dat gepaard gaat met het opleggen van individuele rationaliteit in gecentraliseerde herverdelingsalgoritmen.

6. De aard van het criterium voor de verkiesbaarheid van ruilvoorstellen blijkt nauwelijks van invloed te zijn op de kwaliteit van de oplossingen. Geheel tegen de verwachting in, maakt het nauwelijks uit of men een fijn of een grof criterium hanteert.[11] Nog verrassender is de observatie dat een IG-mechanisme *zonder* verkiesbaarheidscriterium (d.w.z. een mechanisme waarin de selectie van ruilvoorstellen volledig aan het toeval wordt overgelaten) op doorsnee problemen een *hogere* gemiddelde oplossingskwaliteit bereikt dan IG-mechanismen met een verkiesbaarheidscriterium. De heuristiek om agenten die een veelgevraagd goed kunnen leveren te stimuleren om "in de bieding te blijven" blijkt dus contraproduktief te zijn voor doorsnee problemen. Voor herverdelingsproblemen met een grote discrepantie tussen vraag en aanbod leveren IG-mechanismen met een verkiesbaarheidscriterium wel een iets hogere gemiddelde oplossingskwaliteit. De verschillen in oplossingskwaliteit zijn echter in alle onderzochte gevallen marginaal.

De uit het onderzoek getrokken conclusies geven aan dat verder onderzoek naar TR-mechanismen zinvol is. Dit betreft zowel fundamenteel als toegepast onderzoek. Extra fundamenteel onderzoek is gewenst met betrekking tot het gebruik van onzekerheid als middel om agenten in open systemen te weerhouden van gedrag dat vanuit globaal oogpunt schadelijk of nutteloos is. Toegepast onderzoek is nodig om te bepalen in hoeverre Informed Gambling een geschikte basis is voor het gedecentraliseerd oplossen van discrete optimaliseringproblemen uit de praktijk.

[11] Een fijn criterium houdt in dat de relatieve verkiesbaarheid van een ruilvoorstel afhangt van het antwoord op de vraag *hoe* schaars het in het voorstel aangebodene is; een grof onderscheid betekent dat alleen gekeken wordt *of* het aangebodene al dan niet schaars is.

Curriculum Vitae

Jacques Lenting werd op 19 juli 1957 geboren in Bergen op Zoom. In 1975 behaalde hij het diploma gymnasium B aan de Scholengemeenschap "De Amersfoortse Berg" in Amersfoort. In datzelfde jaar begon hij aan een studie wiskunde aan de Rijksuniversiteit Groningen. Tijdens de studie is hij werkzaam geweest als freelance toneelregisseur en als docent informatica bij het Noordelijk Informatica Opleidingen Instituut. De studie werd in 1987 afgesloten met een doctoraalexamen in de toegepaste wiskunde en informatica. In het verlengde van zijn afstudeeropdracht heeft hij tot 1988 onderzoek verricht bij het Academisch Ziekenhuis Groningen op het gebied van medische expertsystemen. In 1988 trad hij in dienst als toegevoegd onderzoeker bij de vakgroep Informatica van de Rijksuniversiteit Limburg, inmiddels herdoopt tot de Universiteit Maastricht. Tot 1990 heeft hij zich hier beziggehouden met contractresearch naar kwaliteitsmanagement van expertsystemen. Van 1990 tot 1994 was hij als onderzoeksmedewerker van de Stichting Nederlands Wetenschappelijk Onderzoek (NWO) gedetacheerd bij dezelfde werkgroep. In deze periode heeft hij onderzoek verricht naar multi-agent systemen, Boltzmann neurale netwerken, en computergestuurd probleemoplossen. De resultaten van het multi-agentonderzoek zijn verwerkt in dit proefschrift.

SIKS Dissertatiereeks

In 1999 zijn de volgende SIKS-dissertaties verschenen.

99-1 Mark Sloof (VU)
Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products
Promotor: Prof. dr. J. Treur
Co-promotor: Dr. ir. M. Willems
Promotie: 11 mei 1999

99-2 Rob Potharst (EUR)
Classification using Decision Trees and Neural Nets
Promotor: Prof. dr. A. de Bruin
Co-promotor: Dr. J.C. Bioch
Promotie: 4 juni 1999

99-3 Don Beal (Queen Mary and Westfield College)
The Nature of Minimax Search
Promotor: Prof. dr. H.J. van den Herik
Promotie: 11 juni 1999

99-4 Jacques Penders (KPN Research)
The Practical Art of Moving Physical Objects
Promotor: Prof. dr. H.J. van den Herik
Co-promotor: Dr. P.J. Braspenning
Promotie: 11 juni 1999

99-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems
Promotor: Prof. dr. R.A. Meersman
Co-promotor: Dr. H. Weigand
Promotie: 1 oktober 1999

99-6 Niek J.E. Wijngaards (VU)
Re-design of Compositional Systems
Promotor: Prof. dr. J. Treur
Co-promotor: Dr. F.M.T. Brazier
Promotie: 30 september 1999

99-7 David Spelt (UT)
Verification Support for Object Database Design
Promotor: Prof. dr. P.M.G. Apers
Co-promotor: Dr. H. Balsters
Promotie: 10 september 1999


0
00
2
4
6
8
9
11
12
14
15
17
18
1st 12
1-18
EVEN
12