NASA CR-73434-II

# STUDY OF SYSTEMS AND COST/PERFORMANCE METHODOLOGIES FOR OPTIMIZATION OF VEHICLE ASSIGNMENT

# FINAL REPORT

## VOLUME 2

## PROGRAMMER'S MANUAL
## VEHICLE ASSIGNMENT MACHINE MODEL



8 MAY 1970

PREPARED UNDER CONTRACT NAS2-5202

FOR

MISSION ANALYSIS DIVISION
OFFICE OF ADVANCED RESEARCH AND TECHNOLOGY
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA

BY

LOCKHEED MISSILES & SPACE COMPANY
SUNNYVALE, CALIFORNIA

"Distribution of this report is provided in the interest of information exchange Responsibility for the contents resides in the authors or organization that prepared it "

NASA CR-73434

# STUDY OF SYSTEMS AND COST/PERFORMANCE METHODOLOGIES FOR OPTIMIZATION OF VEHICLE ASSIGNMENT

FINAL REPORT

VOLUME 2

## PROGRAMMER'S MANUAL

## VEHICLE ASSIGNMENT MACHINE MODEL

8 MAY 1970

Prepared Under Contract NAS2-5202

By

LOCKHEED MISSILES & SPACE COMPANY
SUNNYVALE, CALIFORNIA

For

MISSION ANALYSIS DIVISION
OFFICE OF ADVANCED RESEARCH AND TECHNOLOGY
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA

# FOREWORD

This report volume is a programmer's manual for the vehicle assignment model developed during a study of cost and performance methodologies for optimal assignment of space vehicles to advanced space missions. This study is being performed for the National Aeronautics and Space Administration under Contract NAS2-5202, monitored by Mr. Robert Slye and Mr. Harold Hornby of the Mission Analysis Division of the Office of Advanced Research and Technology.

Individuals of Lockheed Missiles & Space Company, Sunnyvale, California, who contributed to this study are L. F Fox, project leader, C. J. Golden, key technical member; and M. A. Brunet.

CONTENTS

## SUMMARY

This document is Volume 2 of a three-volume series comprising a final report of the Study of Systems and Cost/Performance Methodologies for Optimization of Vehicle Assignment. This volume is a programmer's manual for the launch vehicle assignment program. Volume 1 of this series presents a technical description and Volume 3 provides details on a computer program for integrated budget smoothing and vehicle assignment.

This volume contains appendixes that provide model input requirements, a sample case, flow charts, and a program listing. At the beginning of each appendix, descriptive details and technical comments are provided to indicate any special instructions applicable to the use of that Appendix. In addition, the program listing, Appendix D, includes comment cards that state the purpose of each subroutine in the complete program and also describe operations performed within the subroutine.

# Appendix A
# ASSIGNMENT PROGRAM INPUT REQUIREMENTS

## A.1 GENERAL

This appendix includes a complete glossary of input terms and detailed format requirements. Variable names are listed in order of input in corresponding sections of use rather than alphabetically to make the glossary easier to use. Comments are also included that describe either external or internal restrictions associated with the variable. "Blank" spaces are indicated whenever variables not used in this program are input in these locations for the integrated program. The same data cards may thus be used for both programs as long as these blank spaces are not filled.

Figure A-1 illustrates the basic data deck layout for this program. Any section may be eliminated if there are no associated data. However, either a blank card must be inserted in place of the section or the control card must reflect no input for that section. If the control card is coded so no data are input for some section, then values input for the preceding case are automatically supplied. If no data are desired for any one section, then a blank card must be input to terminate that section. Stage performance data to be used in the stage-matching screen may be eliminated entirely, including the final blank card, if the stage-matching screen is not to be used. If this screen is used, then the stage cards must be ordered so that all stages in Class 1 precede those in Class 2, which precede those in Class 3, which precede those in Class 4. Stages not included in the matching screen follow those in the above classifications. If the matching screen is not used, the order of cards within each section is unimportant.

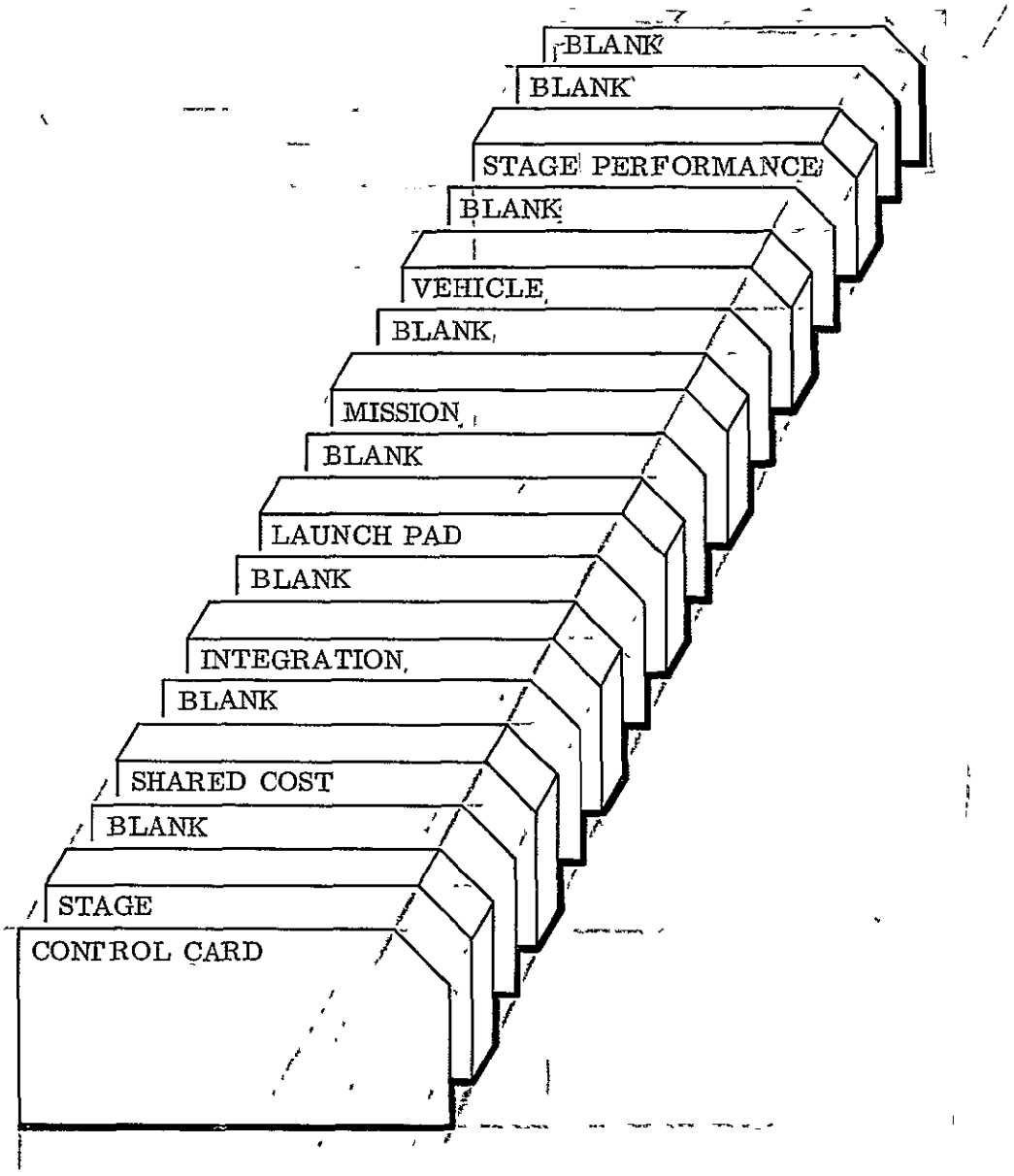



Fig. A-1 Data Deck Layout – Assignment Program

The deck terminates with two blank cards in all situations. Figure A-1 is set up for one case only. A second case would start with a control card following directly behind the blank card ending either the vehicle data section or the stage performance section, whichever is applicable. In general, for each case there must be a control card and the complete data deck must be followed by a blank card.

A 2 INPUT FORM AND DEFINITIONS

| Card Columns | Variable Name | Format | Description and Comments |
|---|---|---|---|
| Control Card | | | |
| 1 – 5 | NOPT | I5 | Code for mission/vehicle compatibility screen<br>1 – ΔV versus payload weight and availability plus a priori assignment<br>2 – Code 1 plus use stage-matching screen<br>3 – All criteria |
| 6 – 10 | MYRS | I5 | Mission model duration in years |
| 11 – 15 | IBY | I5 | Last 2 digits of 1st year of mission model |
| 16 – 27 | GUESS | F12.2 | Upper bound for total mission cost (if GUESS = 0.0, then GUESS is assigned a value 1.0 E10 Storage space is saved if a realistic value is input.) |
| 66 – 67 | IP | I2 | Code for pad input* |
| 68 – 69 | IS | I2 | Code for stage input* |
| 70 – 71 | IFM | I2 | Code for shared cost group input* |
| 72 – 73 | II | I2 | Code for integration cost input* |
| 74 – 75 | IM | I2 | Code for mission input* |
| 76 – 77 | Blank | 2X | Blank |
| 78 – 79 | IV | I2 | Code for vehicle input* |
| 80 | LP | I1 | Code for logic output; if > 0 print logic output, if = 0 no logic output |

*If ≥ 0, new input for this case. If < 0, use data from previous case

| Card Columns | Variable Name | Format | Description and Comments |
|---|---|---|---|
| Stage Cards (input only if IS ≥ 0) | | | I = 1, NSTG ≤ 40 cards |
| 1st Stage Card | | | |
| 1 – 2 | KODE(I) | I2 | Reference # of stage on card I |
| 4 – 7 | STG(I) | A4 | Name of stage on card I |
| 8 – 13 | SR(I, 1) | F6.3 | Hardware recurring or refurbishment cost for 1st unit of stage on card I |
| 14 – 19 | SR(I, 2) | F6.3 | ETR launch support recurring cost for 1st unit of stage on card I |
| 20 – 25 | SR(I, 3) | F6.3 | WTR launch support recurring cost for 1st unit of stage on card I |
| 26 – 30 | PLC(I, 1) | F5.3 | Hardware recurring or refurbishment cost learning curve % for stage on card I in decimal form (e g , 0 95) |
| 31 – 35 | PLC(I, 2) | F5 3 | ETR launch support recurring cost learning curve % for stage on card I in decimal form |
| 36 – 40 | PLC(I, 3) | F5 3 | WTR launch support recurring cost learning curve % for stage on card I in decimal form |
| 44 – 49 | SNR(I) | F6.3 | Development cost of stage on card I |
| 50 – 55 | SUS(I) | F6.3 | Sustaining cost of stage on card I |
| 56 – 58 | NYS(I) | I3 | 1st year stage on card I is available (1 ≤ NYS(I) ≤ MYRS)* |
| 59 – 61 | LSA(I) | I3 | Last year stage on card I is available |
| 62 – 64 | NBY(I) | I3 | Batching duration in years for stage on card I recurring cost |
| 65 – 76 | NFS(I, J), J = 1, 4 | 4I3 | KODEF of the shared cost groups (up to 4) to which stage on card I belongs |
| 78 | MODE(I, 1) | I1 | Code to indicate type of input for hardware recurring cost of stage on card I ** |
| 79 | MODE(I, 2) | I1 | Code to indicate type of input for ETR launch support recurring cost of stage on card I** |

*1 corresponds to year IBY.
**If = 0, learning curve type input. If ≠ 0, jump type input.

| Card Columns | Variable Name | Format | Description and Comments |
|---|---|---|---|
| 80 | MODE(I, 3) | I1 | Code to indicate type of input for WTR launch support recurring cost of stage on card I* |
| 2nd Stage Card | | | |
| 5 – 14 | SUSLS(I, 1) | F10 0 | Sustaining cost of ETR facility for stage I not to be duplicated at each pad |
| 15 – 24 | SUSLS(I, 2) | F10 0 | Sustaining cost at WTR facility for stage I not to be duplicated at each pad |
| 25 – 28 | NU(I) | I4 | Number of reusable units in initial purchase of component I. NU = 0 – unit is expendable. NU > 0 – estimate used by program directly |
| 29 – 35 | UPP(I) | F7 3 | Unit purchase price |
| 36 – 40 | RPLO(I) | F6.0 | Return payload weight in lbs for this component** (vehicle return payload = orbiter return payload) |
| 3rd, 4th and 5th Stage Cards Needed if MODE(I, J) ≠ 0 | | | |
| 5 – 14 | SRJ(LX, 1) | F10 3 | Total recurring cost for up to POJ of stage |
| 15 – 24 | SRJ(LX, 2) | F10 3 | Slope of line defining total recurring cost for over POJ of stage |
| 25 – 34 | SRJ(LX, 3) | F10 3 | Y-intercept of line defining total recurring cost for over POJ of stage |
| 35 – 44 | POJ(LX) | F10 3 | # of stages at which function defining total recurring cost changes slope |
| Last stage card must be followed by blank card | | | |
| Shared Cost Group Cards (Input only if IFM ≥ 0) I = 1, NFAM < 30 | | | |
| 1 – 2 | KODEF(J)=I | I2 | # of group on card J |
| 4 – 7 | FAM(I) | A4 | Name of group I |

*If = 0, learning curve type input. If ≠ 0, jump type input.
**At present any value ≥ 1 0 may be input to indicate that stage I is reusable.

| Card Columns | Variable Name | Format | Description and Comments |
|---|---|---|---|
| 8 – 17 | FMNR(I) | F10 0. | Development cost of group I |
| 18 – 27 | FMSUS(I) | F10.0 | Sustaining cost of group I |
| 28 – 37 | Blank | 10X | Blank |
| 38 – 47 | FMSLS(I, 1) | F10.0 | Sustaining cost at ETR for group I not to be duplicated at each pad |
| 48 – 57 | FMSLS(I, 2) | F10 0 | Sustaining cost at WTR for group I not to be duplicated at each pad |

Last group card must be followed by blank card.

Integration Cost Cards (Input only if II ≥ 0) I = 1, NCI < 40

| | | | |
|---|---|---|---|
| 3 – 5 | NFML(I) | I3 | KODEF of shared cost group which is lower member of integration pair I |
| 6 – 8 | NFMU(I) | I3 | KODEF of shared cost group which is upper member of integration pair I |
| 9 – 18 | RINT(I) | F10.0 | Recurring or refurbishment cost for 1st unit of integration I |
| 19 – 28 | PLCINT(I) | F10 0 | Recurring or refurbishment cost learning curve % for integration I in decimal form |
| 29 – 38 | DINT(I) | F10.0 | Development cost of integration I |
| 39 – 48 | SINT(I) | F10.0 | Sustaining cost of integration I |
| 49 – 58 | Blank | 10X | Blank |
| 59 – 68 | SINTLS(I, 1) | F10.0 | Sustaining cost at ETR for integration I not to be duplicated at each pad |
| 69 – 78 | SINTLS(I, 2) | F10 0 | Sustaining cost at WTR for integration I not to be duplicated at each pad |

Last integration cost card must be followed by blank card

Pad Cards (Input only if IP ≥ 0) I = 1, NP < 30

| | | | |
|---|---|---|---|
| 1 – 4 | KODEP(I) | I4 | # of pad complex on card I |
| 7 – 10 | PAD(I) | A4 | Name of complex on card I |
| 11 – 15 | NPERPD(I) | F5.0 | Maximum # of launches/year/pad possible at complex I |

| Card Columns | Variable Name | Format | Description and Comments |
|---|---|---|---|
| 2nd — 6th Cards needed for each pad complex (stage cost-data, 2/card)(J=1, 10) | | | |
| 3 — 4<br>43 — 44 | NPSTG(I, J) | I2 | KODE corresponding to J-th stage costs of complex I |
| 5 — 10<br>45 — 50 | PSTGD(I, J, 1) | F6.0 | J-th stage development cost of first pad in complex I |
| 11 — 16<br>51 — 56 | PSTGS(I, J, 1) | F6.0 | J-th stage sustaining cost of first pad in complex I |
| 17 — 22<br>57 — 62 | PSTGD(I, J, 2) | F6.0 | J-th stage development cost of second pad in complex I |
| 23 — 28<br>63 — 68 | PSTGS(I, J, 2) | F6.0 | J-th stage sustaining cost of second pad in complex I |
| 29 — 34<br>69 — 74 | PSTGD(I, J, 3) | F6.0 | J-th stage development cost of third pad in complex I |
| 35 — 40<br>75 — 80 | PSTGS(I, J, 3) | F6.0 | J-th stage sustaining cost of third pad in complex I |
| 7th — 9th cards needed for each pad complex (family cost data, 2/card)(J=1, 5) | | | |
| 3 — 4<br>43 — 44 | NPFAM(I, J) | I2 | KODEF corresponding to J-th family costs of complex I |
| 5 — 10<br>45 — 50 | PFAMD(I, J, 1) | F6.0 | J-th family development cost of first pad in complex I |
| 11 — 16<br>51 — 56 | PFAMS(I, J, 1) | F6.0 | J-th family sustaining cost of first pad in complex I |
| 17 — 22<br>57 — 62 | PFAMD(I, J, 2) | F6.0 | J-th family development cost of second pad in complex I |
| 23 — 28<br>63 — 68 | PFAMS(I, J, 2) | F6.0 | J-th family sustaining cost of second pad in complex I |
| 29 — 34<br>69 — 74 | PFAMD(I, J, 3) | F6.0 | J-th family development cost of third pad in complex I |
| 35 — 40<br>75 — 80 | PFAMS(I, J, 3) | F6.0 | J-th family sustaining cost of third pad in complex I |
| 10th and 11th cards needed for each pad complex (integration cost data, 3/card)(J=1; 5) | | | |
| 9 — 11<br>33 — 35<br>57 — 59 | NPINTL(I, J) | I3 | KODEF of lower group corresponding to J-th integration cost of complex I |

| Card Columns | Variable Name | Format | Description and Comments |
|---|---|---|---|
| 12 – 14, 36 – 38, 60 – 62 | NPINTU(I, J) | I3 | KODEF of upper group corresponding to J-th integration cost of complex I |
| 15 – 32, 39 – 56, 63 – 80 | PINTS(I, J, K), K=1, 3 | 3F6.0 | J-th integration sustaining cost of K-th pad in complex I |

Last pad card must be followed by blank card

Mission Cards (Input only if IM ≥ 0) I=1, NMIS ≤ 50

| | | | |
|---|---|---|---|
| 1 – 2 | KODEM(I) | I2 | # of mission on card I |
| 3 – 8 | MISNAM(I) | A6 | Name of mission on card I |
| 9 – 12 | PB(I) | F4.2 | Priority of mission on card I |
| 15 – 18 | Blank | 4X | Blank |
| 19 – 25 | VLR(I) | F7.0 | Characteristic velocity required for mission on card I in fps |
| 26 – 31 | RPLM(I) | F6.0 | Return payload weight in lbs required by mission on card I* |
| 32 – 38 | WPR(I) | F7.0 | Payload weight in lbs of mission on card I |
| 39 – 40 | NTRIP(I) | I2 | Maximum number of launches allowed to carry WPR(I) into orbit, NTRIP = 0 indicates no reusable vehicle should be assigned to this mission |
| 41 – 80 | MISN(I, J), J=1, MYRS | 20I2 | Number of launches for mission on card I in year J+1900+IBY with WPR payload at each launch |

2nd card needed for each mission

| | | | |
|---|---|---|---|
| 3 – 12 | PLR(I) | F10.2 | Payload recurring cost for mission on card I |
| 13 – 22 | PLS(I) | F10.2 | Payload sustaining cost for mission on card I |

*Currently, any value ≥ 1.0 indicates that mission I is to be performed by only reusable vehicles.

| Card Columns | Variable Name | Format | Description and Comments |
|---|---|---|---|
| 23 – 32 | PLD(I) | F10.2 | Payload development cost for mission on card I |
| 33 – 57 | Blank | 25X | Blank |
| 58 – 67 | PLMD(I) | F10.2 | Maximum diameter of payload for mission on card I |
| 68 – 69 | NPLS(I) | I2 | Code for payload stabilization requirement for mission on card I<br>0 – No requirement<br>1 – Must be spin stabilized<br>2 – Must not be spin stabilized |
| 70 – 71 | MR(I) | I2 | Code for man rating requirement for mission on card I<br>0 – No requirement<br>1 – Must be man rated |
| 72 – 73 | LTR(I) | I2 | Code for launch site of mission on card I<br>1 – ETR<br>2 – WTR |
| 74 – 75 | NRR(I) | I2 | Number of restarts required for mission on card I |
| 76 – 77 | Blank | 2X | Blank |
| 78 – 80 | IVEHA(I) | I3 | A priori vehicle assignment for mission on card I<br>If no vehicle assigned = 0<br>Otherwise = KODEV of vehicle |

Last mission card must be followed by blank card.

Vehicle Cards (Input only if IV ≥ 0) J=1, NV ≤ 60

| | | | |
|---|---|---|---|
| 1 – 8 | VEH(I,J), I=1, 4 | 4I2 | KODE of stage in Ith position, starting with lowest stage, for vehicle on card J |
| 9 – 21 | B1(J) | E13.6 | Payload versus velocity curve constants for performance evaluation of vehicle on card J<br>PL = EXP(B1-B2*V-B3/(B4-V)) |
| 22 – 34 | B2(J) | E13.6 | |
| 35 – 47 | B3(J) | E13.6 | |
| 48 – 60 | B4(J) | E13.6 | |
| 79 – 80 | KODEV(J) | I2 | # of vehicle on card J |

| Card Columns | Variable Name | Format | Description and Comments |
|---|---|---|---|
| 2nd card needed for each vehicle | | | |
| 4 — 5 | NVS(J) | I2 | Code for stabilization of vehicle on card J<br>1 — Is spin stabilized<br>2 — Is not spin stabilized |
| 6 — 7 | MRV(J) | I2 | Code for man rating of vehicle on card J<br>0 — Is not man rated<br>1 — Is man rated |
| 8 — 9 | NRP(J) | I2 | Number of restarts possible for vehicle on card J |
| 10 — 12 | NPAD(1, J) | I3 | KODEP of pad complex at ETR from which vehicle J can be launched |
| 13 — 15 | NPAD(2, J) | I3 | KODEP of pad complex at WTR from which vehicle J can be launched |
| 16 — 18 | NYP(1, J) | I3 | 1st year J-th vehicle can be flown from ETR |
| 19 — 21 | NYP(2, J) | I3 | 1st year J-th vehicle can be flown from WTR |

Last vehicle card must be followed by blank card.

| Stage Performance Cards | | (Input only if NOPT = 2 on control card) I ≤ NSTG ≤ 40 | |
|---|---|---|---|
| 1 — 4 | KODE(I) | I4 | Reference number of stage on card I (used to check order of cards) |
| 5 — 9 | NST(I) | I5 | Classification of stage on card I |
| 10 — 19 | THRT(I) | F10.0 | Stage thrust |
| 20 — 29 | DIAM(I) | F10.0 | Stage diameter |
| 30 — 39 | TSL(I) | F10.0 | Stage sea-level thrust |
| 40 — 49 | LENT(I) | F10.0 | Interstage length required to clear engines |
| 50 — 59 | WTFU(I) | F10.0 | Stage fuel weight |
| 60 — 69 | WTIN(I) | F10.0 | Stage total inert weight |
| 70 — 79 | ISP(I) | F10.0 | Stage vacuum specific impulse |

Last card of total data deck must be blank card. Last Performance card must be followed by blank card.

Appendix B
SAMPLE CASE

## B.1 DESCRIPTION

The output from a sample case is presented in this Appendix. Data is synthesized and no significance should be attached to the values used. The listing includes a module map so that storage requirements are defined for each subroutine and common block. The program also prints out input data for easy reference and to provide a check on punched data.

Each section of output is described in detail.

(1) STAGE COST DATA includes stage title or identifying name, recurring cost of first unit and learning curve (LC) factor for hardware, ETR, and WTR recurring cost respectively. If any stage has jump – discontinuous form of recurring cost for any of the above three types, then the second line provides relevant information. Development and sustaining costs for each stage are listed along with years of availability referenced to the launch base year. Each stage may belong to at most four "shared cost groups," whose reference numbers are listed on the output. Each group number is referenced to the "Shared Cost Data" which follows this section. "Batch Fact" defines the number of years over which vehicles may be considered as produced in one period of time for learning curve purposes.

(2) SHARED COST DATA includes data on each shared cost group which was referenced in (1) above. These groups may be families such as the Titan family or they may be subsystems, such as a guidance system shared by several stages. Total development cost for any vehicle equals the sum of the development costs for each of its component stages plus any development costs for any shared groups associated with these stages plus any integration development costs required.

(3) INTEGRATION COST DATA are always between "families." If a specific stage-to-stage integration cost is desired, each stage must be put in a shared cost group by itself. Thus, many shared cost groups in (2) above will have no associated non-recurring costs. These groups will, however, be integrated with other shared cost groups which do have an integration non-recurring cost.

(4) PAD COST DATA lists the complex reference number, identifying name and location, i.e.g. TITE represents Titan ETR complex, and the next entry shows the maximum number of launches per year per pad at this complex. All possible combinations of pad-related costs are listed with their respective values for each pad. Provision for a third pad at each complex has been made although the program currently ignores all third pad costs.

(5) MISSION MODEL DATA includes mission internal reference number, identifying name, total $\Delta V$ required, payload required in lbs., priority value, launch site identification, (1 = ETR, 2 = WTR), and launch rate schedule by year.

(6) Input Data Totals includes total number of each input item along with other pertinent information from control card.

(7) QUANTITIES BRANCHED UPON lists every non-zero, non-recurring cost or "budget option" which the algorithm will consider in the optimization process along with its availability status. The reference number listed is used in the optional logic output described in (9).

(8) VEHICLE/MISSION CAPABILITY is a matrix of final vehicle-to-mission compatibility presenting the results of subroutines CAPABL and AVAIL. Each vehicle is listed by stage components and internal reference number. The vehicle/mission number on the top line represents the mission-year combination number (NM) while the mission number only is given on the following line at the top of the matrix. Separate entries are provided for each vehicle and launch site combination.

(9) BRANCH AND BOUND NODE VALUES presents optional information which enables the user to check the internal logic of the algorithm Each node is given a reference number which it keeps until its associated total bound exceeds the value of a known solution. (**** represents a very large number, denoting an unfeasible combination). The node number from which branching is taking place is provided in the second column. The last new node to be

generated at each branch is given the reference number of its parent node for continuity. The reference number of the cost item under consideration (see (7) on preceding page) is listed in the next column along with the appropriate sustaining year for that node. (0 represents no development or sustaining for that cost item) The recurring, non-recurring and total lower bounds are then provided so that each branch in the decision tree is represented. NEW GUESS alerts that a final solution has been found. The value printed represents the least cost solution found at that time. If some pad costs and small sustaining costs were ignored by the algorithm, then these extra costs are computed and added to the lower bound of the corresponding node. The values of these costs are printed out below the node information for the possible solution under consideration. When the optimum solution has been found – the least cost possible solution already investigated – this fact is designated below the associated node information

(10) Optimum Launch Vehicle Assignment – The optimum launch vehicle for each mission-year combination is printed out – as well as an array of mission information for easy reference. The "Number of Launches" represents the launch rate by year multiplied by the priority factor and the number of trips required by the associated vehicle to satisfy the mission payload requirements.

Sections (9) and (10) are repeated using recurring cost values modified by learning curve effects. The optimum solution at the end of this iteration is so designated. If no new case data is input, then the normal termination of the run is designated by END OF DATA – JOB COMPLETE.

The sample case included in this Appendix required 0.6 minutes on the 360/67 available at Ames Research Center, Moffett Field, California. Estimating run time is quite difficult for a new set of data since the number of solutions "close to" the optimum solution determines how large the decision tree will be and, as a consequence, how much computer time must be expended. As a general rule, the computer time increases linearly with the number of missions in the mission model and exponentially with the number of decision items determined by the stage, shared group, integration and pad cost input.

The flow diagrams in Appendix C define all non-normal exits from the algorithm in CHOOZ. Each non-normal termination of a case is denoted by a print-out of the qualifying reason. The program then reads in new case data, if available, and proceeds as normal.

## B.2 SAMPLE CASE PRINTOUT

A sample case printout follows:

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,OVLY,MAP
        VARIABLE OPTIONS USED - SIZE=(126976,245760)              DEFAULT OPTION(S) USED
IEW0000      INCLUDE DECKS(MOX02MN,MOX01PK,MOX02DC,MOX02AV,MOX02CP,MOX02SN)
IEW0000      INCLUDE DECKS(MOX02DP,MOX02MT,MOX02PR,MOX02LB,MOX02PD)
IEW0000      ENTRY MAIN
IEW0000        OVERLAY A
IEW0000      INCLUDE DECKS(MOX02CH)
IEW0000        OVERLAY A
IEW0000      INCLUDE DECKS(MOX02MA,MOX02MM,MOX02PF)
```

MODULE MAP

| CONTROL SECTION NAME | ORIGIN | LENGTH | SEG NO |
|---|---|---|---|
| $SEGTAB | 00 | 24 | 1 |
| MAIN | 28 | 20D4 | 1 |
| PACK | 2100 | E8 | 1 |
| DECISN | 21E8 | 1198 | 1 |
| AVAIL | 3380 | 7D0 | 1 |
| CAPABL | 3B50 | B98 | 1 |
| STGNUM | 46E8 | 1B74 | 1 |
| OUTPUT | 6260 | 5E4 | 1 |
| MATCH | 6848 | DB4 | 1 |
| PRINT | 7600 | D8A | 1 |
| LBOUND | 8390 | A4C | 1 |
| PADCST | 8DE0 | 1290 | 1 |
| IHCSLOG * | A070 | 18A | 1 |
| IHCSSCN * | A230 | 1ED | 1 |
| IHCSEXP * | A420 | 1B0 | 1 |
| IHCFRXPR* | A5D0 | 183 | 1 |
| IHCECOMH* | A758 | F31 | 1 |
| IHCCOMH2* | B690 | 545 | 1 |
| IHCFMAXI* | BBD8 | C9 | 1 |
| IHCSSQRT* | BCA8 | 149 | 1 |
| IHCFCVTH* | BDF8 | 1175 | 1 |
| IHCEFNTH* | CF70 | 512 | 1 |
| FIOCS* * | D488 | 160 | 1 |
| IHCEFIOS* | D5E8 | 111C | 1 |

| ENTRY NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|
| UNPACK | 2152 | ITEM | 2196 | | | | |
| ALOG10 | A070 | ALOG | A08C | | | | |
| COS | A230 | SIN | A24C | | | | |
| EXP | A420 | | | | | | |
| FRXPR* | A5D0 | | | | | | |
| IBCOM* | A758 | FDIOCS* | A814 | INTSWTCH | B676 | | |
| SEQDASD | B8F0 | | | | | | |
| MAXO | BBD8 | MINO | BBEE | AMAXO | BC04 | AMINO | BC1A |
| SQRT | BCA8 | | | | | | |
| ADCON* | BDF8 | FCVADUTP | BEA2 | FCVLOUTP | BF32 | FCVZOUTP | C082 |
| FCVIOUTP | C40E | FCVEOUTP | C910 | FCVCOUTP | C82A | INT6SWCH | CE13 |
| ARITH* | CF70 | ADJSWTCH | D2DC | | | | |
| SET899 | D514 | RES899 | D52E | | | | |

```
   NAME     ORIGIN   LENGTH  SEG. NO.      NAME    LOCATION      NAME    LOCATION      NAME    LOCATION      NAME    LOCATION

                                          FIOCSBEP    D5EE

  IHCERRN *    E708      5AC      1

                                          ERRMON      E708      IHCERRE     E720

  IHCUOPT *    ECB8      398      1
  IHCETRCH*    F050      28E      1

                                          IHCTRCH     F050      ERRTRA      F058

  IHCUATBL*    F2E0      638      1
  DSCOST       F918     1838      1
  STAGE       11150      6E4      1
  MISSON      11838      A90      1
  VEHCLE      122C8      480      1
  NONDIM      12778       1C      1
  INT         12798      3C0      1
  PADS        12B58     378C      1
  VEHREC      162E8     3CF4      1
  REUSBL      19FE0      190      1
  CAPMAT      1A170      FA4      1
  FAMILY      1B118      200      1
  INTREC      1B3E8      140      1
  STGREC      1B528      670      1
  MISREQ      1BB98      640      1
  VEHIN       1C1D8      1F8      1
  CHODZE      1C3D0       84      1
  SAVEOP      1C458      194      1
  SCRACH      1C5F0     53FC      1
  BATCH       219F0      780      1
  SAVEPT      22170      184      1
  SAVELB      222F8       18      1
  NUMBER      22310     2580      1
 $ENTAB       24890       24      1


  CHODZ        248B8     183E      2


  MATE         24BB8      B42      3
  MISMAT       25400      4C6      3
  PERF         258C8      5CE      3

ENTRY ADDRESS       28
TOTAL LENGTH     263F8
```

STAGE COST-DATA

| TITLE | RECURRING (ETR OR HTR) | LC | RECURRING (ETR ONLY) | LC | RECURRING (HTR ONLY) | LC | DEVELOPMENT | SUSTAINING | AVAILABLE FROM | TO | SHARED COST GROUPS | | | | BATCH FACT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SV3A | 6 89 | 0 950 | 6 23 | 0.950 | 3.27 | 0 950 | 0 0 | 3 37 | 1 | 20 | 1 | 0 | 0 | 0 | 5 |
| AG D | 1 87 | 0 900 | 1.40 | 0 900 | 2.50 | 0.900 | 0.0 | 2 60 | 1 | 20 | 2 | 7 | 0 | 0 | 6 |
| RECURRING COST TYPE 1 | FOR X LESS THAN OR = | 2 00, | TOTAL COST = | 2.30 | FOR X GREATER THAN | 2.00, | TOTAL COST = | 0 40 X + | 1.40 | | | | | | |
| SV3C | 7 09 | 0 850 | 6 43 | 0 850 | 3 47 | 0 850 | 0 0 | 0 0 | 1 | 20 | 1 | 0 | 0 | 0 | 4 |
| CENT | 11.85 | 0.900 | 4.62 | 0.900 | 4 62 | 0 900 | 25 30 | 16.00 | 1 | 20 | 4 | 0 | 0 | 0 | 5 |
| T3B | 4 91 | 1 000 | 1.44 | 1.000 | 1 44 | 1 000 | 0 0 | 0 0 | 1 | 20 | 3 | 8 | 0 | 0 | 3 |
| T3D | 8 14 | 0.950 | 5 01 | 0.950 | 3 45 | 0 950 | 0 0 | 0 0 | 1 | 20 | 3 | 6 | 0 | 0 | 3 |
| TRAN | 6 09 | 1 000 | 0 0 | 1 000 | 0 0 | 1 000 | 0 0 | 0 0 | 1 | 20 | 3 | 0 | 0 | 0 | 4 |
| AGLT | 3 50 | 1 000 | 0 0 | 1 000 | 0 0 | 1 000 | 17.50 | 1.20 | 3 | 20 | 2 | 9 | 0 | 0 | 5 |
| RECURRING COST TYPE 2 | FOR X LESS THAN OR = | 2.00, | TOTAL COST = | 2 30 | FOR X GREATER THAN | 2.00, | TOTAL COST = | 0 40 X + | 1 40 | | | | | | |
| SV3X | 6 66 | 0 850 | 5.76 | 0 850 | 2.58 | 0 850 | 13.00 | 0.0 | 3 | 20 | 1 | 10 | 0 | 0 | 4 |
| T3M | 22.00 | 0 950 | 5.01 | 0 950 | 3.45 | 0.950 | 25.00 | 0.0 | 3 | 20 | 3 | 11 | 0 | 0 | 3 |
| B2S | 0 87 | 1.000 | 0 10 | 1 000 | 0.10 | 1.000 | 0 25 | 0.10 | 1 | 20 | 5 | 0( | 0 | 0 | 5 |

SHARED COST DATA

| NO. | TITLE | DEVELOPMENT | SUSTAINING |
|---|---|---|---|
| 1 | ATLS | 0 0 | 4 66 |
| 3 | TITN | 0 0 | 6 00 |
| 5 | B2S | 0.0 | 0 0 |
| 7 | AG D | 0 0 | 0 0 |
| 9 | AGLT | 0 0 | 0.0 |
| 11 | T3M | 0.0 | 0 0 |
| 2 | ACEN | 0.0 | 2.50 |
| 4 | CENT | 0 0 | 0.0 |
| 6 | T3D | 0 0 | 0.0 |
| 8 | T3B | 0.0 | 0.0 |
| 10 | SV3X | 0 0 | 0 0 |

INTEGRATION COST DATA

| LOWER GROUP | UPPER GROUP | RECURRING | LC | DEVELOPMENT | SUSTAINING |
|---|---|---|---|---|---|
| TITN | AGEN | 0 0 | 1 000 | 14 00 | 0 0 |
| T3D | AG D | 0.0 | 1 000 | 2.50 | 0 0 |
| T3D | AGLT | 0 0 | 1.000 | 5.50 | 0 0 |
| TITN | CENT | 0 0 | 1 000 | 80.00 | 0 0 |
| T3B | CENT | 0 0 | 1.000 | 2 00 | 0 0 |
| SV3X | CENT | 0.0 | 1 000 | 4 00 | 0 0 |
| AGEN | B2S | 0 0 | 1.000 | 2 80 | 0 0 |
| TITN | B2S | 0.0 | 1 000 | 2.80 | 0 0 |
| CENT | B2S | 0 0 | 1 000 | 2 80 | 0 0 |
| T3M | AGLT | 0 0 | 1 000 | 5 00 | 0 0 |

PAD COST DATA

| NO | COMPLEX | LAUNCHES/YR | | PAD 1 DEV | PAD 1 SUST | PAD 2 DEV | PAD 2 SUST | PAD 3 DEV | PAD 3 SUST |
|---|---|---|---|---|---|---|---|---|---|
| 1 | TITE | 2 00 | | | | | | | |
| | | | T3B STAGE | 0.0 | 0.0 | 0.0 | 0 0 | 0 0 | 0 0 |
| | | | AG D STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | B2S STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | CENT STAGE | 20.00 | 10.00 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | T3D STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0.0 |
| | | | TRAN STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | AGLT STAGE | 0 0 | 0 0 | 0 10 | 0 10 | 0 0 | 0 0 |
| | | | T3M STAGE | 0.0 | 0 0 | 0 0 | 0.0 | 0 0 | 0 0 |
| | | | TITN SHARED | 0 0 | 0 0 | 0 10 | 0 10 | 0 0 | 0 0 |
| | | | T3B SHARED | 0 0 | 0 0 | 0 0 | 0 0 | 0.0 | 0 0 |
| | | | AGEN SHARED | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | AG D SHARED | 0 0 | 0 0 | 0 0 | 0 0 | 0.0 | 0 0 |
| | | | B2S SHARED | 0 0 | 0 0 | 0.0 | 0 0 | 0 0 | 0 0 |
| | | | INTEGRATION OF TITN AND AGEN | | 0 0 | | 0 10 | | 0 0 |
| | | | INTEGRATION OF TITN AND AG D | | 0.0 | | 0 0 | | 0 0 |
| | | | INTEGRATION OF T3B AND AGEN | | 0 0 | | 0 0 | | 0 0 |
| | | | INTEGRATION OF T3B AND AG D | | 0 0 | | 0 0 | | 0 0 |
| | | | INTEGRATION OF AGEN AND B2S | | 0 0 | | 0 0 | | 0 0 |
| 2 | TITN | 2 00 | | | | | | | |
| | | | T3B STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | AG D STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | B2S STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0.0 | 0 0 |
| | | | CENT STAGE | 20 00 | 10 00 | 20.00 | 10 00 | 0 0 | 0 0 |
| | | | T3D STAGE | 0 0 | 0 0 | 1 00 | 1 00 | 0.0 | 0 0 |
| | | | TRAN STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | AGLT STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | T3M STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | TITN SHARED | 0 0 | 0 0 | 0 0 | 0 0 | 0.0 | 0 0 |
| | | | T3B SHARED | 0 0 | 0 0 | 1.00 | 1.00 | 0 0 | 0 0 |
| | | | AGEN SHARED | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | AG D SHARED | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | B2S SHARED | 0 0 | 0 0 | 1.00 | 1 00 | 0.0 | 0 0 |
| | | | INTEGRATION OF TITN AND AGEN | | 0 0 | | 0 0 | | 0 0 |
| | | | INTEGRATION OF TITN AND AG D | | 0 0 | | 0 0 | | 0 0 |
| | | | INTEGRATION OF T3B AND AGEN | | 0 0 | | 1 00 | | 0 0 |
| | | | INTEGRATION OF T3B AND AG D | | 0 0 | | 0 0 | | 0 0 |
| | | | INTEGRATION OF AGEN AND B2S | | 0 0 | | 1 00 | | 0 0 |
| 3 | ATLE | 2.00 | | | | | | | |
| | | | SV3A STAGE | 0.0 | 0 0 | 0 0 | 0 0 | 0.0 | 0 0 |
| | | | AG D STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | SV3C STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0.0 | 0 0 |
| | | | CENT STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | SV3X STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0.0 | 0 0 |
| | | | ATLS SHARED | 20 00 | 5 00 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | AGEN SHARED | 0.0 | 0 0 | 0 0 | 0.0 | 0 0 | 0 0 |
| | | | AG D SHARED | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | CENT SHARED | 0 0 | 0 0 | 0 0 | 0.0 | 0.0 | 0 0 |
| | | | SV3X SHARED | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | INTEGRATION OF ATLS AND AGEN | | 0 0 | | 0 0 | | 0 0 |
| | | | INTEGRATION OF ATLS AND AG D | | 0 0 | | 0 0 | | 0 0 |
| | | | INTEGRATION OF ATLS AND CENT | | 2 00 | | 0 0 | | 0 0 |
| | | | INTEGRATION OF SV3X AND AGEN | | 0 0 | | 0 0 | | 0 0 |
| | | | INTEGRATION OF SV3X AND CENT | | 0 0 | | 0 0 | | 0 0 |
| 4 | ATLN | 2.00 | | | | | | | |
| | | | SV3A STAGE | 0 0 | 0 0 | 0.0 | 0 0 | 0 0 | 0 0 |
| | | | AG D STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | SV3C STAGE | 0.0 | 0 0 | 0 0 | 0 0 | 0.0 | 0 0 |
| | | | CENT STAGE | 0.0 | 0 0 | 0.0 | 0 0 | 0 0 | 0 0 |
| | | | SV3X STAGE | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| | | | ATLS SHARED | 20.00 | 5 00 | 0.0 | 0 0 | 0.0 | 0 0 |
| | | | AGEN SHARED | 0 0 | 0.0 | 0.0 | 0 0 | 0 0 | 0 0 |
| | | | AG D SHARED | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| CENT SHARED | 0 0 | 0.0 | 0 0 | 0 0 | 0 0 | 0 0 |
| SV3X SHARED | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| INTEGRATION OF ATLS AND AGEN | | 0 0 | | 0 0 | | 0 0 |
| INTEGRATION OF ATLS AND AG D | | 0 0 | | 0 0 | | 0 0 |
| INTEGRATION OF ATLS AND CENT | | 2.00 | | 0 0 | | 0 0 |
| INTEGRATION OF SV3X AND AGEN | | 0.0 | | 0 0 | | 0 0 |
| INTEGRATION OF SV3X AND CENT | | 0 0 | | 0 0 | | 0 0 |

MISSION MODEL

| | MISSION | VELOCITY | PAYLOAD | PRIORITY | | LAUNCH SCHEDULE 71 | 72 | 73 | 74 | 75 | 76 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | MARS71 | 41500 | 1100 | 1 00 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| 2 | PIONER | 49000. | 460. | 1 00 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 3 | MARS73 | 40000 | 6000 | 1 00 | 1 | 0 | 0 | 2 | 0 | 0 | 0 |
| 4 | MERCRY | 38900 | 1000. | 0 50 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 5 | GRDTRT | 49500. | 800. | 0.50 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 6 | MARS75 | 39400 | 3000. | 0 50 | 1 | 0 | 0 | 0 | 0 | 2 | 0 |
| 7 | COMET | 37200. | 2000 | 0.50 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 8 | ASTRA | 26300 | 7000 | 0.50 | 2 | 0 | 0 | 0 | 0 | 0 | 1 |
| 9 | RELTIV | 14200 | 2000 | 0 50 | 2 | 0 | 0 | 0 | 1 | 0 | 0 |
| 10 | PIONDE | 51400 | 1000 | 0.50 | 2 | 0 | 0 | 0 | 0 | 1 | 0 |
| 11 | ATS | 33600 | 4000 | 1.00 | 2 | 0 | 1 | 1 | 0 | 0 | 0 |
| 12 | DRELAY | 14200. | 2000. | 0 50 | 2 | 0 | 0 | 0 | 0 | 0 | 1 |
| 13 | USAMSC | 33600 | 4100. | 0 50 | 2 | 0 | 0 | 0 | 0 | 1 | 0 |
| 14 | USAMSM | 30000. | 6800. | 0 50 | 2 | 0 | 0 | 0 | 0 | 0 | 2 |

| | |
|---|---|
| NUMBER OF STAGES | 11 |
| NUMBER OF VEHICLES | 20 |
| NUMBER OF FAMILIES | 11 |
| NUMBER OF INTEGRATION COSTS | 10 |
| NUMBER OF PAD COMPLEXES | 4 |
| NUMBER OF MISSIONS | 14 |
| NUMBER OF YEARS | 6 |
| LAUNCH BASE YEAR | 71 |
| TOTAL COST ESTIMATE | 470 00 |
| OPTION NUMBER | 3 |

QUANTITIES BRANCHED UPON

| | DEVELOPMENT | SUSTAINING | | YEAR AVAIL | LAST YEAR |
|---|---|---|---|---|---|
| 1 | 0 0 | 3 37 | SV3A STAGE HARDWARE | 1 | 6 |
| 2 | 0 0 | 10 00 | SV3A STAGE ETR | 1 | 6 |
| 3 | 0 0 | 2.60 | AG D STAGE HARDWARE | 1 | 6 |
| 4 | 25 30 | 16 00 | CENT STAGE HARDWARE | 1 | 6 |
| 5 | 0 0 | 10 00 | CENT STAGE WTR | 1 | 6 |
| 6 | 17 50 | 1 20 | AGLT STAGE HARDWARE | 3 | 6 |
| 7 | 13 00 | 0 0 | SV3X STAGE HARDWARE | 3 | 6 |
| 8 | 25 00 | 0.0 | T3M STAGE HARDWARE | 3 | 6 |
| 9 | 0 25 | 0.10 | B2S STAGE HARDWARE | 1 | 6 |
| 10 | 0 0 | 4 66 | ATLS SHARED HARDWARE | 1 | 6 |
| 11 | 0 0 | 5 00 | ATLS SHARED ETR | 1 | 6 |
| 12 | 0 0 | 5 00 | ATLS SHARED WTR | 1 | 6 |
| 13 | 0 0 | 6 00 | TITN SHARED HARDWARE | 1 | 6 |
| 14 | 0.0 | 2.50 | AGEN SHARED HARDWARE | 1 | 6 |
| 15 | 14 00 | 0 0 | INTEGRATION OF TITN AND AGEN HARDWARE | 1 | 6 |
| 16 | 2.50 | 0.0 | INTEGRATION OF T3D AND AG D HARDWARE | 1 | 6 |
| 17 | 5.50 | 0 0 | INTEGRATION OF T3D AND AGLT HARDWARE | 3 | 6 |
| 18 | 80 00 | 0 0 | INTEGRATION OF TITN AND CENT HARDWARE | 1 | 6 |
| 19 | 0 0 | 5 00 | INTEGRATION OF TITN AND CENT ETR | 1 | 6 |
| 20 | 0 0 | 5 00 | INTEGRATION OF TITN AND CENT WTR | 1 | 6 |
| 21 | 2 00 | 0 0 | INTEGRATION OF T3B AND CENT HARDWARE | 1 | 6 |
| 22 | 4 00 | 0 0 | INTEGRATION OF SV3X AND CENT HARDWARE | 3 | 6 |
| 23 | 2 80 | 0 0 | INTEGRATION OF AGEN AND B2S HARDWARE | 1 | 6 |
| 24 | 2 80 | 0 0 | INTEGRATION OF TITN AND B2S HARDWARE | 1 | 6 |
| 25 | 2 80 | 0 0 | INTEGRATION OF CENT AND B2S HARDWARE | 1 | 6 |
| 26 | 5.00 | 0 0 | INTEGRATION OF T3M AND AGLT HARDWARE | 3 | 6 |
| 27 | 20.00 | 10.00 | CENT STAGE AT PAD TITE | 1 | 6 |
| 28 | 20 00 | 10 00 | CENT STAGE AT PAD TITW | 1 | 6 |
| 29 | 20 00 | 5.00 | ATLS SHARED AT PAD ATLE | 1 | 6 |
| 30 | 0 0 | 2.00 | INTEGRATION OF ATLS AND CENT AT PAD ATLE | 1 | 6 |
| 31 | 20.00 | 5 00 | ATLS SHARED AT PAD ATLW | 1 | 6 |
| 32 | 0 0 | 2.00 | INTEGRATION OF ATLS AND CENT AT PAD ATLW | 1 | 6 |

## VEHICLE/MISSION CAPABILITY
(1 = POSSIBLE, 0 = IMPOSSIBLE)

| VEHICLE / MISSION | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MISSION NUMBER | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | | | | | | | | | | | | | | | | | | | | | |
| 1 T38 AG D | E | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 2 T38 AG D B2S | E | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 3 T38 CENT | E | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 4 T38 CENT B2S | E | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 5 T3D TRAN | E | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 6 T3D TRAN B2S | E | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 7 T3D AG D | E | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 8 T3D AG D B2S | E | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 9 T3D AGLT | E | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 10 T3D AGLT B2S | E | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 11 T3D CENT | E | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 12 T3D CENT B2S | E | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 13 SV3A AG D | E | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 14 SV3A AG D B2S | E | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 15 SV3C CENT | E | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 16 SV3C CENT B2S | E | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 17 SV3X CENT | E | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 18 SV3X CENT B2S | E | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 19 T3M AGLT | E | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 20 T3M AGLT B2S | E | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | |
| 1 T38 AG D | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | |
| 2 T38 AG D B2S | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | |
| 3 T38 CENT | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 4 T38 CENT B2S | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 5 T3D TRAN | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 6 T3D TRAN B2S | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 7 T3D AG D | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 8 T3D AG D B2S | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 9 T3D AGLT | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 10 T3D AGLT B2S | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 11 T3D CENT | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 12 T3D CENT B2S | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 13 SV3A AG D | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | |
| 14 SV3A AG D B2S | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | |
| 15 SV3C CENT | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | |
| 16 SV3C CENT B2S | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | |
| 17 SV3X CENT | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 18 SV3X CENT B2S | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 19 T3M AGLT | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 20 T3M AGLT B2S | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | |

BRANCH AND BOUND NODE VALUES

| NODE NO | BRANCHED FROM | COST NO | YEARS SUSTAIN | RECURRING BOUND | NON-RECURRING BOUND | TOTAL BOUND |
|---|---|---|---|---|---|---|
| 2 | 1 | 13 | 0 | ********** | 0 0 | ********** |
| 3 | 1 | 13 | 1 | ********** | 6 00 | ********** |
| 4 | 1 | 13 | 2 | ********** | 12 00 | ********** |
| 5 | 1 | 13 | 3 | ********** | 18 00 | ********** |
| 6 | 1 | 13 | 4 | ********** | 24 00 | ********** |
| 7 | 1 | 13 | 5 | 239 93 | 41 67 | 281.60 |
| 1 | 1 | 13 | 6 | 216 86 | 47 67 | 264 53 |
| 6 | 1 | 15 | 0 | 272 33 | 216 85 | 489 18 |
| 1 | 1 | 15 | 6 | 216 86 | 61 67 | 278.53 |
| 6 | 1 | 18 | 0 | 228.56 | 107 90 | 336 46 |
| 1 | 1 | 18 | 6 | 216 86 | 141 67 | 358 53 |
| 5 | 7 | 7 | 0 | ********** | 30 00 | ********** |
| 7 | 7 | 7 | 6 | 239 93 | 54 67 | 294 60 |
| 5 | 7 | 22 | 0 | ********** | 43 00 | ********** |
| 7 | 7 | 22 | 6 | 239 93 | 58 67 | 298 60 |
| 5 | 7 | 10 | 0 | ********** | 47.00 | ********** |
| 4 | 7 | 10 | 1 | ********** | 51 66 | ********** |
| 3 | 7 | 10 | 2 | ********** | 56 32 | ********** |
| 2 | 7 | 10 | 3 | ********** | 60 98 | ********** |
| 8 | 7 | 10 | 4 | ********** | 65 64 | ********** |
| 9 | 7 | 10 | 5 | ********** | 70 30 | ********** |
| 7 | 7 | 10 | 6 | 239 93 | 86 63 | 326 56 |
| 9 | 7 | 12 | 0 | ********** | 74 96 | ********** |
| 8 | 7 | 12 | 1 | ********** | 79 96 | ********** |
| 5 | 7 | 12 | 2 | ********** | 84 96 | ********** |
| 4 | 7 | 12 | 3 | ********** | 89 96 | ********** |
| 3 | 7 | 12 | 4 | ********** | 94 96 | ********** |
| 2 | 7 | 12 | 5 | ********** | 99 96 | ********** |
| 7 | 7 | 12 | 6 | 239 93 | 116 63 | 356 56 |
| 9 | 6 | 6 | 0 | ********** | 50.00 | ********** |
| 6 | 6 | 6 | 6 | 228 56 | 107 90 | 336 46 |
| 9 | 6 | 8 | 0 | ********** | 67 50 | ********** |
| 6 | 6 | 8 | 6 | 228 56 | 107 90 | 336.46 |
| 9 | 6 | 16 | 0 | ********** | 92 50 | ********** |
| 6 | 6 | 16 | 6 | 228 56 | 107 90 | 336 46 |
| 9 | 6 | 23 | 0 | ********** | 95 00 | ********** |
| 6 | 6 | 23 | 6 | 228.56 | 107 90 | 336 46 |
| 9 | 6 | 26 | 0 | ********** | 97 80 | ********** |
| 6 | 6 | 26 | 6 | 228 56 | 107 90 | 336 46 |
| 9 | 6 | 14 | 0 | ********** | 102 80 | ********** |
| 8 | 6 | 14 | 1 | ********** | 105.30 | ********** |
| 5 | 6 | 14 | 2 | ********** | 107 80 | ********** |
| 4 | 6 | 14 | 3 | ********** | 110 30 | ********** |
| 3 | 6 | 14 | 4 | ********** | 112 80 | ********** |
| 2 | 6 | 14 | 5 | 243 39 | 117.90 | 361 29 |
| 6 | 6 | 14 | 6 | 228 56 | 120 40 | 348 96 |
| 9 | 6 | 3 | 0 | ********** | 117 80 | ********** |
| 8 | 6 | 3 | 1 | ********** | 120 40 | ********** |
| 5 | 6 | 3 | 2 | 252 41 | 128 50 | 380 91 |
| 4 | 6 | 3 | 3 | 250 44 | 131 10 | 301 54 |
| 3 | 6 | 3 | 4 | 245 14 | 130 62 | 375 77 |
| 10 | 6 | 3 | 5 | 237 92 | 132 74 | 370 66 |
| 6 | 6 | 3 | 6 | 228 56 | 133 40 | 361 96 |
| 9 | 7 | 31 | 0 | ********** | 104 96 | ********** |
| 8 | 7 | 31 | 1 | ********** | 129.96 | ********** |
| 11 | 7 | 31 | 2 | ********** | 134 96 | ********** |
| 12 | 7 | 31 | 3 | ********** | 139.96 | ********** |
| 13 | 7 | 31 | 4 | ********** | 144.96 | ********** |
| 14 | 7 | 31 | 5 | ********** | 149 96 | ********** |
| 7 | 7 | 31 | 6 | 239 93 | 166 63 | 406 56 |
| 14 | 1 | 4 | 0 | 228 56 | 187 90 | 416 46 |
| 13 | 1 | 4 | 1 | 228.56 | 229 20 | 457 76 |
| 12 | 1 | 4 | 2 | 228.56 | 245 20 | 473 76 |
| 11 | 1 | 4 | 3 | 218 24 | 263.14 | 481 38 |
| 9 | 1 | 4 | 4 | 218 24 | 279 14 | 497 38 |
| 8 | 1 | 4 | 5 | 216 86 | 246.97 | 463 83 |
| 1 | 1 | 4 | 6 | 216 86 | 262.97 | 479 83 |
| 12 | 2 | 3 | 0 | ********** | 115 30 | ********** |
| 11 | 2 | 3 | 1 | ********** | 117.90 | ********** |
| 9 | 2 | 3 | 2 | 257 88 | 126 00 | 383.88 |
| 15 | 2 | 3 | 3 | 255 91 | 128.60 | 384.51 |
| 16 | 2 | 3 | 4 | 250 61 | 127 15 | 377.77 |
| 17 | 2 | 3 | 5 | 243 39 | 128.30 | 371.69 |
| 2 | 2 | 3 | 6 | 243 39 | 130 90 | 374 29 |

NEW GUESS = 361 96

********** POSSIBLE SOLUTION **********

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | | | | 228 56 | 133 40 | 361 96 |

EXTRA PAD COSTS = 0 50

EXTRA PAD & SMALL SUST COSTS = 4.85

NEW GUESS = 366 81

********** POSSIBLE SOLUTION **********

6 228.56 138 25 366.81

THIS ASSIGNMENT IS OPTIMUM AT A TOTAL COST OF 366.81

| MISSION TITLE | CHARACTERISTIC VELOCITY(FT/SEC) | PAYLOAD (LBS) | RETURN PAYLOAD | LAUNCH YEAR | NUMBER OF LAUNCHES | OPTIMUM LAUNCH VEHICLE | |
|---|---|---|---|---|---|---|---|
| MARS71 | 41500. | 1100 | 0 | 1971 | 2 00 | T3D AG D | E |
| PIONER | 49000 | 460 | 0. | 1972 | 1 00 | T3D AG D B2S | E |
| | | | | 1973 | 1 00 | T3D AG D B2S | E |
| MARS73 | 40000. | 6000 | 0. | 1973 | 2.00 | T3M AGLT | E |
| MERCRY | 38900. | 1000 | 0 | 1975 | 0 50 | T3B AG D B2S | E |
| GRDTRT | 49500 | 800 | 0 | 1974 | 0 50 | T3D AG D B2S | E |
| MARS75 | 39400. | 3000 | 0. | 1975 | 1 00 | T3D AG D | E |
| COMET | 37200 | 2000. | 0 | 1976 | 1 00 | T3D AG D | E |
| ASTRA | 26300 | 7000 | 0 | 1976 | 0 50 | T3B AG D | W |
| RELTIV | 14200 | 2000 | 0 | 1974 | 0 50 | T3B AG D | W |
| PIONOE | 51400. | 1000. | 0 | 1975 | 0 50 | T3M AGLT B2S | W |
| ATS | 33600 | 4000. | 0 | 1972 | 1.00 | T3D AG D | W |
| | | | | 1973 | 1 00 | T3D AG D | W |
| DRELAY | 14200 | 2000. | 0. | 1976 | 0 50 | T3B AG D | W |
| USAMSC | 33600. | 4100. | 0 | 1975 | 0.50 | T3D AG D | W |
| USAMSM | 30000. | 6800. | 0 | 1976 | 1 00 | T3D AG D | W |

BRANCH AND BOUND NODE VALUES

| NODE NO. | BRANCHED FROM | COST NO. | YEARS SUSTAIN | RECURRING BOUND | NON-RECURRING BOUND | TOTAL BOUND |
|---|---|---|---|---|---|---|
| 2 | 1 | 13 | 0 | ********** | 0 0 | ********** |
| 3 | 1 | 13 | 1 | ********** | 6 00 | ********** |
| 4 | 1 | 13 | 2 | ********** | 12 00 | ********** |
| 5 | 1 | 13 | 3 | ********** | 18 00 | ********** |
| 6 | 1 | 13 | 4 | ********** | 24 00 | ********** |
| 7 | 1 | 13 | 5 | 243 28 | 43 12 | 286 41 |
| 1 | 1 | 13 | 6 | 220 90 | 49 12 | 270 03 |
| 6 | 1 | 15 | 0 | 275 69 | 217.54 | 493 22 |
| 1 | 1 | 15 | 6 | 220 90 | 63.12 | 284 03 |
| 6 | 1 | 18 | 0 | 236 33 | 107.90 | 344 23 |
| 1 | 1 | 18 | 6 | 220 90 | 143 12 | 364 03 |
| 5 | 7 | 7 | 0 | ********** | 30 00 | ********** |
| 7 | 7 | 7 | 6 | 243 28 | 56 12 | 299 41 |
| 5 | 7 | 22 | 0 | ********** | 43 00 | ********** |
| 7 | 7 | 22 | 6 | 243.28 | 60 12 | 303.41 |
| 5 | 7 | 10 | 0 | ********** | 47 00 | ********** |
| 4 | 7 | 10 | 1 | ********** | 51 66 | ********** |
| 3 | 7 | 10 | 2 | ********** | 56 32 | ********** |
| 2 | 7 | 10 | 3 | ********** | 60 98 | ********** |
| 8 | 7 | 10 | 4 | ********** | 65 64 | ********** |
| 9 | 7 | 10 | 5 | ********** | 70 30 | ********** |
| 7 | 7 | 10 | 6 | 243 28 | 88 08 | 331 37 |
| 9 | 7 | 12 | 0 | ********** | 74 96 | ********** |
| 8 | 7 | 12 | 1 | ********** | 79 96 | ********** |
| 5 | 7 | 12 | 2 | ********** | 84 96 | ********** |
| 4 | 7 | 12 | 3 | ********** | 89 96 | ********** |
| 3 | 7 | 12 | 4 | ********** | 94 96 | ********** |
| 2 | 7 | 12 | 5 | ********** | 99.96 | ********** |
| 7 | 7 | 12 | 6 | 243 28 | 118 08 | 361 37 |
| 9 | 6 | 6 | 0 | ********** | 50.00 | ********** |
| 6 | 6 | 6 | 6 | 236 33 | 107.90 | 344 23 |
| 9 | 6 | 8 | 0 | ********** | 67.50 | ********** |
| 6 | 6 | 8 | 6 | 236 33 | 107 90 | 344 23 |
| 9 | 6 | 16 | 0 | ********** | 92 50 | ********** |
| 6 | 6 | 16 | 6 | 236 33 | 107.90 | 344 23 |
| 9 | 6 | 23 | 0 | ********** | 95.00 | ********** |
| 6 | 6 | 23 | 6 | 236 33 | 107 90 | 344.23 |
| 9 | 6 | 26 | 0 | ********** | 97 80 | ********** |
| 6 | 6 | 26 | 6 | 236 33 | 107 90 | 344 23 |
| 9 | 6 | 14 | 0 | ********** | 102 80 | ********** |
| 8 | 6 | 14 | 1 | ********** | 105 30 | ********** |
| 5 | 6 | 14 | 2 | ********** | 107 80 | ********** |
| 4 | 6 | 14 | 3 | ********** | 110 30 | ********** |
| 3 | 6 | 14 | 4 | ********** | 112 80 | ********** |
| 2 | 6 | 14 | 5 | 251 60 | 117 90 | 369 50 |
| 6 | 6 | 14 | 6 | 236.33 | 120 40 | 356 73 |
| 9 | 6 | 3 | 0 | ********** | 117 80 | ********** |
| 8 | 6 | 3 | 1 | ********** | 120 40 | ********** |
| 5 | 6 | 3 | 2 | 261 23 | 128 50 | 389 73 |
| 4 | 6 | 3 | 3 | 258 98 | 131 10 | 390 08 |
| 3 | 6 | 3 | 4 | 253 55 | 130 62 | 384 17 |
| 10 | 6 | 3 | 5 | 246 13 | 132 74 | 378 87 |
| 6 | 6 | 3 | 6 | 236 33 | 133 40 | 369 73 |
| 9 | 7 | 31 | 0 | ********** | 104 96 | ********** |
| 8 | 7 | 31 | 1 | ********** | 129 96 | ********** |
| 11 | 7 | 31 | 2 | ********** | 134 96 | ********** |
| 12 | 7 | 31 | 3 | ********** | 139 96 | ********** |
| 13 | 7 | 31 | 4 | ********** | 144.96 | ********** |
| 14 | 7 | 31 | 5 | ********** | 149 96 | ********** |
| 7 | 7 | 31 | 6 | 243.28 | 168.08 | 411 37 |
| 14 | 1 | 4 | 0 | 236 33 | 187 90 | 424.23 |
| 13 | 1 | 4 | 1 | 236 33 | 229 20 | 465 53 |
| 12 | 1 | 4 | 2 | 236 33 | 245.20 | 481.53 |
| 11 | 1 | 4 | 3 | 224 16 | 263 14 | 487.30 |
| 9 | 1 | 4 | 4 | 224 16 | 279 14 | 503 30 |
| 8 | 1 | 4 | 5 | 220 90 | 248 42 | 469 33 |
| 1 | 1 | 4 | 6 | 220.90 | 264.42 | 485 33 |
| 12 | 2 | 3 | 0 | ********** | 115 30 | ********** |
| 11 | 2 | 3 | 1 | ********** | 117.90 | ********** |
| 9 | 2 | 3 | 2 | 266 70 | 126 00 | 392 70 |
| 15 | 2 | 3 | 3 | 264 45 | 128.60 | 393 05 |
| 16 | 2 | 3 | 4 | 259.02 | 127.15 | 386.17 |
| 17 | 2 | 3 | 5 | 251.60 | 128 30 | 379 90 |
| 2 | 2 | 3 | 6 | 251.60 | 130.90 | 382.50 |

NEW GUESS = 369.73

********** POSSIBLE SOLUTION **********

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | | | | 236 33 | 133 40 | 369.73 |

EXTRA PAD COSTS = 0.50

EXTRA PAD & SHALL SUST COSTS = 4.85
NEW GUESS = 374.58

********** POSSIBLE SOLUTION **********

6 236 33 138.25 374.58

THIS ASSIGNMENT IS OPTIMUM AT A TOTAL COST OF 374 58

| MISSION TITLE | CHARACTERISTIC VELOCITY(FT/SEC) | PAYLOAD (LBS) | RETURN PAYLOAD | LAUNCH YEAR | NUMBER OF LAUNCHES | OPTIMUM LAUNCH VEHICLE | |
|---|---|---|---|---|---|---|---|
| MARS71 | 41500 | 1100 | 0 | 1971 | 2.00 | T3D AG D | E |
| PIONER | 49000. | 460. | 0. | 1972 | 1 00 | T3D AG D B2S | E |
| | | | | 1973 | 1 00 | T3D AG D B2S | E |
| MARS73 | 40000 | 6000 | 0. | 1973 | 2 00 | T3M AGLT | E |
| MERCRY | 38900. | 1000 | 0. | 1975 | 0 50 | T3B AG D B2S | E |
| GRDTRT | 49500 | 800 | 0 | 1974 | 0 50 | T3D AG D B2S | E |
| MARS75 | 39400 | 3000. | 0 | 1975 | 1 00 | T3D AG D | E |
| COMET | 37200. | 2000. | 0. | 1976 | 1.00 | T3D AG D | E |
| ASTRA | 26300 | 7000. | 0 | 1976 | 0 50 | T3B AG D | W |
| RELTIV | 14200. | 2000. | 0. | 1974 | 0 50 | T3B AG D | W |
| PIONDE | 51400. | 1000 | 0. | 1975 | 0 50 | T3M AGLT B2S | W |
| ATS | 33600 | 4000 | 0. | 1972 | 1 00 | T3D AG D | W |
| | | | | 1973 | 1.00 | T3D AG D | W |
| DRELAY | 14200. | 2000 | 0 | 1976 | 0 50 | T3B AG D | W |
| USAMSC | 33600. | 4100 | 0 | 1975 | 0 50 | T3D AG D | W |
| USAMSM | 30000 | 6800. | 0 | 1976 | 1.00 | T3D AG D | W |

THE OPTIMUM SOLUTION HAS BEEN DETERMINED

END OF DATA -*JOB COMPLETE

# Appendix C
# FLOW CHARTS

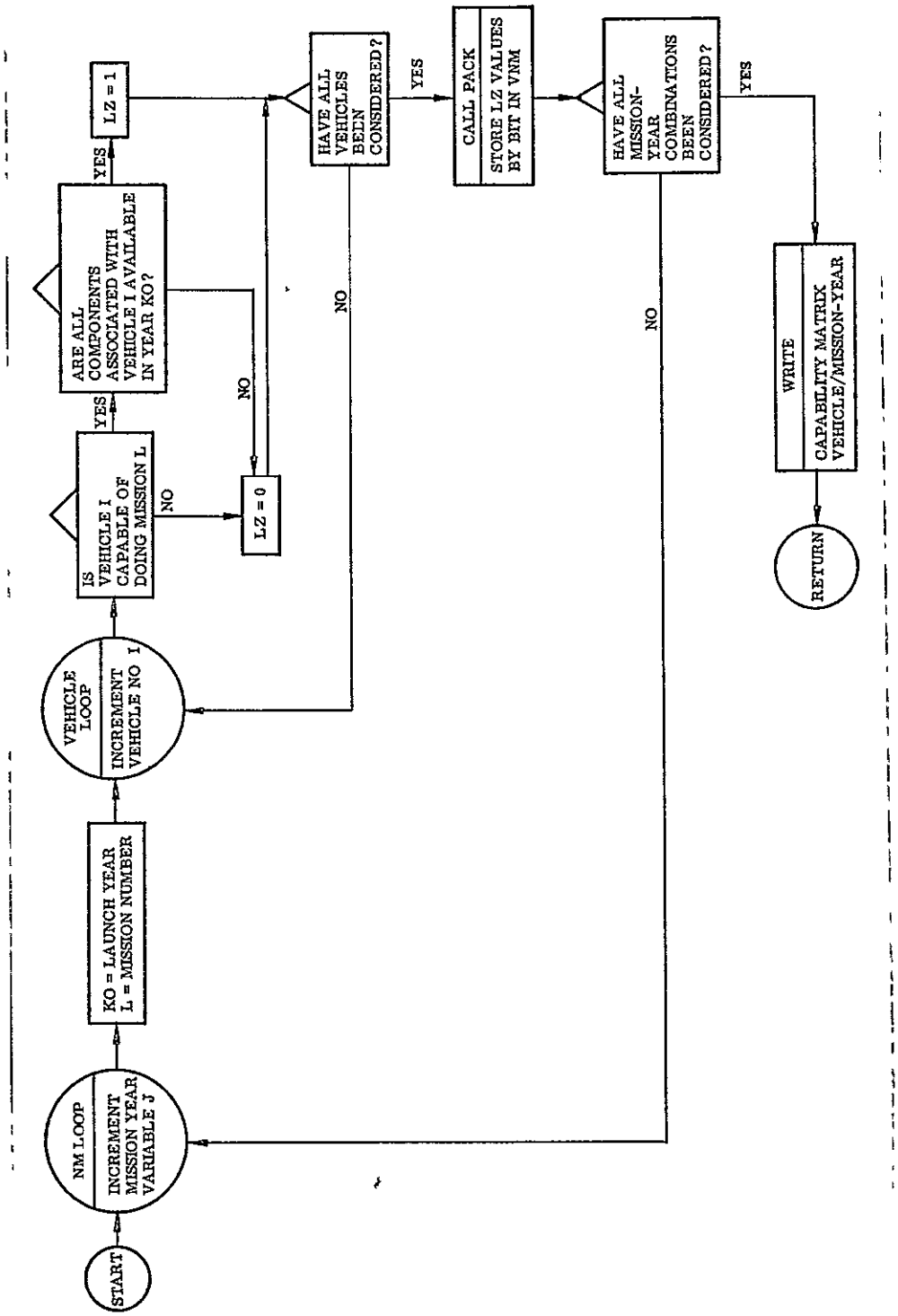
## C.1 DESCRIPTION

Detailed flow charts are included in this appendix for each of the major subroutines and the main program MAIN. They appear in alphabetical order by subroutine name.

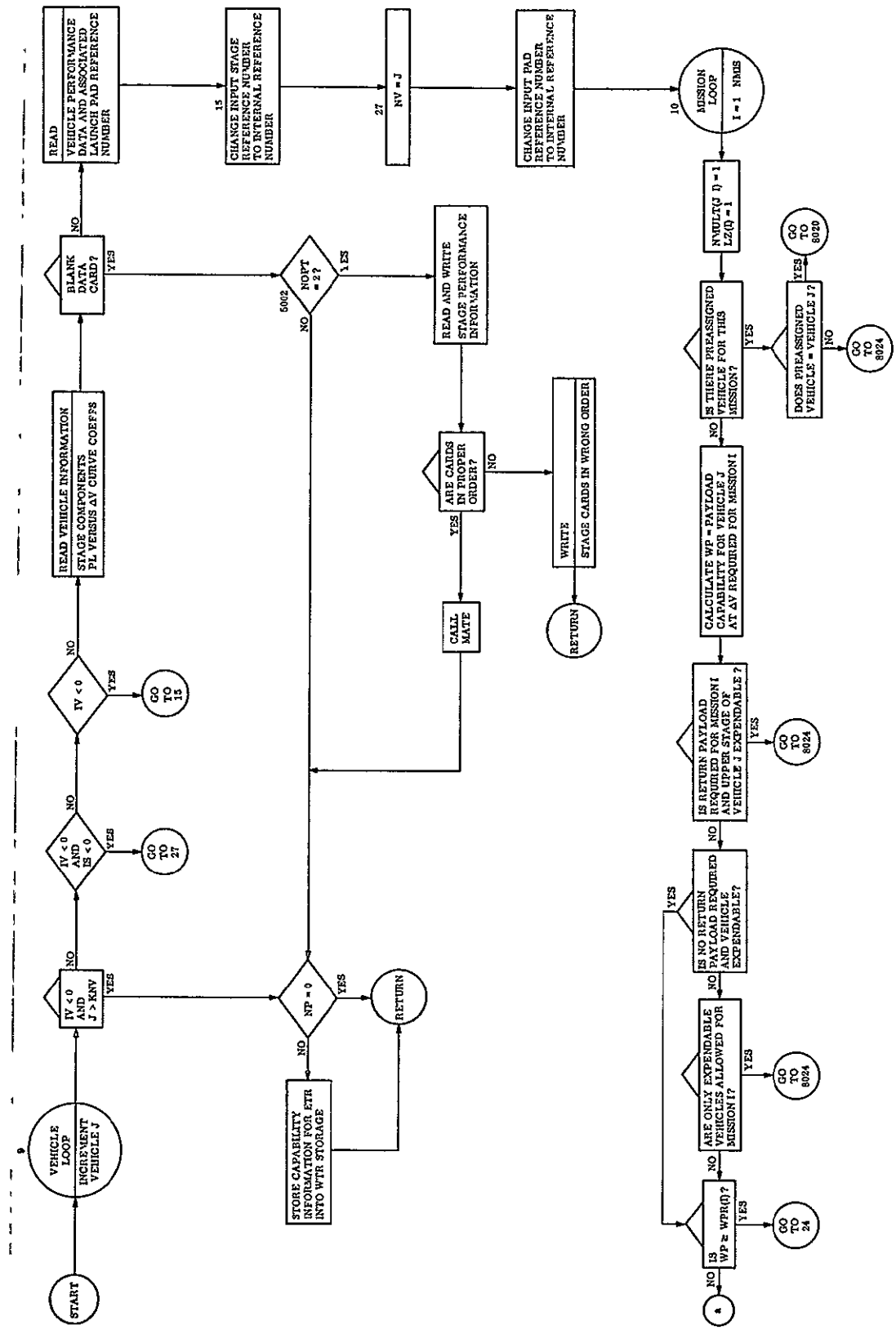
## C.2 MAJOR SUBROUTINE CHARTS

The subroutine charts follow

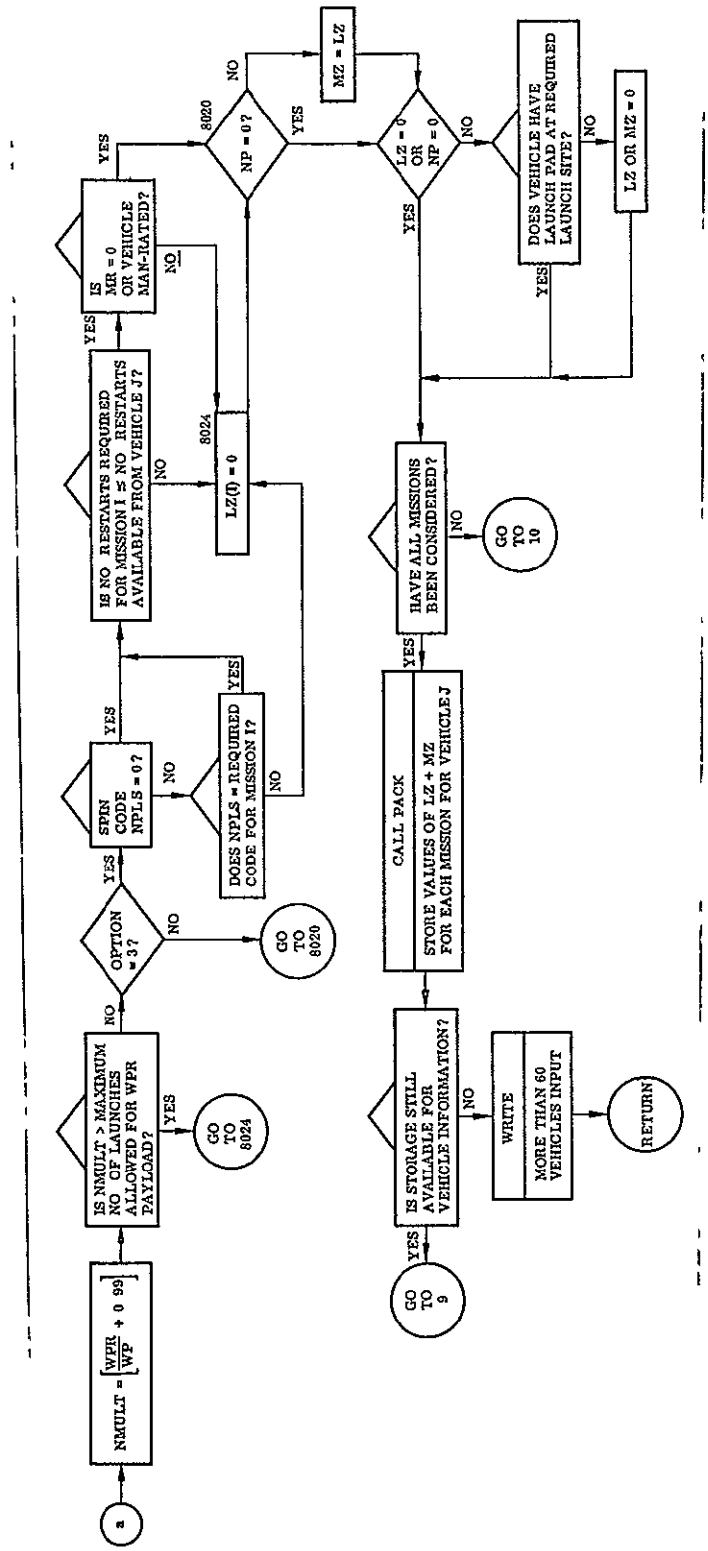
SUBROUTINE AVAIL



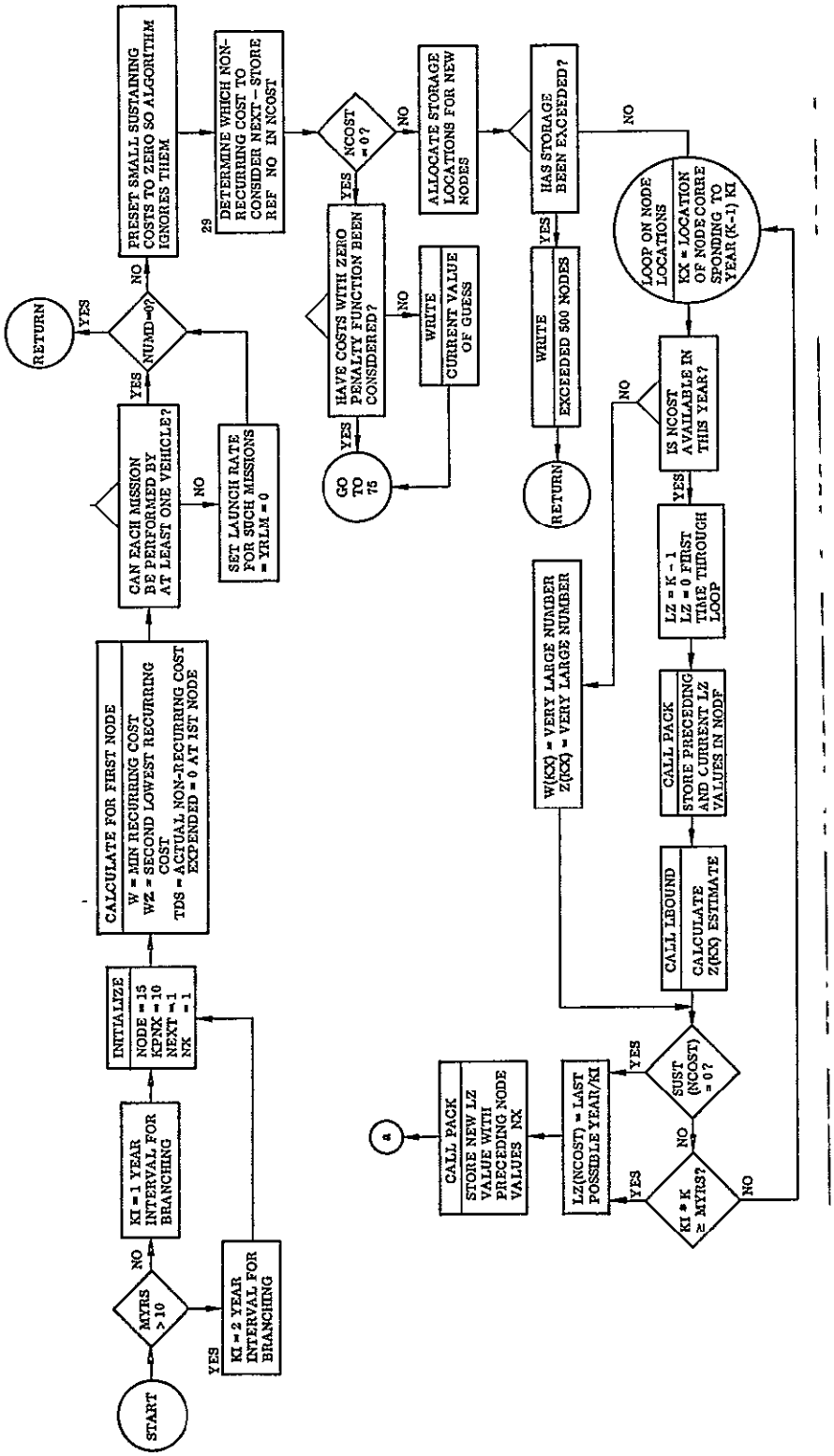
# SUBROUTINE CAPABL



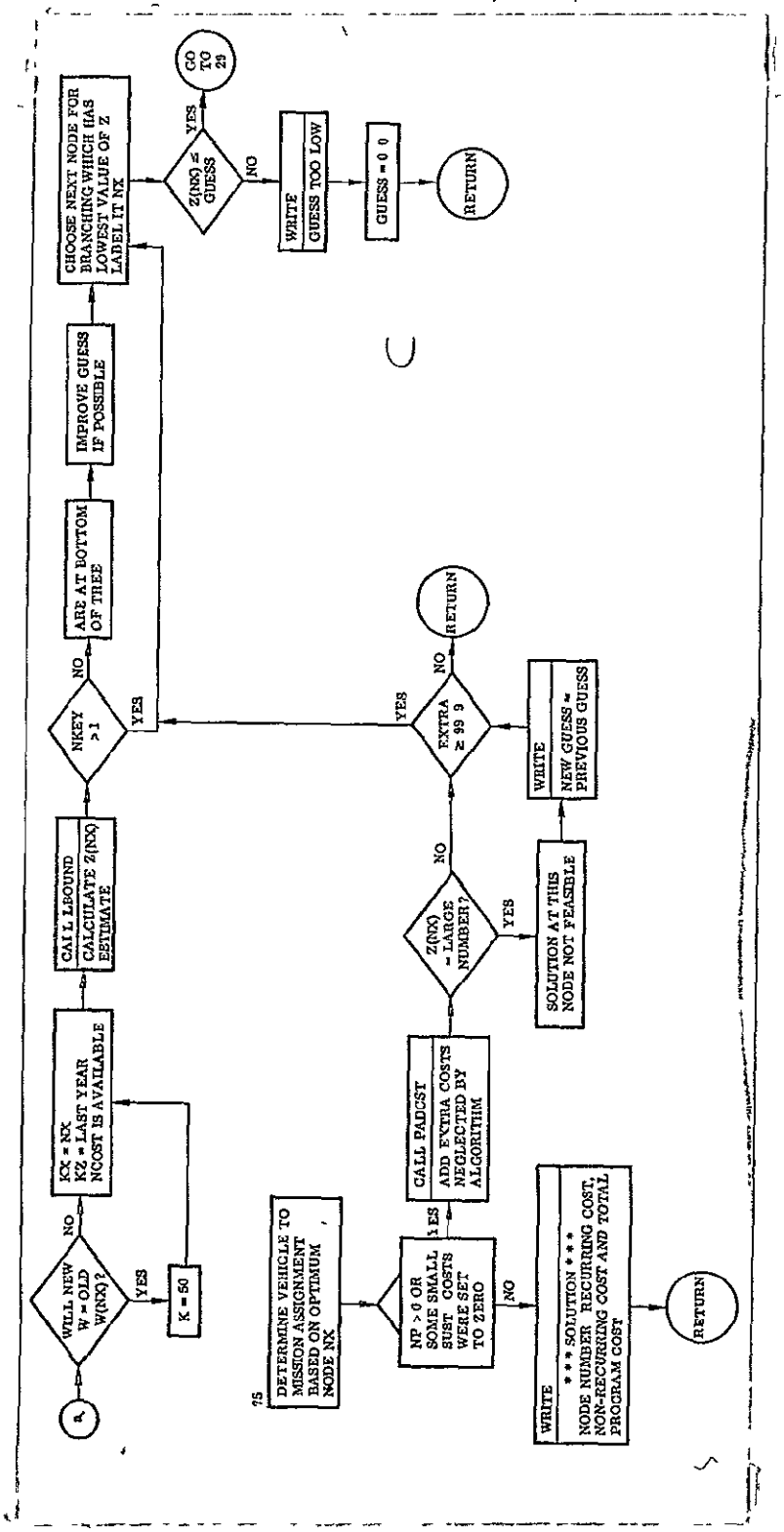
SUBROUTINE CAPABL (Cont.)



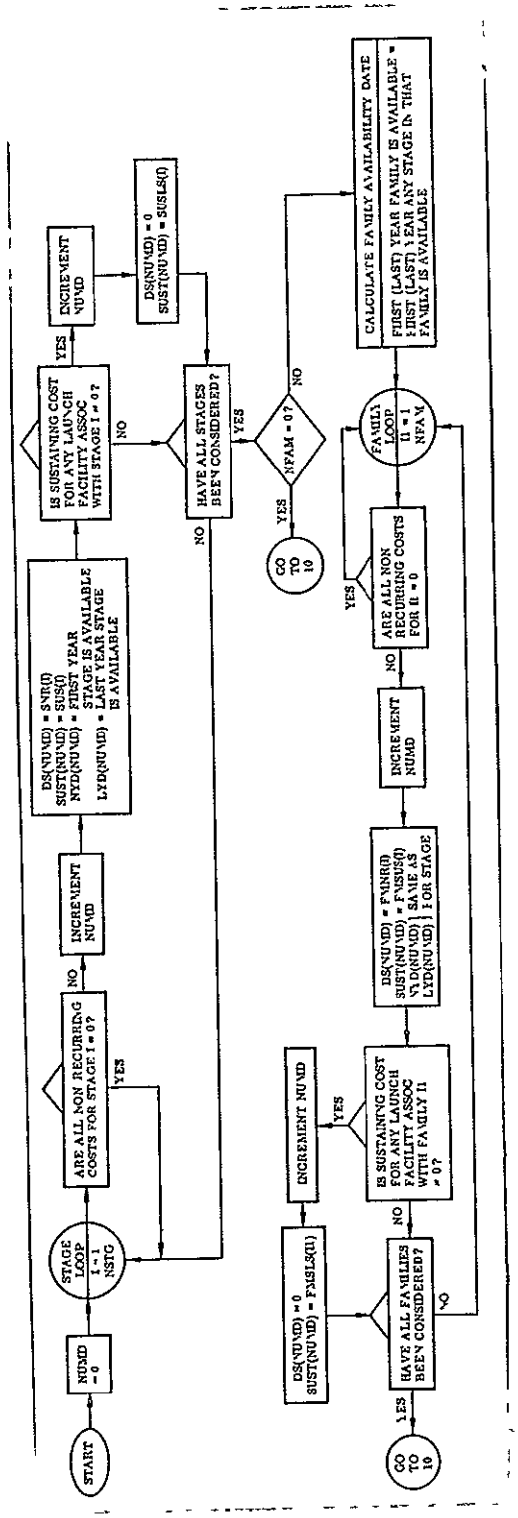
# SUBROUTINE CHOOZ

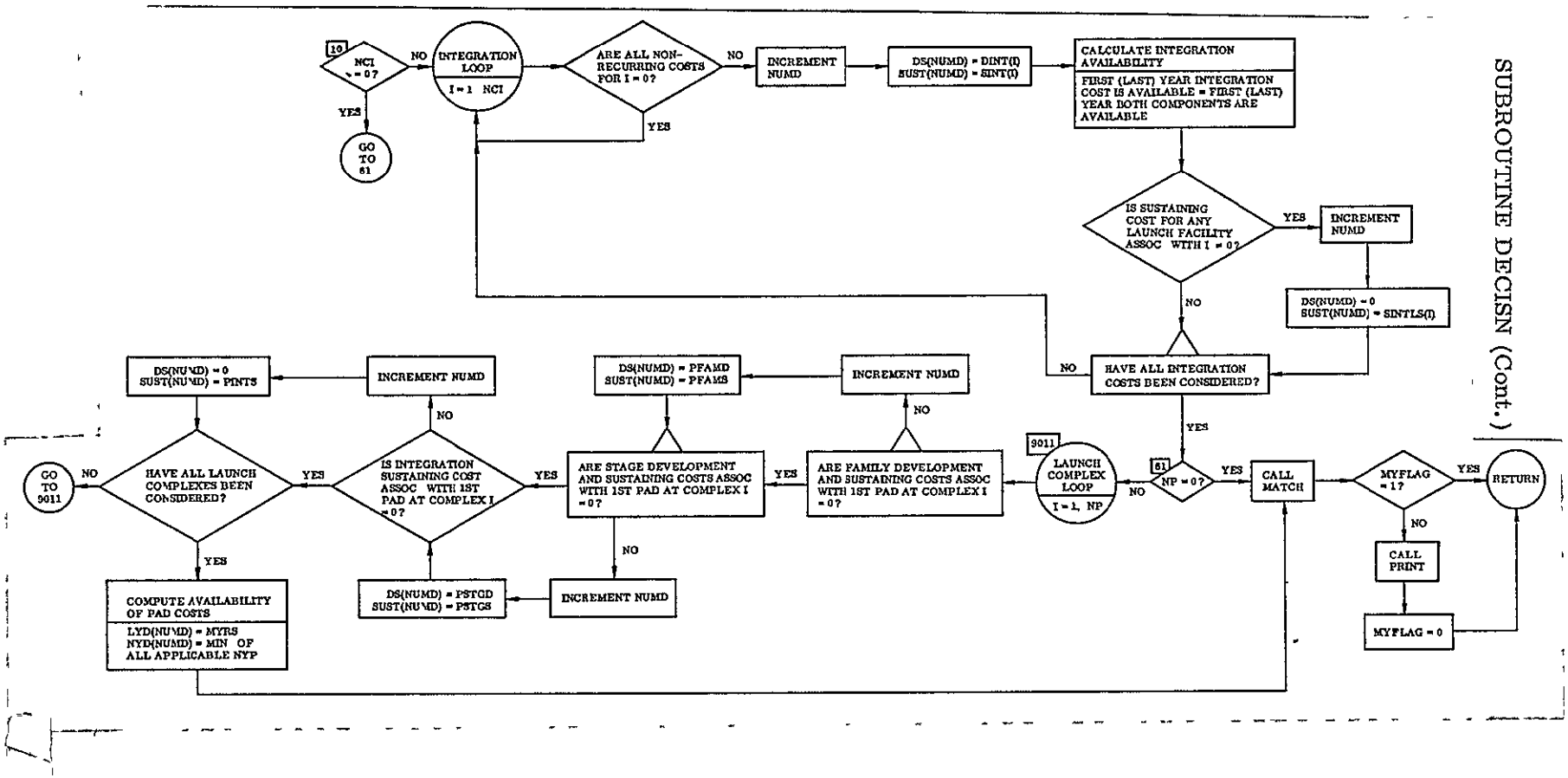


SUBROUTINE CHOOZ (Cont.)

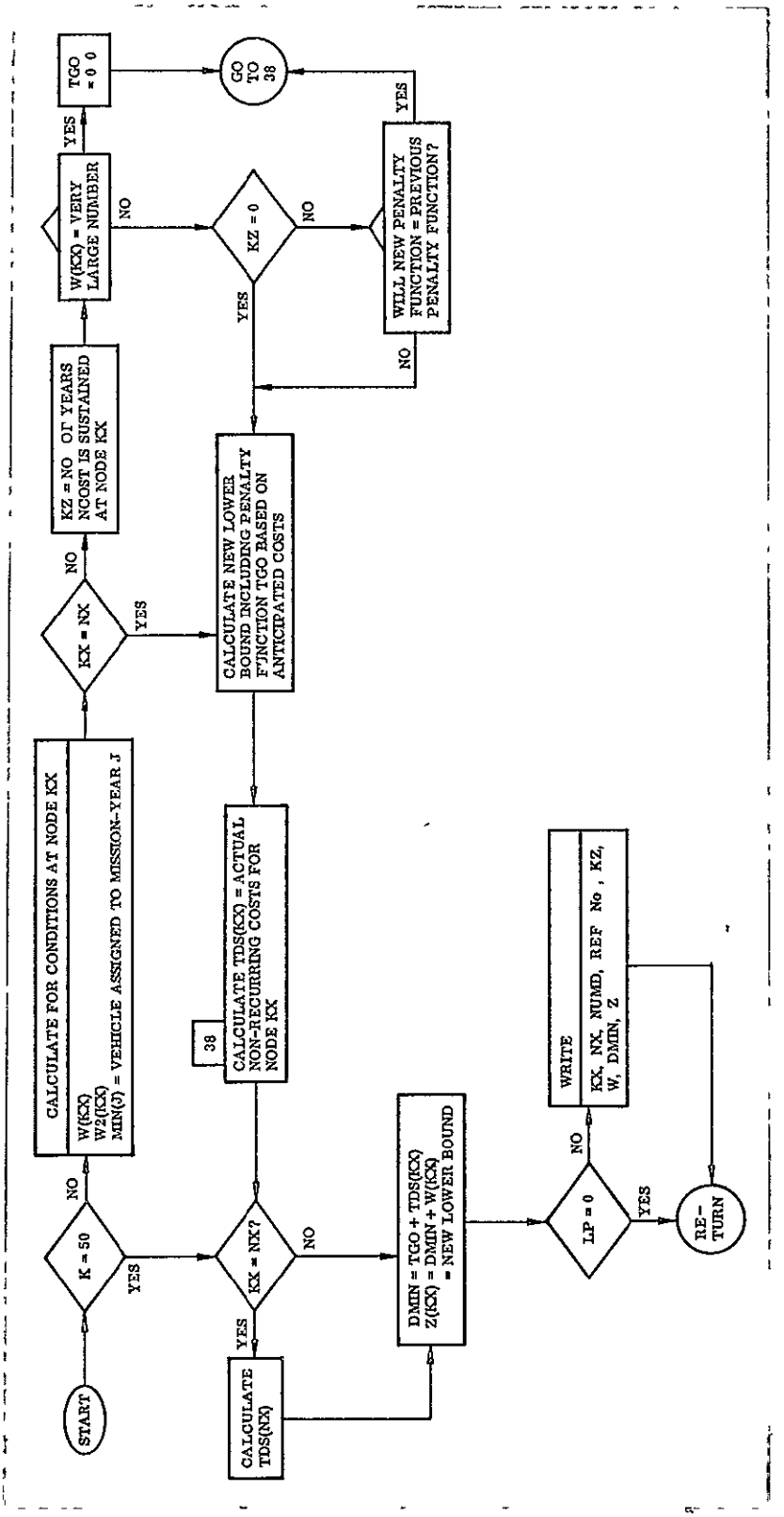


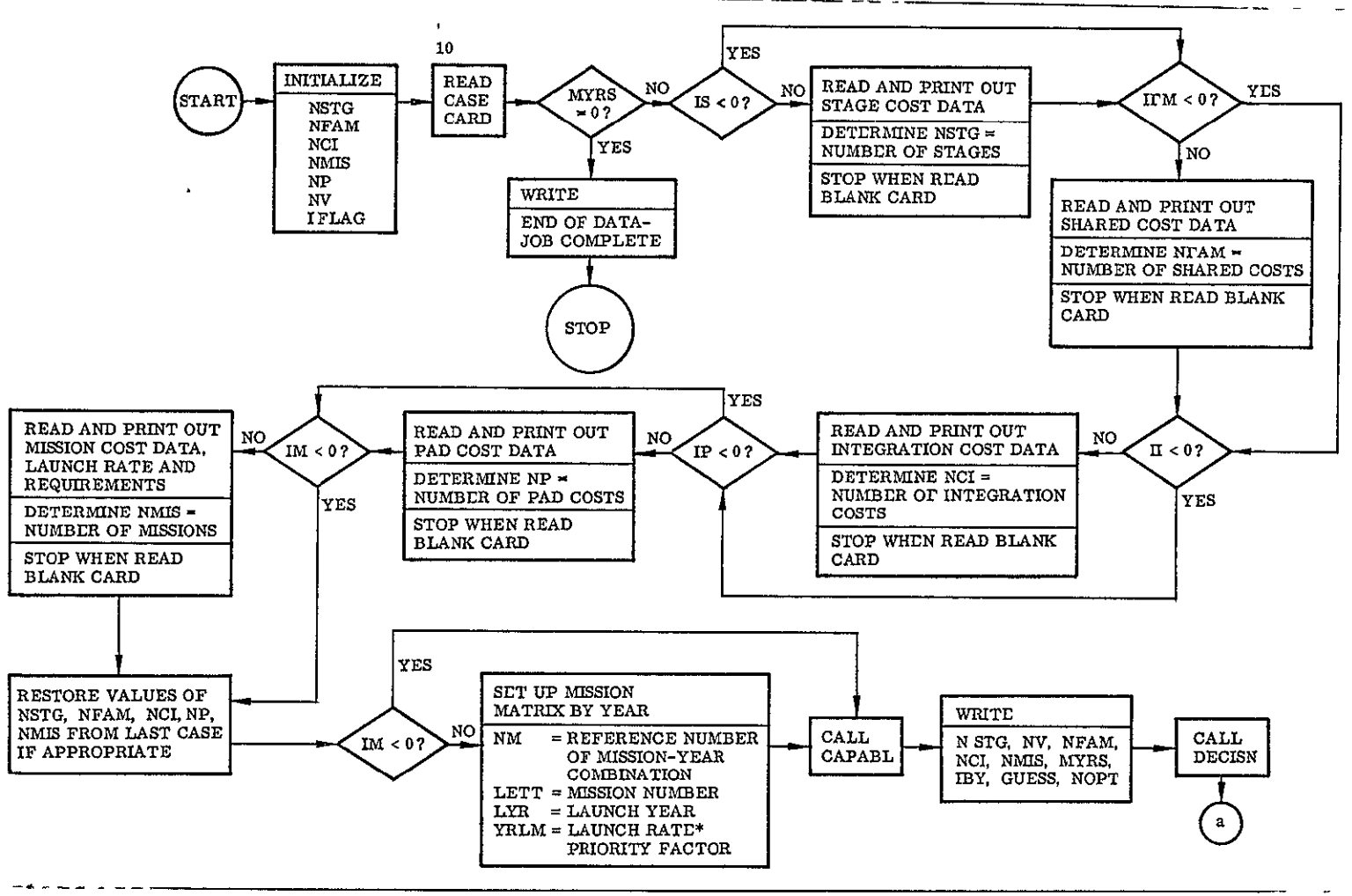
## SUBROUTINE DECISN



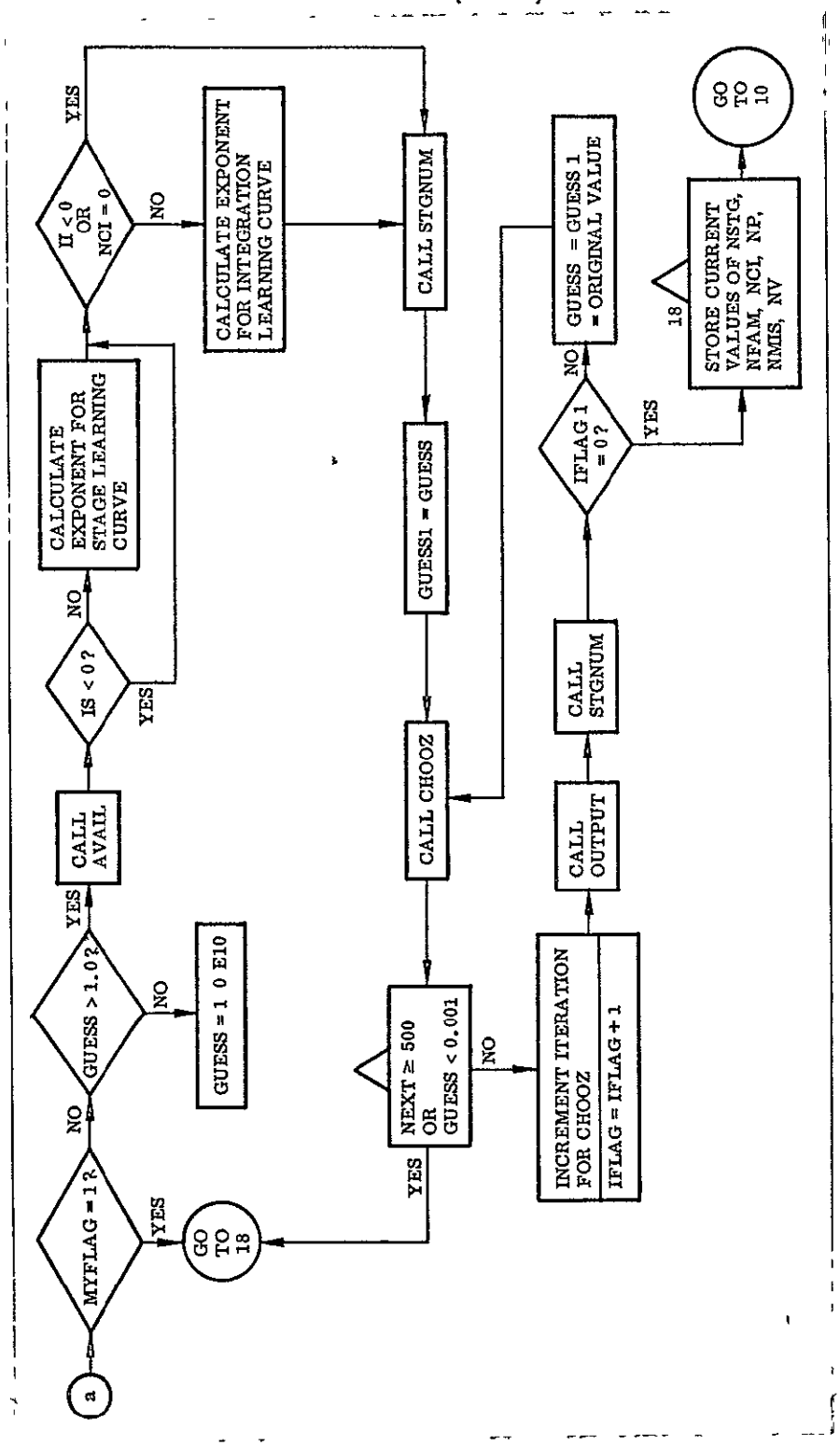

SUBROUTINE DECISN (Cont.)
10
NCI = 0?
NO
YES
GO TO 61
INTEGRATION LOOP
I = 1 NCI
ARE ALL NON-RECURRING COSTS FOR I = 0?
NO
YES
INCREMENT NUMD
DS(NUMD) = DINT(I)
SUST(NUMD) = SINT(I)
CALCULATE INTEGRATION AVAILABILITY
FIRST (LAST) YEAR INTEGRATION COST IS AVAILABLE = FIRST (LAST) YEAR BOTH COMPONENTS ARE AVAILABLE
IS SUSTAINING COST FOR ANY LAUNCH FACILITY ASSOC WITH I = 0?
YES
INCREMENT NUMD
DS(NUMD) = 0
SUST(NUMD) = SINTLS(I)
NO
HAVE ALL INTEGRATION COSTS BEEN CONSIDERED?
NO
YES
9011
LAUNCH COMPLEX LOOP
I = 1, NP
61
NP = 0?
NO
YES
CALL MATCH
MYFLAG = 1?
YES
RETURN
NO
CALL PRINT
MYFLAG = 0
ARE FAMILY DEVELOPMENT AND SUSTAINING COSTS ASSOC WITH 1ST PAD AT COMPLEX I = 0?
YES
NO
INCREMENT NUMD
DS(NUMD) = PFAMD
SUST(NUMD) = PFAMS
ARE STAGE DEVELOPMENT AND SUSTAINING COSTS ASSOC WITH 1ST PAD AT COMPLEX I = 0?
YES
NO
INCREMENT NUMD
DS(NUMD) = PSTGD
SUST(NUMD) = PSTGS
IS INTEGRATION SUSTAINING COST ASSOC WITH 1ST PAD AT COMPLEX I = 0?
YES
NO
INCREMENT NUMD
DS(NUMD) = 0
SUST(NUMD) = PINTS
HAVE ALL LAUNCH COMPLEXES BEEN CONSIDERED?
NO
YES
GO TO 9011
COMPUTE AVAILABILITY OF PAD COSTS
LYD(NUMD) = MYRS
NYD(NUMD) = MIN OF ALL APPLICABLE NYP

SUBROUTINE LBOUND

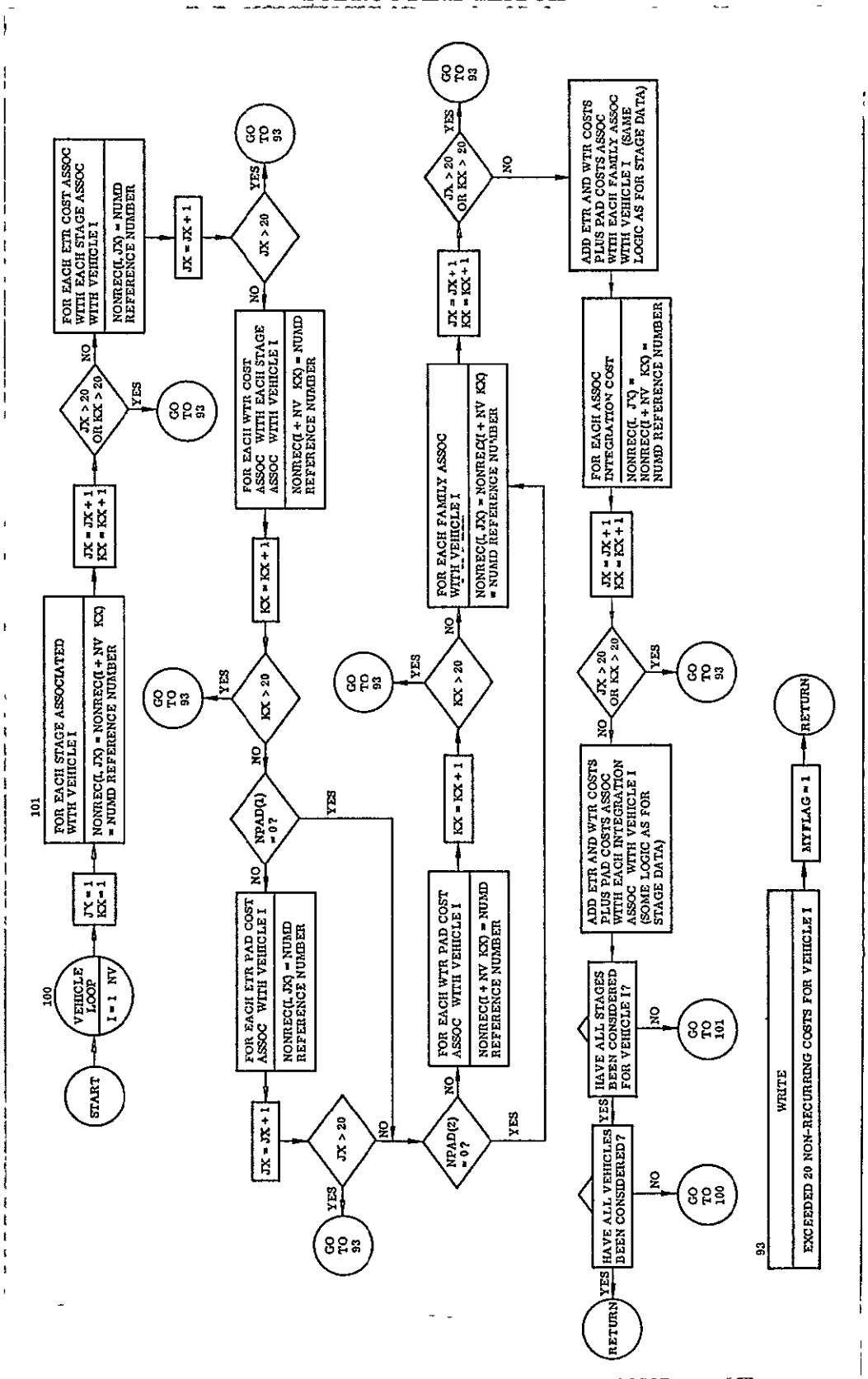


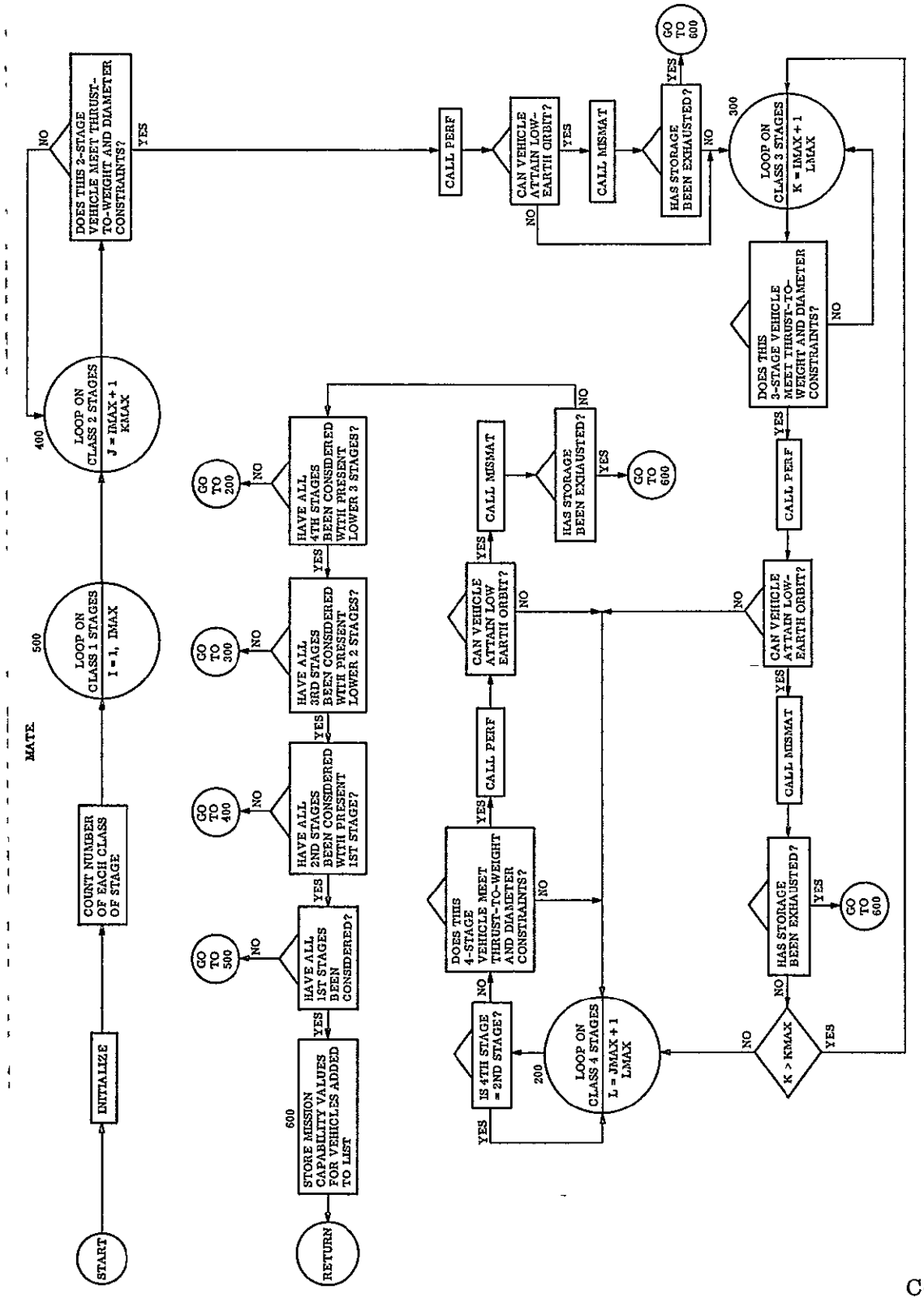

SUBROUTINE MAIN
START
INITIALIZE
NSTG
NFAM
NCI
NMIS
NP
NV
IFLAG
10
READ CASE CARD
MYRS = 0?
NO
YES
WRITE
END OF DATA-JOB COMPLETE
STOP
IS < 0?
NO
YES
READ AND PRINT OUT STAGE COST DATA
DETERMINE NSTG = NUMBER OF STAGES
STOP WHEN READ BLANK CARD
ITM < 0?
YES
NO
READ AND PRINT OUT SHARED COST DATA
DETERMINE NFAM = NUMBER OF SHARED COSTS
STOP WHEN READ BLANK CARD
II < 0?
NO
YES
READ AND PRINT OUT INTEGRATION COST DATA
DETERMINE NCI = NUMBER OF INTEGRATION COSTS
STOP WHEN READ BLANK CARD
IP < 0?
NO
YES
READ AND PRINT OUT PAD COST DATA
DETERMINE NP = NUMBER OF PAD COSTS
STOP WHEN READ BLANK CARD
IM < 0?
NO
YES
READ AND PRINT OUT MISSION COST DATA, LAUNCH RATE AND REQUIREMENTS
DETERMINE NMIS = NUMBER OF MISSIONS
STOP WHEN READ BLANK CARD
RESTORE VALUES OF NSTG, NFAM, NCI, NP, NMIS FROM LAST CASE IF APPROPRIATE
IM < 0?
NO
YES
SET UP MISSION MATRIX BY YEAR
NM = REFERENCE NUMBER OF MISSION-YEAR COMBINATION
LETT = MISSION NUMBER
LYR = LAUNCH YEAR
YRLM = LAUNCH RATE* PRIORITY FACTOR
CALL CAPABL
WRITE
N STG, NV, NFAM, NCI, NMIS, MYRS, IBY, GUESS, NOPT
CALL DECISN
a

SUBROUTINE MAIN (Cont.)

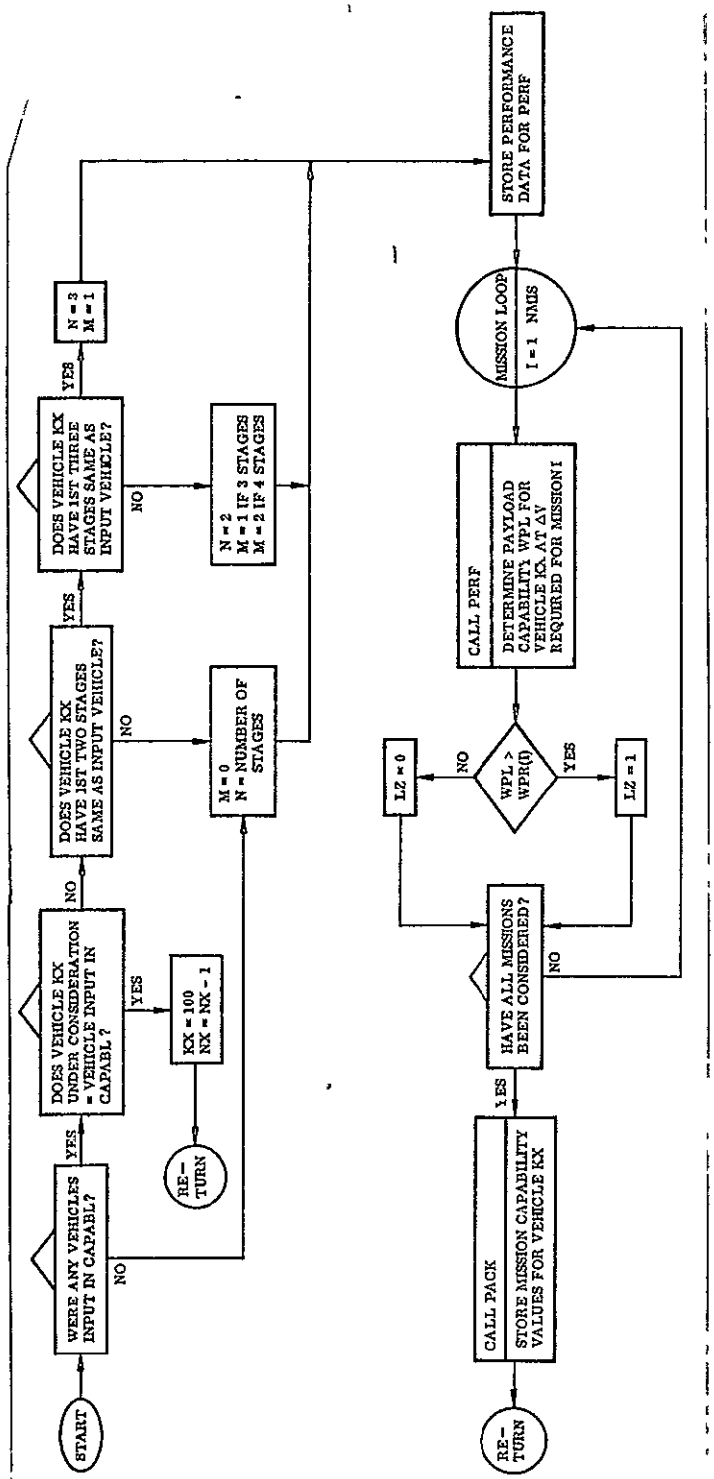


# SUBROUTINE MATCH

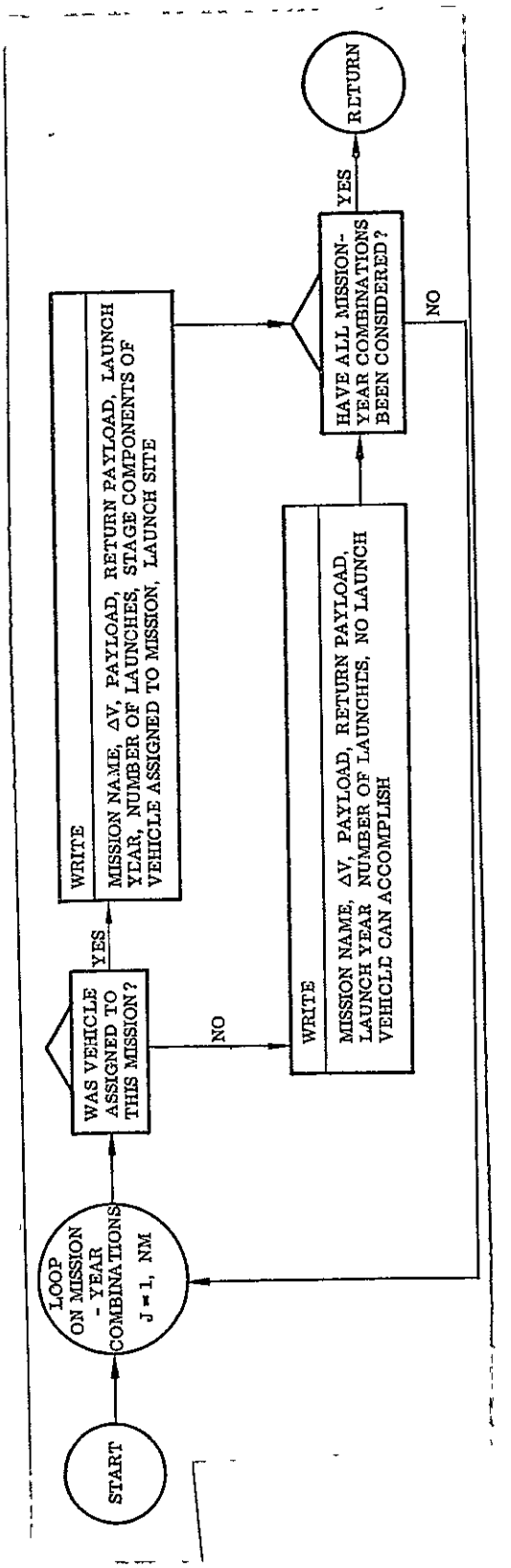



MATE
START
INITIALIZE
COUNT NUMBER OF EACH CLASS OF STAGE
LOOP ON CLASS 1 STAGES I = 1, IMAX
500
LOOP ON CLASS 2 STAGES J = IMAX + 1 KMAX
400
DOES THIS 2-STAGE VEHICLE MEET THRUST-TO-WEIGHT AND DIAMETER CONSTRAINTS?
NO
YES
HAVE ALL 1ST STAGES BEEN CONSIDERED?
GO TO 500
HAVE ALL 2ND STAGES BEEN CONSIDERED WITH PRESENT 1ST STAGE?
GO TO 400
HAVE ALL 3RD STAGES BEEN CONSIDERED WITH PRESENT LOWER 2 STAGES?
GO TO 300
HAVE ALL 4TH STAGES BEEN CONSIDERED WITH PRESENT LOWER 3 STAGES?
GO TO 200
600
STORE MISSION CAPABILITY VALUES FOR VEHICLES ADDED TO LIST
RETURN
CALL PERF
CAN VEHICLE ATTAIN LOW-EARTH ORBIT?
CALL MISMAT
HAS STORAGE BEEN EXHAUSTED?
GO TO 600
LOOP ON CLASS 3 STAGES K = IMAX + 1 LMAX
300
DOES THIS 3-STAGE VEHICLE MEET THRUST-TO-WEIGHT AND DIAMETER CONSTRAINTS?
CALL PERF
CAN VEHICLE ATTAIN LOW-EARTH ORBIT?
CALL MISMAT
HAS STORAGE BEEN EXHAUSTED?
GO TO 600
K > KMAX
LOOP ON CLASS 4 STAGES L = JMAX + 1 LMAX
200
IS 4TH STAGE = 2ND STAGE?
DOES THIS 4-STAGE VEHICLE MEET THRUST-TO-WEIGHT AND DIAMETER CONSTRAINTS?
CALL PERF
CAN VEHICLE ATTAIN LOW-EARTH ORBIT?
CALL MISMAT
HAS STORAGE BEEN EXHAUSTED?
GO TO 600

# SUBROUTINE MISMAT



SUBROUTINE OUTPUT



# SUBROUTINE PACK

## IDENTIFICATION

Subroutine PACK
Deck Name MOX01PK
Fortran IV subroutine coded in 360 Assembler Language (also MAP coded for the 7094)
Written by R. E. Slye

## PURPOSE

This subroutine is used to pack an array of integer or logical data into a smaller array in a packed binary format.

## METHOD

The unpacked (source) data is treated as an array of unsigned integers. The integer words are truncated on the left and only the N low order bits are retained. The N low order bits are then placed sequentially, left adjusted, in a packed array word until that word is filled. Packing then continues into the next word, etc., until the source data is exhausted.

Since a storage word contains 32 bits, a packed word may contain 32/N data items. Note that since only the N low order bits are retained, the largest integer item that will be represented correctly is $2^N - N$. For example, if N = 4, the packed items will represent digits from 0 to 15. For a larger integer, the packed item will in effect be the modulus of the source item.

USAGE

This subroutine has three entry points. The three entries are PACK, UNPACK, and ITEM. To pack data, the Fortran call statement is

CALL PACK (L, M, I, N)

where

- L is the name of the array containing the source data.
- M is the name of the array containing the packed data.
- I is the number of data items in L.
- N is the number of low order bits to be retained.

The array L should be dimentioned I.
The array M should be dimentioned $\lceil (I-1)/[32/N] \rceil + 1$
To unpack data, the Fortran call statement is

CALL UNPACK (L, M, I, N)

where the arguments are as listed above.

I may be less than the actual number of items in the packed array.

Packed data in the array M is unpacked and placed right adjusted in the array L. (The unused high order part of the word is cleared.)

The third entry point to the routine may be used to recover a single item from the packed array M. It is called by the Fortran statement

J = ITEM (M, I, N)

The Ith item in the packed array M is returned to the calling program.

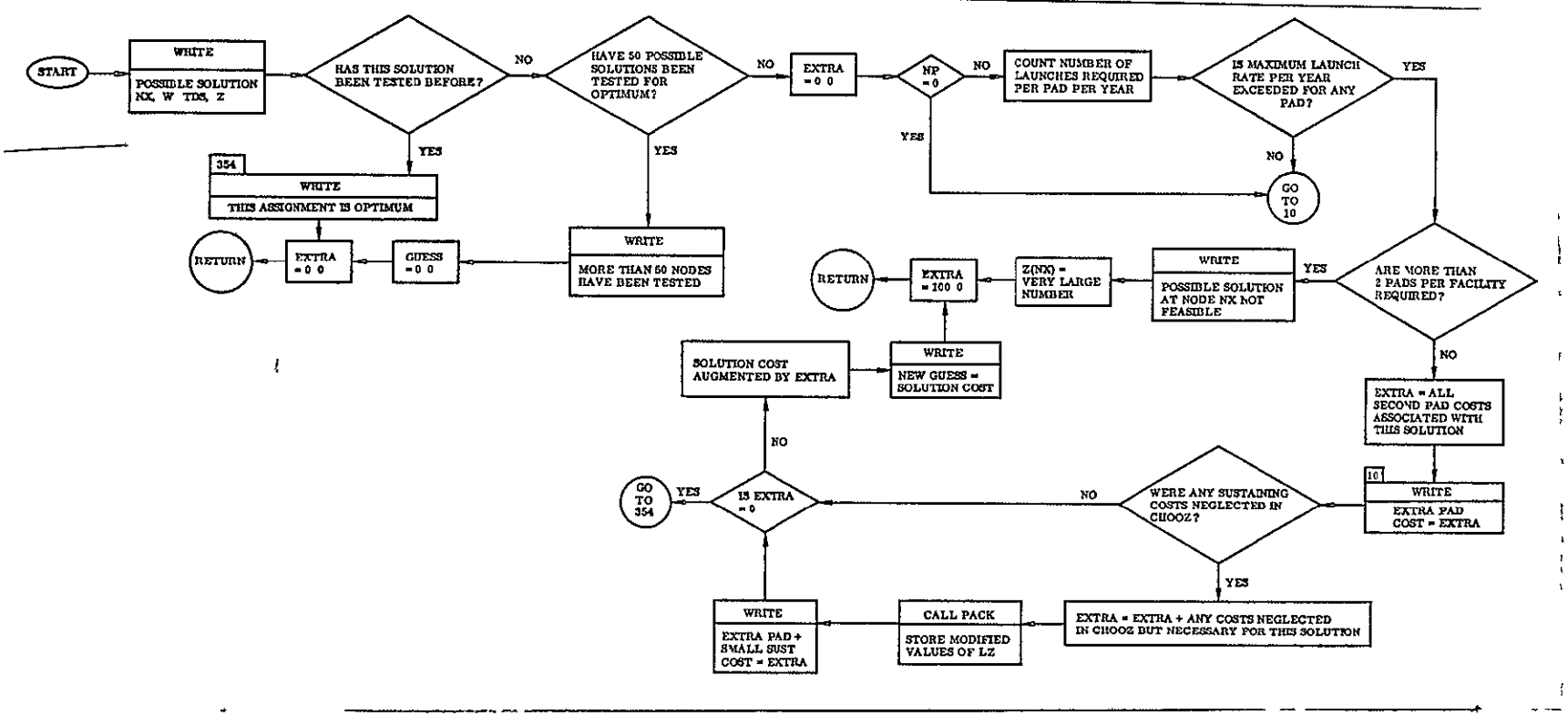
ADDITIONAL INFORMATION

If [32/N] is not an even integer, some low order bits in a packed word are unused. For example, if N = 6 the word may contain 5 items and the last 2 bits are unused The 6th item will then start at the beginning of the 2nd word.

This subroutine is also available for use on the 7094. Since the 7094 has 36 bits/word rather than 32, the data will be packed differently. This should not concern the user except that the size of the M array may be slightly smaller.

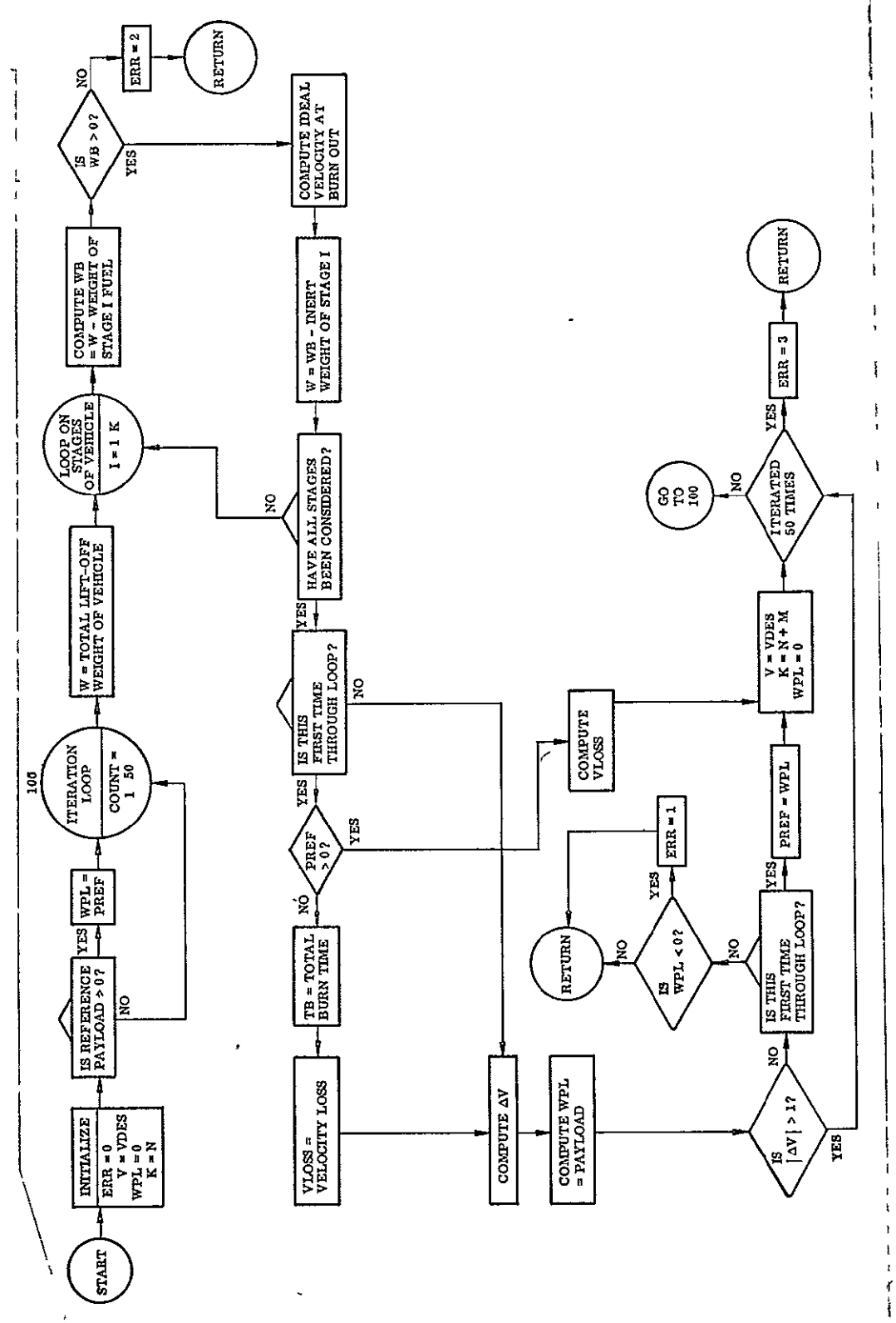
PRECAUTION

Integers in the source data may be negative. However, if negative integers are used, the results will be different on the 360 from that on the 7094 since negative 360 integers are carried in complement form.

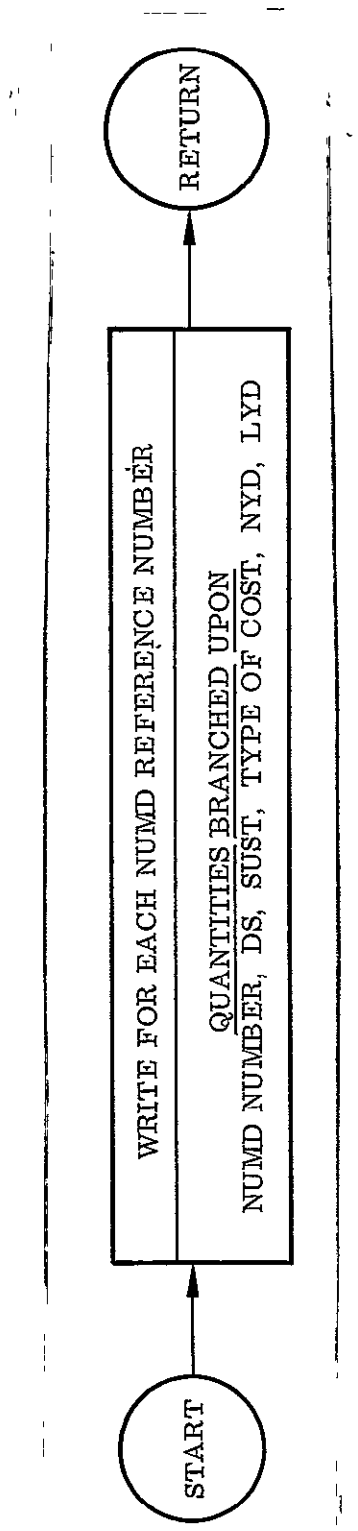


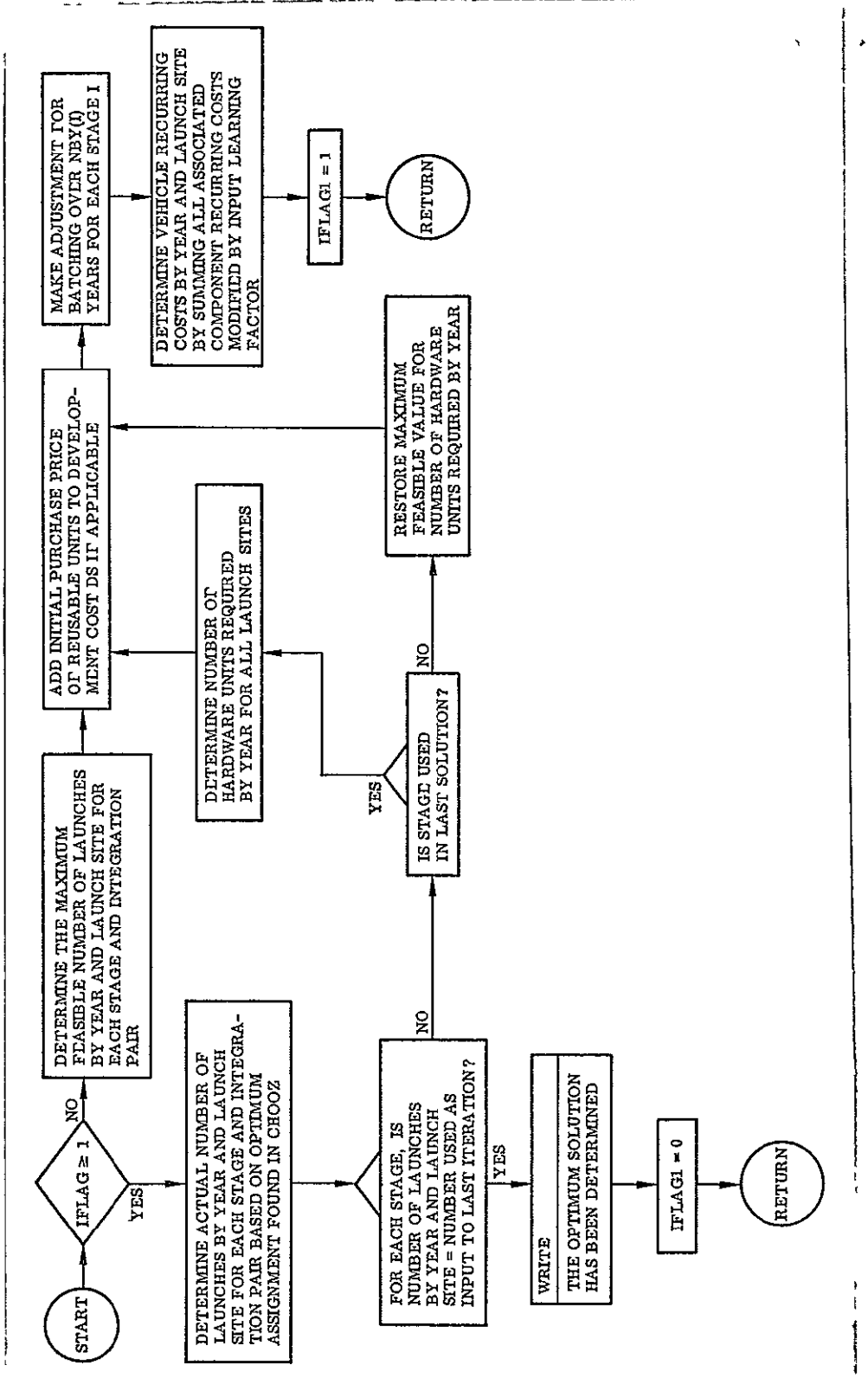

SUBROUTINE PADCST

SUBROUTINE PERF



SUBROUTINE PRINT




START
IFLAG ≥ 1
NO
YES
DETERMINE THE MAXIMUM FEASIBLE NUMBER OF LAUNCHES BY YEAR AND LAUNCH SITE FOR EACH STAGE AND INTEGRATION PAIR
DETERMINE ACTUAL NUMBER OF LAUNCHES BY YEAR AND LAUNCH SITE FOR EACH STAGE AND INTEGRATION PAIR BASED ON OPTIMUM ASSIGNMENT FOUND IN CHOOZ
FOR EACH STAGE, IS NUMBER OF LAUNCHES BY YEAR AND LAUNCH SITE = NUMBER USED AS INPUT TO LAST ITERATION?
NO
YES
IS STAGI USED IN LAST SOLUTION?
YES
NO
DETERMINE NUMBER OF HARDWARE UNITS REQUIRED BY YEAR FOR ALL LAUNCH SITES
RESTORE MAXIMUM FEASIBLE VALUE FOR NUMBER OF HARDWARE UNITS REQUIRED BY YEAR
ADD INITIAL PURCHASE PRICE OF REUSABLE UNITS TO DEVELOPMENT COST DS IF APPLICABLE
MAKE ADJUSTMENT FOR BATCHING OVER NBY(I) YEARS FOR EACH STAGE I
DETERMINE VEHICLE RECURRING COSTS BY YEAR AND LAUNCH SITE BY SUMMING ALL ASSOCIATED COMPONENT RECURRING COSTS MODIFIED BY INPUT LEARNING FACTOR
IFLAGI = 1
RETURN
WRITE
THE OPTIMUM SOLUTION HAS BEEN DETERMINED
IFLAGI = 0
RETURN

Appendix D
PROGRAM LISTING

D-1 DESCRIPTION

A compile-and-save Fortran listing of each subroutine in the launch vehicle/mission assignment program is included in this section. Storage requirements for each subroutine are listed on the output along with the code name under which the subroutine was saved. Total storage requirements are listed at the beginning of the sample case presented in Appendix B. Comment cards are liberally placed throughout the deck so that new users may familiarize themselves with the logical function of each subsection within the program.

Subroutine PACK is written in 360 assembler language and is stored for general NASA use. Therefore, its listing is provided in that language

Labeled common blocks were used for storage whenever possible to avoid long argument lists for each subroutine. These blocks are found at the beginning of each listing with a brief explanation concerning the nature of the variables found in each block.

The block labeled SCRACH stores variables only required in that subroutine so that the same storage locations may be used for storing new variables in the next subroutine. All other labeled common blocks contain variables used in several subroutines.

The listings are presented in alphabetical order according to subroutine name for easy reference.

## D.2 COMPILE-AND-SAVE LISTING

The compile-and-save listing follows

```
(17)          OS/360  FORTRAN H                                  DATE  70 113/15 08.52

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN 0002        SUBROUTINE AVAIL
                C     ADD AVAILABILITY TO CAPABILITY MATRIX
ISN 0003        INTEGER*2 LETT,LYR,MIN,NONREC,LYD,NPAD,VEH
                C     STORAGE FOR MAIN, DECISN, STGNUM, CAPABL, AVAIL, AND CHOOZ
ISN 0004        COMMON/NONDIM/ NUMD,NSTG,MYRS,NFAM,NCI,NP,NV
                C     STORAGE FOR MAIN, DECISN, STGNUM, CAPABL, AND AVAIL
ISN 0005        COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
                C     STORAGE FOR MAIN, STGNUM, CAPABL, AVAIL, AND CHOOZ
ISN 0006        COMMON/MISSON/ LETT(250),LYR(250),LTR(50),YRLM(250),MIN(250),NM
                C     STORAGE FOR MAIN, DECISN, STGNUM, AND AVAIL
ISN 0007        COMMON/STAGE/ LSA(40),SNR(41),STG(40),SUS(40),NYS(40),SUSLS(40,2),
                1  NFS(40,4)
                C     STORAGE FOR MAIN, DECISN, AVAIL, AND CHOOZ
ISN 0008        COMMON/DSCOST/ DS(100),SUST(100),NYD(100),LYD(100),NONREC(120,20)
                C     STORAGE FOR MAIN, AVAIL, AND CHOOZ
ISN 0009        COMMON/CAPMAT/ VNM(4,250),NV2
                C     STORAGE FOR AVAIL ONLY
ISN 0010        COMMON/SCRACH/ LZ(120),DUMMY(5015),VM(2,120)
                C     ***ADD AVAILABILITY TO VEHICLE CAPABILITY MATRIX***
ISN 0011        DO 39 J = 1,NM
ISN 0012        KO = LYR(J)
ISN 0013        L = LETT(J)
ISN 0014        DO 35 I = 1,NV2
ISN 0015        IF(ITEM(VM(1,I),L,1).EQ.0)  GO TO 38
ISN 0017        DO 36 M = 1,20
ISN 0018        IF(NONREC(I,M).EQ 0)  GO TO 37
ISN 0020        NO = NONREC(I,M)
ISN 0021        IF(KO.LT.NYD(NO))  GO TO 38
ISN 0023        IF(KO GT LYD(NO))  GO TO 38
ISN 0025          36 CONTINUE
ISN 0026        37 LZ(I) = 1
ISN 0027        GO TO 35
ISN 0028        38 LZ(I) = 0
ISN 0029          35 CONTINUE
ISN 0030        CALL PACK(LZ,VNM(1,J),NV2,1)
ISN 0031          39 CONTINUE
                C NM LESS THAN 136 FOR PRESENT FORMATS I E. ITER LE 3
ISN 0032          284 WRITE (6,4000)
ISN 0033        DO 421 ITER = 1,3
ISN 0034        KNM = MIN0(ITER*45,NM)
ISN 0035        K = 1 + (ITER - 1)*45
ISN 0036        285 WRITE(6,4002)  (LETT(J),J=K,KNM)

ISN 0037        DO 420 I1 = 1,NV2
ISN 0038        I = I1 + MIN0(0,ISIGN(NV,NV-I1))
ISN 0039        IA=VEH(1,I)
ISN 0040        IB=VEH(2,I)
ISN 0041        IC=VEH(3,I)
ISN 0042        ID=VEH(4,I)
ISN 0043        DO 286 J = K,KNM
ISN 0044        LZ(J61-K) = ITEM(VNM(1,J),I1,1)
ISN 0045         286 CONTINUE
ISN 0046        IF (I1 LE NV)
                1WRITE(6,4100)I,STG(IA),STG(IB),STG(IC),STG(ID),(LZ(J61-K),J=K,KNM)
ISN 0048        IF (I1 GT NV)
                1WRITE(6,4103)I,STG(IA),STG(IB),STG(IC),STG(ID),(LZ(J61-K),J=K,KNM)
ISN 0050         420 CONTINUE
ISN 0051        IF(NM LE KNM) RETURN
ISN 0053        IF(ITER EQ 2)  WRITE(6,4001)
ISN 0055        IF(ITER EQ 3)  WRITE(6,4003)
ISN 0057         421 CONTINUE
ISN 0058        RETURN
ISN 0059        4000 FORMAT (1H1,34X,51HV E H I C L E / M I S S I O N   C A P A B I L I
                1 T Y/46X,30H(1 = POSSIBLE, 0 = IMPOSSIBLE)/1HO,43X,10(2H1 ),
                2  10(2H2 ),10(2H3 ),6(2H4 )/18H VEHICLE / MISSION,9X,4(20H1 2 3 4
                35 6 7 8 9 0 ),9H1 2 3 4 5//)
ISN 0060        4001 FORMAT(1HO/ 1HO,25X,4(2H4 ),10(2H5 ),10(2H6 ),10(2H7 ),10(2H8 ),
                1  2H9 /18H VEHICLE / MISSION,9X,9H6 7 8 9 0,4(20H1 2 3 4 5 6 7 8 9
                2 0 )//)
ISN 0061        4002 FORMAT (1HO,7X,14HMISSION NUMBER,   4X,45I2)
ISN 0062        4003 FORMAT(1HO/ 1HO,25X,9(2H9 ),10(2H10),10(2H11),10(2H12),6(2H13)/
                1    18H VEHICLE / MISSION, 9X,4(20H1 2 3 4 5 6 7 8 9 0 ),
                2    9H1 2 3 4 5//)
ISN 0063        4100 FORMAT (1H ,I2,1X,4(A4,1X),2HE ,45I2)
ISN 0064        4103 FORMAT (1H ,I2,1X,4(A4,1X),2HM ,45I2)
ISN 0065            END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
        VARIABLE OPTIONS USED - SIZE=(126976,24576)                DEFAULT OPTION(S) USED
IEW0000     NAME=NOX02AV(R)
IEW0461  ITEM
IEW0461  PACK
IEW0461  IBCOM=
```

CROSS REFERENCE TABLE

| CONTROL SECTION NAME | ORIGIN | LENGTH |
|---|---|---|
| AVAIL | 00 | 7D0 |
| NONDIM | 7D0 | 1C |
| VEHCLE | 7F0 | 480 |
| MISSON | CA0 | A90 |
| STAGE | 1730 | 6E4 |
| DSCOST | 1E18 | 1838 |
| CAPMAT | 3650 | FA4 |
| SCRACH | 45F8 | 53FC |

ENTRY

| NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|

| LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION | LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION |
|---|---|---|---|---|---|
| 2F8 | NONDIM | NONDIM | 2FC | VEHCLE | VEHCLE |
| 300 | MISSON | MISSON | 304 | STAGE | STAGE |
| 308 | DSCOST | DSCOST | 30C | CAPMAT | CAPMAT |
| 310 | SCRACH | SCRACH | 314 | SCRACH | SCRACH |
| 318 | ITEM | $UNRESOLVED | 31C | PACK | $UNRESOLVED |
| 320 | IBCOM= | $UNRESOLVED | 268 | SCRACH | SCRACH |
| 270 | CAPMAT | CAPMAT | | | |

ENTRY ADDRESS 00
TOTAL LENGTH 99F8

****NOX02AV  NOW REPLACED IN DATA SET

DIAGNOSTIC MESSAGE DIRECTORY

IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED.

```
1(17)          OS/360  FORTRAN H                              DATE  70 113/15 09.23

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002          SUBROUTINE CAPABL
                  C   INPUT VEHICLE INFORMATION AND COMPUTE CAPABILITY MATRIX
   ISN 0003          REAL ISP,LENT
   ISN 0004          INTEGER*2 LETT,LYR,MIN,VEH,NPAD,NMULT
                  C STORAGE FOR MAIN, DECISN, STGNUM, CAPABL, AVAIL, AND CHOOZ
   ISN 0005          COMMON/NONDIM/ NUMD,NSTG,MYRS,NFAM,NCI,NP,NV
                  C STORAGE FOR MAIN, DECISN, STGNUM, CAPABL, AND AVAIL
   ISN 0006          COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
                  C STORAGE FOR MAIN, STGNUM, CAPABL, AVAIL, AND CHOOZ
   ISN 0007          COMMON/MISSON/ LETT(250),LYR(250),LTR(50),YRLM(250),MIN(250),NM
                  C    STORAGE FOR MAIN, STGNUM, CHOOZ, AND CAPABL
   ISN 0008          COMMON/VEHREC/ RECUR(60,20,2),NMULT(60,50),IFLAG
                  C STORAGE FOR MAIN AND CAPABL
   ISN 0009          COMMON/MISREQ/ IVEHA(50),VLR(50),WPR(50),NPLS(50),MR(50),NRR(50),
                     1   RPLM(50),NTRIP(50)
   ISN 0010          COMMON/VEHIN/ IV,KNV,IS,NMIS,NOPT,KODE(40),KODEP(30),IP,RPLO(50)
                  C STORAGE FOR CAPABL ONLY - SAVE FOR BATCHING
   ISN 0011          COMMON/BATCH/ NRP(60),
                     1          B1(60),B2(60),B3(60),B4(60),KODEV(60),NVS(60),NRV(60)
   ISN 0012          COMMON/SCRACH/ LZ(50),MZ(50),VMH(2,60),NEH(4,60),NPAX(2,60),
                     1 NST(40),THRT(40),DIAM(40),TSL(40),LENT(40),WTFU(40),WTIN(40),
                     2  ISP(40),WINT(3,60),KX,NX,NGHT(40),WF(41,WT(4),ISPA(4),THUT(4),
                     3  PRT(60), DUMMY(3937),VM(2,120)
                  C
   ISN 0013          DO 281 J = 1,60
   ISN 0014          IF(IV LT 0 AND.J.GT.KNV) GO TO 5004
   ISN 0016          IF(IV.LT 0 AND IS.LT.0)  GO TO 27
   ISN 0018          IF(IV LT 0)  GO TO 15
   ISN 0020          READ(5,106)  (NEH(I,J),I=1,4),B1(J),B2(J),B3(J),B4(J),KODEV(J)
   ISN 0021          IF(KODEV(J) EQ 0) GO TO 5002
   ISN 0023          DO 700 I = 1,4
   ISN 0024      700 VEH(I,J) = NEH(I,J)
   ISN 0025          READ(5,108)NVS(J),NRV(J),NRP(J),(NPAX(I,J),I=1,2),(NYP(I,J),I=1,2)
   ISN 0026          DO 701 I = 1,2
   ISN 0027      701 NPAD(I,J) = NPAX(I,J)
   ISN 0028       15 DO 26 I = 1,4
   ISN 0029          IF(VEH(I,J) EQ 0)  GO TO 27
   ISN 0031          DO 25 K = 1,NSTG
   ISN 0032          IF(VEH(I,J).NE.KODE(K))  GO TO 25
   ISN 0034          VEH(I,J)  = K
   ISN 0035          GO TO 26
   ISN 0036       25 CONTINUE
```

```
ISN 0037        26 CONTINUE
ISN 0038        27 NV = J
ISN 0039           IF(IV.LT 0.AND IP.LT 0) GO TO 9007
ISN 0041           DO 9008 I = 1,2
ISN 0042           IF (NPAD(I,J) EQ.0) GO TO 9008
ISN 0044           DO 9009 K = 1,NP
ISN 0045           IF (NPAD(I,J) NE KODEP(K)) GO TO 9009
ISN 0047           NPAD(I,J) = K
ISN 0048           GO TO 9008
ISN 0049      9009 CONTINUE
ISN 0050      9008 CONTINUE
ISN 0051      9007 C1 = B1(J)
ISN 0052           C2 = B2(J)
ISN 0053           C3 = B3(J)
ISN 0054           C4 = B4(J)
ISN 0055           DO 28 I=1,NMIS
ISN 0056           NMULT(J,I) = 1
ISN 0057           LZ(I) = 1
ISN 0058           IF (IVEHA(I) EQ.0) GO TO 21
ISN 0060           IF (IVEHA(I) NE KODEV(J)) GO TO 8024
ISN 0062           GO TO 8020
ISN 0063        21 VLX=VLR(I)-25573
ISN 0064           IF(VLX GE C4) GO TO 8024
ISN 0066           WP=EXP(C1-C2*VLX-C3/(C4-VLX))
ISN 0067           DO 22 JJ = 1,4
ISN 0068           IF(VEH(5-JJ,J) EQ 0)  GO TO 22
ISN 0070           LL = VEH(5-JJ,J)
ISN 0071           IF(RPLH(I) GE 1 0.AND RPLO(LL) LT..001)  GO TO 8024
ISN 0073           IF(RPLH(I) & RPLO(LL) LE  001)  GO TO 23
ISN 0075           GO TO 255
ISN 0076        22 CONTINUE
ISN 0077       255 IF(NTRIP(I).EQ 0)  GO TO 8024
ISN 0079           IF(WP GE WPR(I))    GO TO 24
ISN 0081           NMULT(J,I) = INT(WPR(I)/WP & .99)
ISN 0082           IF(NMULT(J,I) GT.NTRIP(I))  GO TO 8024
ISN 0084           GO TO 24
ISN 0085        23 IF (WP LE.WPR(I)) GO TO 8024
ISN 0087        24 IF (NOPT NE 3) GO TO 8020
ISN 0089           IF (NPLS(I).EQ 0) GO TO 8023
ISN 0091           IF (NPLS(I) NE.NVS(J)) GO TO 8024
ISN 0093      8023 IF (NRR(I) GT.NRP(J)) GO TO 8024
ISN 0095           IF (MR(I) EQ 0 OR MRV(J) EQ.1)  GO TO 8020
```

```
ISN 0097      8024 LZ(I) = 0
ISN 0098      8020 IF(NP NE.0)  MZ(I) = LZ(I)
ISN 0100           IF(LZ(I) EQ 0 OR NP EQ 0)  GO TO 28
              C  NOTE - LTR = 1 OR 2 FOR REUSABLE COMPONENTS NEED NOT BE  ETR OR WTR
ISN 0102           IF (LTR(I) EQ.2) GO TO 9019
ISN 0104           MZ(I) = 0
ISN 0105           IF (IVEHA(I) EQ KODEV(J)) GO TO 28
ISN 0107           IF (NPAD(1,J) EQ 0) LZ(I) = 0
ISN 0109           GO TO 28
ISN 0110      9019 LZ(I) = 0
ISN 0111           IF (IVEHA(I) EQ.KODEV(J)) GO TO 28
ISN 0113           IF (NPAD(2,J) EQ 0) MZ(I) = 0
ISN 0115        28 CONTINUE
ISN 0116           CALL PACK(LZ,VM(1,J),NMIS,1)
ISN 0117           IF (NP NE 0) CALL PACK (MZ,VMM(1,J),NMIS,1)
ISN 0119       281 CONTINUE
ISN 0120      5002 IF(NOPT NE 2)  GO TO 5004
ISN 0122           WRITE(6,111)
              C   CARDS MUST BE IN SAME ORDER AS INPUT STAGE CARDS
              C   ALL STAGES NOT TO BE USED IN MATCHING SCREEN MUST BE AT END OF DATA SET
ISN 0123           DO 30 I = 1,NSTG
ISN 0124           READ(5,109) J,NST(I),THRT(I),DIAM(I),TSL(I),LENT(I),WTFU(I),
                  1  WTIN(I),ISP(I)
ISN 0125           IF(J EQ 0)  GO TO 31
ISN 0127           WRITE(6,112) I,NST(I),THRT(I),DIAM(I),TSL(I),LENT(I),WTFU(I),WTIN
                  1(I),ISP(I)
ISN 0128           IF(J NE KODE(I))   GO TO 5005
ISN 0130        30 CONTINUE
ISN 0131        31 CALL MATE
ISN 0132      5004 IF (NP EQ 0) RETURN
ISN 0134           DO 5003 J = 1,NV
ISN 0135           VM(1,J&NV) = VMM(1,J)
ISN 0136      5003 VM(2,J&NV) = VMM(2,J)
ISN 0137           RETURN
ISN 0138      5005 WRITE(6,110)
ISN 0139           RETURN
ISN 0140       106 FORMAT (4I2,4E13.6,18X,I2)
ISN 0141       108 FORMAT (3X,3I2,4I3)
ISN 0142       109 FORMAT(I4,I5,7F10 0)
ISN 0143       110 FORMAT(27HOSTAGE CARDS IN WRONG ORDER)
ISN 0144       111 FORMAT(1H1,8H STG NST,9X,4HTHRT,9X,4HDIAM,9X,4H TSL,9X,4HLENT,9X,
                  1   4HWTFU,9X,4HWTIN,10X,3HISP//)
```

```
   ISN 0145       112 FORMAT(1HO,2I4,7F13 2)
   ISN 0146           END
******* END OF COMPILATION ******
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
           VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02CP(R)
IEW0461  MATE
IEW0461  PACK
IEW0461  EXP
IEW0461  IBCOM#
```

CROSS REFERENCE TABLE

| CONTROL SECTION NAME | ORIGIN | LENGTH |
|---|---|---|
| CAPABL | 00 | B98 |
| NONDIM | B98 | 1C |
| VEHCLE | BB8 | 480 |
| MISSON | 1068 | A90 |
| VEHREC | 1AF8 | 3CF4 |
| MISREQ | 57F0 | 640 |
| VEHIN | 5E30 | 1F8 |
| BATCH | 6028 | 780 |
| SCRACH | 67A8 | 53FC |

ENTRY

| NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|

| LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION |
|---|---|---|
| 1C8 | NONDIM | NONDIM |
| 1D0 | MISSON | MISSON |
| 1D8 | VEHREC | VEHREC |
| 1E0 | MISREQ | MISREQ |
| 1E8 | BATCH | BATCH |
| 1F0 | SCRACH | SCRACH |
| 1F8 | MATE | $UNRESOLVED |
| 200 | EXP | $UNRESOLVED |
| 118 | SCRACH | SCRACH |
| 128 | SCRACH | SCRACH |

| LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION |
|---|---|---|
| 1CC | VEHCLE | VEHCLE |
| 1D4 | VEHREC | VEHREC |
| 1DC | VEHREC | VEHREC |
| 1E4 | VEHIN | VEHIN |
| 1EC | SCRACH | SCRACH |
| 1F4 | SCRACH | SCRACH |
| 1FC | PACK | $UNRESOLVED |
| 204 | IBCOM# | $UNRESOLVED |
| 120 | VEHIN | VEHIN |
| 130 | VEHIN | VEHIN |

ENTRY ADDRESS 00
TOTAL LENGTH BBA8

****MOX02CP  NOW REPLACED IN DATA SET

```
FORTRAN IV G LEVEL 1, MOD 4           CHOOZ           DATE = 70113        15/09/56          PAGE 0001

0001              SUBROUTINE CHOOZ
                C    DETERMINE OPTIMUM VEHICLE TO MISSION ASSIGNMENT
0002              REAL NPERPD,NPUSED
0003              INTEGER*2 LETT,LYR,MIN,NONREC,LYD,MORE,NSAVE, KOUT,KEEP,NPAD,VEH,
                 1   NPFAM,NPSTG,NPINTL,NPINTU,NMULT
                C STORAGE FOR MAIN, DECISN, STGNUM, CAPABL, AVAIL, AND CHOOZ
0004              COMMON/NONDIM/ NUMD,NSTG,MYRS,NFAM,NCI,NP,NV
0005              COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
                C STORAGE FOR MAIN, DECISN, STGNUM, AVAIL, AND CHOOZ
0006              COMMON/STAGE/ LSA(40),SNR(41),STG(40),SUS(40),NYS(40),SUSLS(40,2),
                 1  NPS(40,4)
                C STORAGE FOR MAIN, STGNUM, CAPABL, AVAIL, AND CHOOZ
0007              COMMON/MISSON/ LETT(250),LYR(250),LTR(50),YRLH(250),MIN(250),NM
                C STORAGE FOR MAIN, DECISN, AVAIL, AND CHOOZ
0008              COMMON/DSCOST/ DS(100),SUST(100),NYD(100),LYD(100),NONREC(120,20)
                C STORAGE FOR MAIN, DECISN, AND CHOOZ
0009              COMMON/PADS/ PFAMD(30,5,3),PFAMS(30,5,3),NPFAM(30,5),
                 1  PSTGD(30,10,3),PSTGS(30,10,3),NPSTG(30,10),PINTS(30,5,3),
                 2  NPINTL(30,5),NPINTU(30,5),PAD(30)
                C    STORAGE FOR MAIN, STGNUM, CHOOZ, AND CAPABL
0010              COMMON/VEHREC/ RECUR(60,20,2),NMULT(60,50),IFLAG
                C STORAGE FOR MAIN, AVAIL, AND CHOOZ
0011              COMMON/CAPMAT/ VNM(4,250),NV2
                C STORAGE FOR MAIN AND CHOOZ
0012              COMMON/CHOOZE/ NEXT,GUESS,NPERPD(30),LP
                C STORAGE FOR CHOOZ ONLY
0013              COMMON/SCRACH/ TDS(500),W2(500),Z(500),W(500),NSAVE(10),
                 1  NODE(5,500),LZ(100),MZ(120),KEEP(100),COST(2,250),WR(100)
                C   STORAGE FOR CHOOZ AND PADCST
0014              COMMON/SAVEPT/MORE(50),KOUT(100),LOUT,EXTRA,SAVS(20)
                C STORAGE FOR CHOOZ AND LBOUND
0015              COMMON/SAVELB/ KX,KI,NX,NCOST,K,KZ
                C  THE FOLLOWING DATA STATEMENT-USED ON THE 360- PLACES 1'S IN ALL LOCATIONS
0016              DATA IONES /-1/
                C
0017              IF(MYRS GT 10) GO TO 2
0018              KI = 1
0019              KNEX = MYRS
0020              GO TO 3
0021            2 KI = 2
0022              KNEX = (MYRS + 1)/2
0023            3 CONTINUE
                C

FORTRAN IV G LEVEL 1, MOD 4           CHOOZ           DATE = 70113        15/09/56          PAGE 0002

                C     ***   INITIALIZE FUNCTIONS   ***
0024            7 NEXT=1
0025              NX=1
0026              KPNX = 10
0027              DO 16 I = 1,5
0028           16 NODE(I,1)=IONES
0029              DO 400 I = 1,50
0030          400 MORE(I) = 0
0031              WRITE (6,205)
                C
                C     ***   FIND W(1) = SUM OF COLUMN MINIMUMS OF FIRST CASE   ***
0032              W(1)=0 0
0033              W2(1) = 0 0
0034              TDS(1) = 0 0
0035              DO 19 J=1,NM
0036              IY = LYR(J)
0037              JX = LETT(J)
0038              ITR = LTR(JX)
0039              CALL UNPACK(MZ,VNM(1,J),NV2,1)
0040              COST(1,J) = 1 0E30
0041              COST(2,J) = 1 0E30
0042              DO 18 I=1,NV2
0043              IF(MZ(I) EQ 0)  GO TO 18
0044              II = I + MIN0(0,ISIGN(NV,NV-I))
0045              X = NMULT(II,JX)
0046              CX = YRLH(J)*RECUR(II,IY,ITR)*X
0047              IF(CX.GE.COST(2,J))  GO TO 18
0048              IF(CX LT.COST(1,J))  GO TO 176
0049              COST(2,J) = CX
0050              GO TO 18
0051          176 COST(2,J) = COST(1,J)
0052              COST(1,J)  = CX
0053              MIN(J) = I
0054           18 CONTINUE
0055              IF(COST(1,J) LT.1 0E25)  GO TO 20
0056              YRLH(J)=0 0
0057              COST(1,J) = 0 0
0058              COST(2,J) = 0 0
0059           20 W2(1) = W2(1) + COST(2,J)
0060           19 W(1) = W(1) + COST(1,J)
0061              IF(NUMD EQ.0)  RETURN
                C
                C  PRESET SMALL SUST COSTS TO ZERO SO ALGORITHM IGNORES THEM
```

```
FORTRAN IV G LEVEL 1, MOD 4            CHOOZ            DATE = 70113        15/09/56

      C
0062            CALL UNPACK(LZ,NODE(1,1),NUMD,4)
0063            IF(IFLAG GT.0)  GO TO 22
0064            DO 21 I = 1,NUMD
0065           21 KOUT(I) = 0
      C   LOUT = NUMBER OF SUST COSTS GT 0 WHICH HAVE BEEN SET TO 0
0066            LOUT = 0
0067            X = MYRS
0068            G = 4 0/X
0069            IF(GUESS.LT 1.0E8)  G = GUESS/( 50 0*X)
0070            DO 27 I = 1,NUMD
0071            IF(SUST(I).LT. 001)  GO TO 27
0072            IF(SUST(I).GT.G)  GO TO 27
0073            LOUT = LOUT + 1
0074            IF(LOUT.GT 20)  GO TO 28
0075            KOUT(I) = LOUT
0076            SAVS(LOUT) = SUST(I)
0077            SUST(I) = 0 0
0078            IF(DS(I).GE..5)  GO TO 27
0079            LZ(I) = KNEX
0080           27 CONTINUE
0081           28 IF(LOUT GT 0)  CALL PACK(LZ,NODE(1,1),NUMD,4)
0082            GO TO 29
0083           22 IF(LOUT EQ 0)  GO TO 29
0084            DO 23 I = 1,NUMD
0085            IF(KOUT(I).EQ 0 OR DS(I) GE. 5)  GO TO 23
0086            LZ(I) = KNEX
0087           23 CONTINUE
0088            CALL PACK(LZ,NODE(1,1),NUMD,4)
      C
      C     ***   PICK COST TO CONSIDER NEXT   ***
0089           29 NCOST = 0
0090            NKEY = 0
0091            FMAX = -1 0E35
0092            IF (KPNX.NE.NX)
               1CALL UNPACK (LZ,NODE(1,NX),NUMD,4)
0093           30 DO 35 NIC = 1,NUMD
0094            IF(LZ(NIC) LT 15) GO TO 35
0095            NKEY = NKEY + 1
0096            IF (KPNX.EQ NX) GO TO 300
0097            WR(NIC) = 0.0
0098            DO 33 J= 1,NM
0099            IF(YRLM(J) EQ 0 0) GO TO 33


FORTRAN IV G LEVEL 1, MOD 4            CHOOZ            DATE = 70113        15/09/56

0100            CALL UNPACK(MZ,VNM(1,J),NV2,1)
0101            CMIN = 1 0E30
0102            KO = LYR(J)
0103            JX = LETT(J)
0104            ITR = LTR(JX)
0105            DO 32 I = 1,NV2
0106            IF(MZ(I) EQ 0)  GO TO 32
0107            DO 31 M = 1, 20
0108            IF(NONREC(I,M).EQ 0)  GO TO 315
0109            NO = NONREC(I,M)
0110            IF(NO EQ NIC)  GO TO 32
0111            IF(KI*LZ(NO) LT XO)  GO TO 32
0112           31 CONTINUE
0113          315 I1 = I + MINO(0,ISIGN(NV,NV-I))
0114            X = NMULT(I1,JX)
0115            CX = YRLM(J)*RECUR(I1,KO,ITR)*X
0116            IF(CX LT CMIN)  CMIN = CX
0117           32 CONTINUE
0118            WR(NIC) = WR(NIC) + CMIN
0119           33 CONTINUE
0120          300 PF = WR(NIC) - W(NX)
0121            IF(PF LT..001)  GO TO 35
0122          301 IF(SUST(NIC).GE..001) DF = DS(NIC)*0 5 + SUST(NIC) + PF
                1  -1 0E4/(SUST(NIC)**4)
0123            IF(SUST(NIC).LT  001)  DF = 0 5*DS(NIC) + 4.0 + PF
0124            IF(SUST(NIC).LT..001 AND.PF GT 1 0E10)  DF = 1.0E34
0125            IF(DF LE FMAX)  GO TO 35
0126            FMAX = DF
0127            NCOST = NIC
0128           35 CONTINUE
0129           36 IF(NCOST.NE.0)  GO TO 295
      C
0130            IF(NKEY.EQ.0)  GO TO 75
0131            DO 37 I = 1,NUMD
0132            IF(LZ(I).EQ 15)  LZ(I) = 0
0133           37 CONTINUE
0134            CALL PACK(LZ,NODE(1,NX),NUMD,4)
0135            IF(Z(NX).LT.GUESS)  GUESS = Z(NX)
0136            WRITE(6,210) GUESS
0137            GO TO 75
      C
      C     ALLOCATE SPACE FOR NEW NODES
      C
```

```
FORTRAN IV G LEVEL 1, MOD 4            CHOOZ              DATE = 70113         15/09/56

0138          295 IF(SUST(NCOST) GE  001)  J=0 + (NYD(NCOST) - 1)/KI
0139              IF(SUST(NCOST) LT .001)   J = KNEX - 1
0140              IF(NEXT EQ.1)  GO TO 41
0141              DO 40  I = 2,NEXT
0142              K = NEXT + 2 - I
0143              IF(Z(K) LE GUESS)  GO TO 40
0144              J=J+1
0145              IF(SUST(NCOST).GE .001)  NSAVE(J) = K
0146              IF(SUST(NCOST) LT .001)  NSAVE(1) = K
0147              IF(J EQ KNEX) GO TO 44
0148           40 CONTINUE
0149           41 IF(J EQ KNEX)  GO TO 44
0150              J=J+1
0151              NEXT=NEXT+1
0152              IF(NEXT.EQ 500) GO TO 74
0153              IF(SUST(NCOST) GE  001)  NSAVE(J) = NEXT
0154              IF(SUST(NCOST).LT .001)  NSAVE(1) = NEXT
0155              GO TO 41
                C
                C     ***   BRANCH WITH VARYING YEARS OF SUSTAINING COST   ***
0156           44 DO 52 K=1,10
0157              IF(SUST(NCOST) GE  001 AND K LT.1+(NYD(NCOST)-1)/KI)  GO TO 52
0158              KX=NSAVE(K)
0159              IF((K-1)*KJ.LT.LYD(NCOST)) GO TO 45
0160              W(KX) = 1 0E30
0161              Z(KX) =20.0E30
0162              GO TO 509
0163           45 DO 46 I=1,5
0164           46 NODE(I,KX)=NODE(I,NX)
0165              LZ(NCOST)=K-1
0166              IF(K EQ 1 + (NYD(NCOST)-1)/KI)   LZ(NCOST) = 0
0167              CALL PACK(LZ,NODE(1,KX),NUMD,4)
                C
0168              CALL LBOUND
                C
0169          509 IF(SUST(NCOST).LT  001) GO TO 53
0170              IF(KI*K GE MYRS) GO TO 53
0171           52 CONTINUE
                C     ***   BRANCH INCLUDING NCOST AND ALL SUSTAINING - PUT IN NODE NX   ***
0172           53 LZ(NCOST) = (LYD(NCOST) + KI - 1)/KI
0173              CALL PACK (LZ,NODE(1,NX),NUMD,4)
0174              IF(W(NX).GT.W(KX)- 0001 AND W2(NX).GT W2(KX)-0 0001)  K=50
0175              IF(W(NX) GT W(KX)-.0001 AND W2(NX).GT.1 0E25.AND.W2(KX)-W2(NX).LT.


FORTRAN IV G LEVEL 1, MOD 4            CHOOZ              DATE = 70113         15/09/56

                  1  1 0E25)  K=50
0176              KX = NX
0177              KZ = LYD(NCOST)
                C
0178              CALL LBOUND
                C
                C     ***   IMPROVE GUESS IF AT BOTTOM OF TREE   ***
0179              IF(NKEY.GT.1) GO TO 55
0180              GUESS1 = GUESS
0181              IF(Z(NX).LT.GUESS)  GUESS = Z(NX)
0182              DO 51 I=1,KNEX
0183              IF(SUST(NCOST) GE .001.AND I.LT.1+(NYD(NCOST)-1)/KI)  GO TO 51
0184              K=NSAVE(I)
0185              IF(Z(K) LT GUESS)  GUESS = Z(K)
0186              IF(SUST(NCOST).LT. 001) GO TO 515
0187           51 CONTINUE
0188          515 WRITE (6,210)  GUESS
                C
                C     PICK NEXT NODE FOR BRANCHING AS THE ONE WITH LEAST LOWER BOUND Z
0189           55 KPNX = NX
0190              NX = 1
0191              DO 59  I=2,NEXT
0192              IF(Z(NX).GT.Z(I))  NX = I
0193           ,59 CONTINUE
0194              IF(Z(NX).LE GUESS)  GO TO 29
0195              WRITE(6,202)
0196              GUESS = 0 0
0197              RETURN
0198           74 WRITE(6,203)
0199              RETURN
0200           75 DO 80 J=1,NM
0201              IF(YRLM(J) EQ.0 0) GO TO 80
0202              CALL UNPACK(NZ,VNM(1,J),NV2,1)
0203              CMIN=1 0E30
0204              KO = LYR(J)
0205              JX = LETT(J)
0206              ITR = LTR(JX)
0207              DO 78 I=1,NV2
0208              IF(NZ(I).EQ 0)  GO TO 78
0209              DO 77 K=1,20
0210              IF(NONREC(I,K) EQ.0) GO TO 775
0211              NO =NONREC(I,K)
0212              IF(KI*LZ(NO) LT.KO)     GO TO 78
```

```
FORTRAN IV G LEVEL 1, MOD 4            CHOOZ            DATE = 70113        15/09/56

0213           77 CONTINUE
0214          775 I1 = I + MINO(I0,ISIGN(NV,NV-I))
0215              X = NMULT(I1,JX)
0216              CX=YRLM(J)*RECUR(I1,KO,ITR)*X
0217              IF(CX GE CMIN) GO TO 78
0218              CMIN=CX
0219              MIN(J)=I
0220           78 CONTINUE
0221           80 CONTINUE
0222              IF(NP GT 0 OR LOUT GT.0)  GO TO 354
0223              DMIN = Z(NX) - W(NX)
0224              WRITE(6,201) NX,W(NX),DMIN,Z(NX)
0225              RETURN
                C
0226          354 CALL PADCST
0227              IF(Z(NX).LE.19.0E30)  GO TO 320
0228              GUESS = GUESS1
0229              WRITE (6,210) GUESS
0230          320 IF(EXTRA GE.99 9)  GO TO 55
0231          201 FORMAT (1HO,13(1H*),17H S O L U T I O N ,12(1H*)/1H ,I3,29X,
                 1  3(F9.2,5X))
0232          202 FORMAT(14H1GUESS TOO LOW)
0233          203 FORMAT (19H1EXCEEDED 500 NODES)
0234          205 FORMAT (1H1,55HB R A N C H   A N D   B O U N D   N O D E   V A L U
                 1 E S/58HONODE  BRANCHED  COST   YEARS    RECURRING   NON-RECURRING
                2,5X,5HTOTAL/13H  NO.    FROM,5X,12HNO.  SUSTAIN,5X,3(5HBOUND,9X)/)
0235          210 FORMAT (12H NEW GUESS =, F12.2)
0236           99 RETURN
0237              END

FORTRAN IV G LEVEL 1, MOD 4            CHOOZ            DATE = 70113        15/09/56

  TOTAL MEMORY REQUIREMENTS 00183E BYTES
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
        VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02CH(R)
IEW0461  IBCOM#
IEW0461  UNPACK
IEW0461  PACK
IEW0461  LBOUND
IEW0461  PADCST
IEW0461  MINO

                                        MODULE MAP

  CONTROL SECTION                    ENTRY

    NAME     ORIGIN  LENGTH          NAME  LOCATION    NAME  LOCATION    NAME  LOCATION    NAME  LOCATION

  CHOOZ          00    183E
  NONDIM       1840      1C
  VEHCLE       1860     4B0
  STAGE        2010     6E4
  MISSON       26F8     A90
  DSCOST       3188    1838
  PADS         49C0    378C
  VEHREC       8150    3CF4
  CAPMAT       BE48     FA4
  CHOOZE       CDF0      84
  SCRACH       CE78    53FC
  SAVEPT      12278     184
  SAVELB      12400      18

  ENTRY ADDRESS       00
  TOTAL LENGTH     12418

****MOX02CH   NOW REPLACED IN DATA SET

                                DIAGNOSTIC MESSAGE DIRECTORY

  IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED
```

```
(17)          OS/360  FORTRAN H                              DATE  70 113/15.10.35

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002            SUBROUTINE DECISN (MYFLAG)
                  C*     SET UP DECISION COST CATEGORIES AND THEIR AVAILABILITY
   ISN 0003            INTEGER*2 NONREC,LYD,NPAD,VEH,NPFAM,NPSTG,NPINTL,NPINTU
                  C STORAGE FOR MAIN, DECISN, STGNUM, CAPABL, AVAIL, AND CHOOZ
   ISN 0004            COMMON/NONDIM/ NUMD,NSTG,MYRS,NFAM,NCI,NP,NV
                  C STORAGE FOR MAIN, DECISN, STGNUM, CAPABL, AND AVAIL
   ISN 0005            COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
                  C STORAGE FOR MAIN, DECISN, STGNUM, AND AVAIL
   ISN 0006            COMMON/STAGE/ LSA(40),SNR(41),STG(40),SUS(40),NYS(40),SUSLS(40,2),
                      1  NFS(40,4)
                  C STORAGE FOR MAIN, DECISN, AVAIL, AND CHOOZ
   ISN 0007            COMMON/DSCOST/ DS(100),SUST(100),NYD(100),LYD(100),NONREC(120,20)
                  C STORAGE FOR MAIN, DECISN, AND STGNUM
   ISN 0008            COMMON/INT/ NFML(40),NFMU(40),DINT(40),SINT(40),SINTLS(40,2)
                  C STORAGE FOR MAIN AND DECISN
   ISN 0009            COMMON/FAMILY/ KODEF(30),FMNR(30),FMSUS(30),FMSLS(30,2),FAM(30)
   ISN 0010            COMMON/PADS/ PFAMD(30,5,3),PFAMS(30,5,3),NPFAM(30,5),
                      1  PSTGD(30,10,3),PSTGS(30,10,3),NPSTG(30,10),PINTS(30,5,3),
                      2  NPINTL(30,5),NPINTU(30,5),PAD(30)
                  C STORAGE FOR DECISN ONLY
   ISN 0011            COMMON/SCRACH/          MAT(100),LYF(30),NYF(30),MAF(30,3),
                      1  MATC(40,3),MAPF(30,5),MAPS(30,10),MAPI(30,5),DUMMY(4045),
                      2  MAS(40,3), VM(2,120)
                  C
                  C*     ***SET UP OS  COSTS FOR BRANCH AND BOUND PROCEDURE***
                  C      CALCULATE AVAILABILITY OF EACH DECISION COST
                  C
   ISN 0012            NUMD = 0
   ISN 0013            DO 3 I = 1,NSTG
   ISN 0014            LSA(I) = MINO(LSA(I),MYRS)
   ISN 0015            MAS(I,1) = 0
   ISN 0016            IF(SNR(I)&SUS(I).LT. 01) GO TO 9024
   ISN 0018            NUMD = NUMD + 1
   ISN 0019            DS (NUMD)=SNR(I)
   ISN 0020            SUST (NUMD)=SUS(I)
   ISN 0021            MAT(NUMD) = I
   ISN 0022            MAS(I,1) = NUMD
                  C  NYD = FIRST YEAR COMPONENTS DEPENDENT ON DEV COST NUMD ARE AVAILABLE
   ISN 0023            NYD(NUMD) = NYS(I)
                  C  LYD = LAST YEAR COMPONENTS DEPENDENT ON DEV COST NUMD ARE AVAILABLE
   ISN 0024            LYD(NUMD) = LSA(I)
   ISN 0025     9024 DO 9025 J = 1,2
```

```
ISN 0117          MAPF(I,J) = NUMD
ISN 0118          NYD(NUMD) = MYRS
ISN 0119          LYD(NUMD) = MYRS
ISN 0120       9030 CONTINUE
ISN 0121          DO 9031 J = 1,10
ISN 0122          MAPS(I,J) = 0
ISN 0123          IF (PSTGD(I,J,1) & PSTGS(I,J,1).LT .01 ) GO TO 9031
ISN 0125          NUMD = NUMD & 1
ISN 0126          DS(NUMD) = PSTGD(I,J,1)
ISN 0127          SUST(NUMD) = PSTGS(I,J,1)
ISN 0128          MAT(NUMD) = -300 - I
ISN 0129          MAPS(I,J) = NUMD
ISN 0130          NYD(NUMD) = MYRS
ISN 0131          LYD(NUMD) = MYRS
ISN 0132       9031 CONTINUE
ISN 0133          DO 9032 J = 1,5
ISN 0134          MAPI(I,J) = 0
ISN 0135          IF (PINTS(I,J,1)  LT. .01) GO TO 9032
ISN 0137          NUMD = NUMD & 1
ISN 0138          DS(NUMD) = 0.0
ISN 0139          SUST(NUMD) = PINTS(I,J,1)
ISN 0140          MAT(NUMD) = -400 - I
ISN 0141          MAPI(I,J) = NUMD
ISN 0142          NYD(NUMD) = MYRS
ISN 0143          LYD(NUMD) = MYRS
ISN 0144       9032 CONTINUE
ISN 0145       9011 CONTINUE
               C     COMPUTE AVAILABILITY OF PAD COSTS
ISN 0146          DO 9033 I = 1,NV
ISN 0147          DO 9033 J = 1,2
ISN 0148          IF (NPAD(J,I).EQ 0) GO TO 9033
ISN 0150          N = NPAD(J,I)
ISN 0151          J1 = J & 1
ISN 0152          DO 9034 K = 1,4
ISN 0153          IF (VEH(K,I) EQ.0) GO TO 9033
ISN 0155          L = VEH(K,I)
ISN 0156          IF (MAS(L,J1) EQ 0) GO TO 9035
ISN 0158          M = MAS(L,J1)
ISN 0159          NYD(M) = MINO(NYD(M),NYP(J,I))
ISN 0160       9035 DO 9036 I1 = 1,10
ISN 0161          IF (NPSTG(N,I1) NE.L) GO TO 9036
ISN 0163          IF (MAPS(N,I1) EQ 0) GO TO 9037

ISN 0165          M = MAPS(N,I1)
ISN 0166          NYD(M) = MINO(NYD(M),NYP(J,I))
ISN 0167          GO TO 9037
ISN 0168       9036 CONTINUE
ISN 0169       9037 DO 9038 I1 = 1,4
ISN 0170          IF (NFS(L,I1) EQ.0) GO TO 9038
ISN 0172          KX = NFS(L,I1)
ISN 0173          IF (MAF(KX,J1) EQ.0) GO TO 9039
ISN 0175          M = MAF(KX,J1)
ISN 0176          NYD(M) = MINO(NYD(M),NYP(J,I))
ISN 0177       9039 DO 9040 I2 = 1,5
ISN 0178          IF (NPFAM(N,I2) NE KX) GO TO 9040
ISN 0180          IF (MAPF(N,I2).EQ 0) GO TO 9041
ISN 0182          M = MAPF(N,I2)
ISN 0183          NYD(M) = MINO(NYD(M),NYP(J,I))
ISN 0184          GO TO 9041
ISN 0185       9040 CONTINUE
ISN 0186       9041 IF (K EQ.4) GO TO 9034
ISN 0188          IF (VEH(K&1,I) EQ 0) GO TO 9034
ISN 0190          L1 = VEH(K&1,I)
ISN 0191          IF (NC1.EQ 0) GO TO 9047
ISN 0193          DO 9045 IX = 1,NC1
ISN 0194          IF (NFML(IX).NE KX) GO TO 9045
ISN 0196          DO 9046 IY = 1,4
ISN 0197          IF (NFMU(IX) EQ NFS(L1,IY))GO TO 9048
ISN 0199       9046 CONTINUE
ISN 0200          GO TO 9045
ISN 0201       9048 IF (MAIC(IX,J1).EQ 0) GO TO 9045
ISN 0203          M = MAIC(IX,J1)
ISN 0204          NYD(M) = MINO(NYD(M),NYP(J,I))
ISN 0205       9045 CONTINUE
ISN 0206       9047 DO 9042 I3 = 1,5
ISN 0207          IF (NPINTL(N,I3).NE KX) GO TO 9042
ISN 0209          DO 9043 I4 = 1,4
ISN 0210          IF (NPINTU(N,I3).EQ NFS(L1,I4)) GO TO 9044
ISN 0212       9043 CONTINUE
ISN 0213          GO TO 9042
ISN 0214       9044 IF(MAPI(N,I3) EQ 0) GO TO 9042
ISN 0216          M = MAPI(N,I3)
ISN 0217          NYD(M) = MINO(NYD(M),NYP(J,I))
ISN 0218       9042 CONTINUE
ISN 0219       9038 CONTINUE
```

```
ISN 0026          MAS(I,JG1) = 0
ISN 0027          IF (SUSLS(I,J).LT..01) GO TO 9025
ISN 0029          NUMD = NUMD & 1
ISN 0030          DS(NUMD) = 0 0
ISN 0031          SUST(NUMD) = SUSLS(1,J)
ISN 0032          MAT(NUMD) = I
ISN 0033          MAS(I,JG1) = NUMD
ISN 0034          NYD(NUMD) = MYRS
ISN 0035          LYD(NUMD) = LSA(I)
ISN 0036       9025 CONTINUE
ISN 0037          3 CONTINUE
ISN 0038          IF(NFAM EQ  0) GO TO 601
                C     CALCULATE FAMILY AVAILABILITY DATE
                C FIRST YR. FAMILY IS AVAIL. = 1ST YR. ANY STAGE IN THAT FAMILY IS AVAIL
ISN 0040          DO 422 II = 1,NFAM
ISN 0041          I = KODEF(II)
ISN 0042          LYF(I) = 0
ISN 0043       422 NYF(I) = MYRS
ISN 0044          DO 423 J = 1,NSTG
ISN 0045          DO 424 MS = 1,4
ISN 0046          I = NFS(J,MS)
ISN 0047          IF(I.EQ 0)  GO TO 423
ISN 0049          NYF(I) = MIND (NYF(I),NYS(J))
ISN 0050          LYF(I) = MAXO(LYF(I),LSA(J))
ISN 0051       424 CONTINUE
ISN 0052       423 CONTINUE
ISN 0053          DO 6 II = 1,NFAM
ISN 0054          I = KODEF(II)
ISN 0055          MAF(I,1) = 0
ISN 0056          IF(FMNR(I)&FMSUS(I).LT .01) GO TO 9026
ISN 0058          NUMD = NUMD & 1
ISN 0059          DS (NUMD) = FMNR(I)
ISN 0060          SUST (NUMD)=FMSUS(I)
ISN 0061          MAT(NUMD) = -I
ISN 0062          MAF(I,1) = NUMD
ISN 0063          NYD(NUMD) = NYF(I)
ISN 0064          LYD(NUMD) = LYF(I)
ISN 0065       9026 DO 9027 J = 1,2
ISN 0066          MAF(I,JG1) = 0
ISN 0067          IF (FMSLS(I,J) LT. 01) GO TO 9027
ISN 0069          NUMD = NUMD & 1
ISN 0070          DS(NUMD) = 0 0
```

```
ISN 0071          SUST(NUMD) = FMSLS(I,J)
ISN 0072          MAT(NUMD) = -I
ISN 0073          MAF(I,JG1) = NUMD
ISN 0074          NYD(NUMD) = MYRS
ISN 0075          LYD(NUMD) = LYF(I)
ISN 0076       9027 CONTINUE
ISN 0077          6 CONTINUE
ISN 0078        601 IF(NC1 EQ.0) GO TO 61
ISN 0080          DO 60 I = 1,NC1
ISN 0081          JF = NFML(I)
ISN 0082          KF = NFMU(I)
ISN 0083          MAIC(I,1) = 0
ISN 0084          IF(DINT(I)&SINT(I).LT. 01) GO TO 9028
ISN 0086          NUMD = NUMD & 1
ISN 0087          DS (NUMD) = DINT(I)
ISN 0088          SUST (NUMD)=SINT(I)
ISN 0089          MAT(NUMD) = -100 -I
ISN 0090          MAIC(I,1) = NUMD
                C FIRST YR. INT. COST IS AVAIL. = 1ST YR  BOTH FAMS  ARE AVAIL
ISN 0091          NYD(NUMD) = MAXO(NYF(JF),NYF(KF))
ISN 0092          LYD(NUMD) = MINO(LYF(JF),LYF(KF))
ISN 0093       9028 DO 9029 J = 1,2
ISN 0094          MAIC(I,JG1) = 0
ISN 0095          IF (SINTLS(I,J).LT. 01) GO TO 9029
ISN 0097          NUMD = NUMD & 1
ISN 0098          DS(NUMD) = 0.0
ISN 0099          SUST(NUMD) = SINTLS(I,J)
ISN 0100          MAT(NUMD) = -100 - I
ISN 0101          MAIC(I,JG1) = NUMD
ISN 0102          NYD(NUMD) = MYRS
ISN 0103          LYD(NUMD) = MINO(LYF(JF),LYF(KF))
ISN 0104       9029 CONTINUE
ISN 0105          60 CONTINUE
ISN 0106          61 IF (NP EQ.0) GO TO 9010
ISN 0108          DO 9011 I = 1,NP
ISN 0109          DO 9030 J = 1,5
ISN 0110          MAPF(I,J) = 0
ISN 0111          IF(PFAMD(I,J,1) & PFAMS(I,J,1)  LT  .01) GO TO 9030
ISN 0113          NUMD = NUMD & 1
ISN 0114          DS(NUMD) = PFAMD(I,J,1)
ISN 0115          SUST(NUMD) = PFAMS(I,J,1)
ISN 0116          MAT(NUMD) = -200 - I
```

```
ISN 0220      9034 CONTINUE
ISN 0221      9033 CONTINUE
              C
ISN 0222      9010 CALL MATCH(MYFLAG)
ISN 0223           IF(MYFLAG EQ.1)  RETURN
              C
ISN 0225           CALL PRINT
              C
ISN 0226           MYFLAG = 0
ISN 0227           RETURN
ISN 0228           END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000      NAME H0X02DC(R)
IEW0461  MATCH
IEW0461  PRINT
```

CROSS REFERENCE TABLE

| CONTROL SECTION NAME | ORIGIN | LENGTH |
|---|---|---|
| DECISN | 00 | 1198 |
| NONDIM | 1198 | 1C |
| VEHCLE | 11B8 | 4B0 |
| STAGE | 1668 | 6E4 |
| DSCOST | 1D50 | 1838 |
| INT | 3588 | 3C0 |
| FAMILY | 3948 | 2D0 |
| PADS | 3C18 | 378C |
| SCRACH | 73A8 | 53FC |

ENTRY

| NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|

| LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION | LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION |
|---|---|---|---|---|---|
| 110 | NONDIM | NONDIM | 114 | VEHCLE | VEHCLE |
| 118 | STAGE | STAGE | 11C | DSCOST | DSCOST |
| 120 | INT | INT | 124 | FAMILY | FAMILY |
| 128 | PADS | PADS | 12C | PADS | PADS |
| 130 | PADS | PADS | 134 | SCRACH | SCRACH |
| 138 | SCRACH | SCRACH | 13C | MATCH | $UNRESOLVED |
| 140 | PRINT | $UNRESOLVED | | | |

ENTRY ADDRESS 00
TOTAL LENGTH C7A8

```
****H0X02DC   NOW REPLACED IN DATA SET
```

DIAGNOSTIC MESSAGE DIRECTORY

```
IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED
```

```
(17)          OS/360  FORTRAN H                          DATE  70 113/15 12 04

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002          SUBROUTINE LBOUND
                  C CALCULATE LOWER BOUND PENALTY FUNCTION
   ISN 0003          REAL NPERPD
   ISN 0004          INTEGER*2 LETT,LYR,MIN,NONREC,LYD,NSAVE,KEEP,NMULT
                  C STORAGE FOR CHOOZ AND LBOUND
   ISN 0005          COMMON/SAVELB/ KX,KI,NX,NCOST,K,KZ
   ISN 0006          COMMON/NONDIM/ NUMD,NSTG,HYRS,NFAM,NCI,NP,NV
   ISN 0007          COMMON/MISSON/ LETT(250),LYR(250),LTR(50),YRLM(250),MIN(250),NM
   ISN 0008          COMMON/DSCOST/ DS(100),SUST(100),NYD(100),LYD(100),NONREC(120,20)
                  C    STORAGE FOR MAIN, STGNUM, CHOOZ, AND CAPABL
   ISN 0009          COMMON/VEHREC/ RECUR(60,20,2),NMULT(60,50),IFLAG
   ISN 0010          COMMON/CAPMAT/ VNM(4,250),NV2
   ISN 0011          COMMON/CHOOZE/ NEXT,GUESS,NPERPD(30),LP
   ISN 0012          COMMON/SCRACH/ TDS(500),W2(500),Z(500),W(500),NSAVE(10),
                    1  NODE(5,500),LZ(100),MZ(120),KEEP(100),COST(2,250),WR(100)
                  C
   ISN 0013          IF (K EQ 50) GO TO 54
                  C     ***FIND NEW RECURRING LOWER BOUND***
   ISN 0015        49 W(KX)=0.
   ISN 0016          W2(KX) = 0.0
   ISN 0017          DO 50 J=1,NM
   ISN 0018          IF(YRLM(J) EQ 0.0) GO TO 50
   ISN 0020          CALL UNPACK(MZ,VNM(1,J),NV2,1)
   ISN 0021          COST(1,J) = 1.0E30
   ISN 0022          COST(2,J) = 1.0E30
   ISN 0023          KO = LYR(J)
   ISN 0024          JX = LETT(J)
   ISN 0025          ITR = LTR(JX)
   ISN 0026          DO 48 I=1,NV2
   ISN 0027          IF(MZ(I) EQ.0)  GO TO 48
   ISN 0029          DO 47 M=1,20
   ISN 0030          IF(NONREC(I,M) EQ 0) GO TO 475
   ISN 0032          NO = NONREC(I,M)
   ISN 0033          IF(KI*LZ(NO) LT   KO  ) GO TO 48
   ISN 0035        47 CONTINUE
   ISN 0036       475 I1 = I * MIN0(0,ISIGN(NV,NV-I))
   ISN 0037          X = NMULT(I1,JX)
   ISN 0038          CX=YRLM(J)*RECUR(I1,KO,ITR)*X
   ISN 0039          IF(CX .GE. COST(2,J))  GO TO 48
   ISN 0041          IF(CX.LT.COST(1,J))  GO TO 43
   ISN 0043          COST(2,J) = CX
   ISN 0044          GO TO 48

   ISN 0045        43 COST(2,J) = COST(1,J)
   ISN 0046          COST(1,J) = CX
   ISN 0047          MIN(J) = I
   ISN 0048        48 CONTINUE
   ISN 0049          W(KX)=W(KX)&COST(1,J)
   ISN 0050          W2(KX) = W2(KX) & COST(2,J)
   ISN 0051        50 CONTINUE
   ISN 0052          IF(KX EQ NX)  GO TO 510
   ISN 0054          KZ = KI*LZ(NCOST)
   ISN 0055          IF(W(KX).LT.1.0E20)  GO TO 508
   ISN 0057          TGO = 0.0
   ISN 0058          GO TO 38
   ISN 0059       508 IF(KZ EQ 0)  GO TO 510
   ISN 0061          KY = NSAVE(K-1)
   ISN 0062       512 IF(W(KX) GT W(KY)- C001.AND.W2(KX) GT.W2(KY)- 0001)  GO TO 38
   ISN 0064          IF(W(KX) GT.W(KY)- 0001 AND W2(KX) GT 1.0E25.AND W2(KY)-W2(KX) LT.
                    1  1.0E25)  GO TO 38
                  C
                  C     CALCULATE LOWER BOUND USING PENALTY FUNCTION BASED ON VEHICLES
   ISN 0066       510 DO 350 NIC = 1,NUMD
   ISN 0067          KEEP(NIC) = 1
   ISN 0068          IF(LZ(NIC) LT.15)  KEEP(NIC) = 0
   ISN 0070       350 CONTINUE
   ISN 0071       355 TGO = 0.0
   ISN 0072          IV = 0
   ISN 0073          TG = 0.0
   ISN 0074       354 DO 351 IX = 1,NV2
   ISN 0075          IF(IX.EQ.IV)  GO TO 351
   ISN 0077          VGO = 0.0
   ISN 0078       330 DO 90 J = 1,NM
   ISN 0079          IF(YRLM(J) EQ 0.0) GO TO 90
   ISN 0081          IF(MIN(J) EQ IX)  GO TO 91
   ISN 0083        90 CONTINUE
   ISN 0084          GO TO 351
   ISN 0085        91 PF = 0.0
   ISN 0086          KTV = 0
   ISN 0087          DO 341 M = 1,20
   ISN 0088          IF(NONREC(IX,M).EQ 0)  GO TO 3415
   ISN 0090          NO = NONREC(IX,M)
   ISN 0091          IF(KEEP(NO) EQ.0)  GO TO 341
   ISN 0093          VGO = VGO &DS(NO) & FLOAT(KI)*SUST(NO)
   ISN 0094          KTV = 1
```

```
ISN 0095       341 CONTINUE
ISN 0096      3415 IF(KTV EQ 0)  GO TO 351
ISN 0098           DO 331J= 1,NM
ISN 0099           IF(YRLM(J) LT 0 001 OR MIN(J) NE.IX)  GO TO 331
ISN 0101           PF = PF & COST(2,J) - COST(1,J)
ISN 0102       331 CONTINUE
ISN 0103           VGO = AMIN1(VGO,PF)
ISN 0104           IF(VGO LT.TG)  GO TO 351
ISN 0106           IV = IX
ISN 0107           TG = VGO
ISN 0108       351 CONTINUE
ISN 0109           IF(TG.LT GUESS* 01)  GO TO 37
ISN 0111           TGO = TG & TGO
ISN 0112           TG = 0 0
ISN 0113           DO 352 M = 1,20
ISN 0114           IF(NONREC(IV,M).EQ.0)  GO TO 353
ISN 0116           NO = NONREC(IV,M)
ISN 0117           KEEP(NO) = 0
ISN 0118       352 CONTINUE
ISN 0119       353 GO TO 354
ISN 0120        37 TGO = TGO & TG
ISN 0121        38 IF(KZ.EQ.0)  TDS(KX) = TDS(NX)
ISN 0123           IF(KZ GT 0.AND KX.NE NX) TDS(KX) = TDS(NX)
                  1          & DS(NCOST) & FLOAT((K-1)*KI-NYD(NCOST)&1)*SUST(NCOST)
ISN 0125        54 IF (KX.EQ NX) TDS(NX) = TDS(NX) &
                  1          DS(NCOST) & FLOAT(LYD(NCOST)-NYD(NCOST)&1)*SUST(NCOST)
ISN 0127           DMIN = TGO & TDS(KX)
                C
ISN 0128       507 Z(KX) = DMIN & W(KX)
ISN 0129           IF(LP.GT 0)
                  1WRITE(6,204) KX,NX,NCOST,KZ,W(KX),DMIN, Z(KX)
ISN 0131           RETURN
ISN 0132       204 FORMAT (1H ,4(I3,5X),3(F9 2,5X))
ISN 0133           END

****** END OF COMPILATION ******
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
           VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02LB(R)
IEW0461  IBCOM#
IEW0461  UNPACK
```

CROSS REFERENCE TABLE

| CONTROL SECTION NAME | ORIGIN | LENGTH |
|---|---|---|
| LBOUND | 00 | A4C |
| SAVELB | A50 | 18 |
| NONDIM | A68 | 1C |
| MISSON | A88 | A90 |
| DSCOST | 1518 | 1838 |
| VEHREC | 2D50 | 3CF4 |
| CAPMAT | 6A48 | FA4 |
| CHOOZE | 79F0 | 84 |
| SCRACH | 7A78 | 53FC |

ENTRY

| NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|

| LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION | LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION |
|---|---|---|---|---|---|
| 150 | SAVELB | SAVELB | 154 | NONDIM | NONDIM |
| 158 | MISSON | MISSON | 15C | DSCOST | DSCOST |
| 160 | VEHREC | VEHREC | 164 | VEHREC | VEHREC |
| 168 | VEHREC | VEHREC | 16C | CAPMAT | CAPMAT |
| 170 | CHOOZE | CHOOZE | 174 | SCRACH | SCRACH |
| 178 | SCRACH | SCRACH | 17C | SCRACH | SCRACH |
| 180 | IBCOM# | $UNRESOLVED | 184 | UNPACK | $UNRESOLVED |
| 8C | SCRACH | SCRACH | 94 | CAPMAT | CAPMAT |

ENTRY ADDRESS 00
TOTAL LENGTH CE78

***MOX02LB  NOW REPLACED IN DATA SET

DIAGNOSTIC MESSAGE DIRECTORY

```
FORTRAN IV G LEVEL 1, MOD 4            MAIN             DATE = 70113        15/13/01

      C     THIS PROGRAM GENERATES THE LEAST COST ASSIGNMENT OF LAUNCH
      C     VEHICLES TO SPACE MISSIONS   A BRANCH AND BOUND TECHNIQUE IS USED
      C     TO REDUCE THE COMBINATORIAL COMPLEXITY OF THE PROBLEM  SEVERAL BRANCHES
      C     ARE CREATED AT EACH NODE   ONE OF THE BRANCHES EXCLUDES THE NEXT
      C     COST AND THE OTHERS    ASSUME EXPENDITURE OF A NON-RECURRING COST
      C     WITH 1-2 YEARS OF SUSTAINING COST ADDED AT EACH NODE
      C     PENALTY FUNCTIONS ARE USED TO SHARPEN THE LOWER BOUND
      C     MODIFIED PENALTY FUNCTION IN LOWER BOUND BASED ON VEHICLE INFORMATION
      C ****THIS VERSION USES RATE EFFECTS IN RECURRING COSTS****
      C*****THIS VERSION INCORPORATES PAD COSTS*****
      C
      C
0001        DOUBLE PRECISION  HISNAM
0002        REAL NPERPD
0003        INTEGER*2 LETT,LYR,MIN,NONREC,LYD,NPAD,VEH,NPFAM,NPSTG,NPINTL,
           1  NPINTU,NMULT
      C STORAGE FOR MAIN, DECISN, STGNUM, CAPABL, AVAIL, AND CHOOZ
0004        COMMON/NONDIM/ NUMD,NSTG,MYRS,NFAM,NCI,NP,NV
0005        COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
      C STORAGE FOR MAIN, STGNUM, CAPABL, AVAIL, AND CHOOZ
0006        COMMON/MISSON/ LETT(250),LYR(250),LTR(50),YRLH(250),MIN(250),NM
      C STORAGE FOR MAIN, DECISN, STGNUM, AVAIL, AND CHOOZ
0007        COMMON/STAGE/ LSA(40),SNR(41),STG(40),SUS(40),NYS(40),SUSLS(40,2),
           1  NFS(40,4)
      C STORAGE FOR MAIN, DECISN, AVAIL, AND CHOOZ
0008        COMMON/DSCOST/ DS(100),SUST(100),NYD(100),LYD(100),NONREC(120,20)
      C STORAGE FOR MAIN, DECISN, AND STGNUM
0009        COMMON/INT/ NFML(40),NFMU(40),DINT(40),SINT(40),SINTLS(40,2)
      C STORAGE FOR MAIN, DECISN, AND CHOOZ
0010        COMMON/PADS/ PFAMD(30,5,3),PFAMS(30,5,3),NPFAM(30,5),
           1  PSTGD(30,10,3),PSTGS(30,10,3),NPSTG(30,10),PINTS(30,5,3),
           2  NPINTL(30,5),NPINTU(30,5),PAD(30)
      C     STORAGE FOR MAIN, STGNUM, CHOOZ, AND CAPABL
0011        COMMON/VEHREC/ RECUR(60,20,2),NMULT(60,50),IFLAG
      C     STORAGE FOR MAIN AND STGNUM AND REUSE
0012        COMMON/REUSBL/ NU(50),UPP(50)
      C STORAGE FOR MAIN, AVAIL, AND CHOOZ
0013        COMMON/CAPMAT/ VNM(4,250),NV2
      C STORAGE FOR MAIN AND DECISN
0014        COMMON/FAMILY/ KODEF(30),FMNR(30),FMSUS(30),FMSLS(30,2),FAM(30)
      C STORAGE FOR MAIN AND STGNUM
0015        COMMON/INTREC/ RINT(40),PLCINT(40)
0016        COMMON/STGREC/ NBY(40),MODE(40,3),SR(40,3),PLC(40,3),POJ(3),
```

```
FORTRAN IV G LEVEL 1, MOD 4            MAIN             DATE = 70113        15/13/01

           1  SRJ(3,3)
      C STORAGE FOR MAIN AND CAPABL
0017        COMMON/MISREQ/ IVEHA(50),VLR(50),WPR(50),NPLS(50),MR(50),NRR(50),
           1   RPLM(50),NTRIP(50)
0018        COMMON/VEHIN/ IV,KNV,IS,NMIS,NOPT,KODE(40),KODEP(30),IP,RPLO(50)
      C STORAGE FOR MAIN AND CHOOZ
0019        COMMON/CHOOZE/ NEXT,GUESS,NPERPD(30),LP
      C  STORAGE FOR MAIN AND OUTPUT
0020        COMMON/SAVEOP/MISNAM(50),IBY
      C STORAGE FOR MAIN ONLY
0021        COMMON/SCRACH/ MISN(50,20),KVEH(50),PB(50),MZ(50),KODEM(50),
           1 LZ(20),NPSTX(30,10),NPINXL(30,5),NPINXU(30,5),NPFAX(30,5),
           1  PLR(50),PLS(50),PLD(50),PLMD(50),DUMMY(2965),VM(2,120)
      C
0022        DATA BLANK/1H /
0023        SNR(41) = BLANK
      C
0024       11 NSTG = 0
0025          NFAM = 0
0026          NCI = 0
0027          NP = 0
0028          NMIS = 0
0029          NV=0
0030          IFLAG = 0
0031          READ(5,100) NOPT,MYRS,IBY,GUESS,IP,IS,IFM,II,IM,IV,LP
0032          IF (MYRS.EQ 0) GO TO 806
0033          WRITE (6,104)
0034          IF(IS LT 0)  GO TO 12
0035          WRITE(6,213)
0036          LX = 0
0037          DO 8000 I = 1,41
0038          READ(5,101)  KODE(I),STG(I),(SR(I,J),J=1,3),(PLC(I,J),J=1,3),
           1  SNR(I),SUS(I),NYS(I),LSA(I),NBY(I),(NFS(I,J),J=1,4),
           2  (MODE(I,J), J = 1,3)
0039          IF(KODE(I) EQ 0)  GO TO 12
      C  INPUT NU(I) = -2 IF WANT PROGRAM TO CALCULATE ESTIMATE FOR NU
0040          READ (5,9021) (SUSLS(I,J),J=1,2),NU(I),UPP(I),RPLO(I)
0041          NSTG = NSTG + 1
0042          WRITE (6,8001) STG(I),(SR(I,J),PLC(I,J),J=1,3),SNR(I),SUS(I),
           1  NYS(I),LSA(I),(NFS(I,J),J=1,4),NBY(I)
0043          DO 8002 J = 1,3
0044          IF (MODE(I,J).EQ.0) GO TO 8002
0045          LX = LX + 1
```

```
FORTRAN IV G LEVEL 1, MOD 4            MAIN              DATE = 70113        15/13/01

 0046              MODE(I,J) = LX
 0047              READ(5,8003) (SRJ(LX,K), K = 1,3), POJ(LX)
 0048              WRITE(6,8004) J,POJ(LX),SRJ(LX,1),POJ(LX),(SRJ(LX,K),K=2,3)
 0049         8002 CONTINUE
 0050              IF(RPLO(I) GT  001) WRITE(6,8005)  UPP(I)
 0051         8000 CONTINUE
 0052           12 IF(IFM LT 0)  GO TO 14
 0053              WRITE(6,214)
 0054              DO 13 J = 1,30
 0055              READ(5,102) I,FAM(I),FMNR(I),FMSUS(I),(FMSLS(I,K),K=1,2)
 0056              IF(I EQ 0)  GO TO 14
 0057              NFAM = NFAM + 1
 0058              KODEF(J) = I
 0059           13 WRITE(6,2141)KODEF(J), FAM(I),FMNR(I),FMSUS(I)
 0060           14 IF(II LT 0)  GO TO 1716
 0061              WRITE(6,215)
 0062              DO 1715 I = 1,40
 0063              READ(5,103)  NFML(I),NFMU(I),RINT(I),PLCINT(I),DINT(I),SINT(I),
                  1  (SINTLS(I,J),J=1,2)
 0064              IF(NFML(I) EQ.0)  GO TO 1716
 0065              NCI = NCI + 1
 0066              J=NFML(I)
 0067              K=NFMU(I)
 0068         1715 WRITE(6,216) FAM(J),FAM(K),RINT(I),PLCINT(I),DINT(I),SINT(I)
 0069         1716 IF (IP LT.0) GO TO 9002
 0070              WRITE (6,9003)
 0071              DO 9004 I = 1,30
 0072              READ (5,9005) KODEP(I),PAD(I),NPERPD(I)
 0073              IF (KODEP(I).EQ 0) GO TO 9002
 0074              WRITE (6,9006) KODEP(I),PAD(I),NPERPD(I)
 0075              NP = NP + 1
 0076              READ(5,5000)(NPSTX(I,J),(PSTGD(I,J,K),PSTGS(I,J,K),K=1,3),J=1,10)
 0077              DO 700 J = 1,10
 0078          700 NPSTG(I,J) = NPSTX(I,J)
 0079              READ(5,5000)(NPFAX(I,J),(PFAMD(I,J,K),PFAMS(I,J,K),K=1,3),J=1,5)
 0080              READ(5,5002)(NPINXL(I,J),NPINXU(I,J),(PINTS(I,J,K),K=1,3),J=1,5)
 0081              DO 701 J = 1,5
 0082              NPFAM(I,J) = NPFAX(I,J)
 0083              NPINTL(I,J) = NPINXL(I,J)
 0084          701 NPINTU(I,J) = NPINXU(I,J)
 0085              DO 9022 J = 1,10
 0086              IF (NPSTG(I,J).EQ 0) GO TO 5009
 0087              DO 9023 L = 1,NSTG


FORTRAN IV G LEVEL 1, MOD 4            MAIN              DATE = 70113        15/13/01

 0088              IF (NPSTG(I,J).NE KODE(L)) GO TO 9023
 0089              NPSTG(I,J) = L
 0090              WRITE(6,5003) STG(L),(PSTGD(I,J,K),PSTGS(I,J,K),K=1,3)
 0091              GO TO 9022
 0092         9023 CONTINUE
 0093         9022 CONTINUE
 0094         5009 DO 5006 J = 1,5
 0095              IF (NPFAM(I,J) EQ 01 GO TO 5007
 0096              L = NPFAM(I,J)
 0097         5006 WRITE(6,5004) FAM(L),(PFAMD(I,J,K),PFAMS(I,J,K),K=1,3)
 0098         5007 DO 5008 J = 1,5
 0099              IF (NPINTL(I,J) EQ.0) GO TO 9004
 0100              L = NPINTL(I,J)
 0101              LX = NPINTU(I,J)
 0102         5008 WRITE(6,5005) FAM(L),FAM(LX),(PINTS(I,J,K),K = 1,3)
 0103         9004 CONTINUE
 0104         9002 CONTINUE
 0105              IF(IM LT 0)  GO TO 20
 0106              DO 1719 I=1,MYRS
 0107         1719 LZ(I)=IBY+I-1
 0108              WRITE(6,217) (LZ(I),I=1,MYRS)
 0109         1717 DO 1718 I=1,50
 0110              READ(5,105) KODEM(I),MISNAM(I),PB(I),VLR(I),RPLM(I),WPR(I),
                  1  NTRIP(I), (MISN(I,J),J=1,MYRS)
 0111              IF(KODEM(I).EQ 0)  GO TO 20
 0112              NMIS = NMIS + 1
 0113              READ (5,107) PLR(I),PLS(I),PLO(I),
                  1  PLMO(I),NPLS(I),HR(I),LTR(I),NRR(I),IVEHA(I)
 0114         1718 WRITE(6,219) I,MISNAM(I),VLR(I),WPR(I),PB(I),LTR(I),
                  1  (MISN(I,J),J=1,MYRS)
 0115           20 IF(IS LT.0)  NSTG = KNSTG
 0116              IF(IFM LT.0)  NFAM = KNFAM
 0117              IF(II LT 0)  NCI = KNCI
 0118              IF (IP LT 0) NP=KNP
 0119              IF(IM LT.0)  NMIS = KNMIS
 0120              IF(IM.LT 0)  GO TO 3000
                   ***SET UP MISSION MATRIX BY YEAR***
 0121              NM = 0
 0122              DO 4 I = 1,NMIS
 0123              DO 4 J=1,MYRS
 0124              IF(MISN(I,J).EQ 0) GO TO 4
 0125              NM = NM + 1
 0126              YRLM(NM)= FLOAT(MISN(I,J)) * PB(I)
```

```
FORTRAN IV G LEVEL 1, MOD 4           MAIN              DATE = 70113        15/13/01

0127              LETT(NM)= I
0128              LYR(NM) = J
0129            4 CONTINUE
                C
0130            3000 CALL CAPABL
                C
0131              16 WRITE(6,2001) NSTG,NV,NFAM,NCI,NP,NMIS,MYRS,IBY,GUESS,NOPT
0132                 NV2 = 2 * NV
                C
0133                 CALL DECISN (MYFLAG)
                C
0134                 IF (MYFLAG EQ.1) GO TO 1
                C
0135                 IF(GUESS GT.1.0)  GO TO 3005
0136                 GO TO 3004
                C    THE FOLLOWING SECTION WILL BE USED WHEN SUBROUTINE COUNT IS AVAILABLE
                C      CALCULATE GUESS FROM INPUT VEHICLE/MISSION SELECTION
                C      READ(5,4020) (KVEH(I),I=1,NMIS)
                C      WRITE(6,4021)
                C      DO 30 I = 1,NMIS
                C      K1 = KVEH(I)
                C      I1 = VEH(1,K1)
                C      I2 = VEH(2,K1)
                C      I3 = VEH(3,K1)
                C      I4 = VEH(4,K1)
                C      WRITE(6,4022) MISNAM(I),STG(I1),STG(I2),STG(I3),STG(I4)
                C   30 CONTINUE
                C      DO 31 J = 1,NM
                C      I = LETT(J)
                C      MIN(J) = KVEH(I)
                C   31 CONTINUE
                C      CALL COUNT
                C    THE FOLLOWING CARD SHOULD BE REPLACED
0137             3004 TOT = 1.0E10
0138                 GUESS = TOT +  01
                C
0139             3005 CALL AVAIL
                C
                C      CALCULATE EXPONENT FOR LEARNING CURVE
0140             8020 ALOG2 = ALOG(2.)
0141                 IF (IS LT.0) GO TO 8030
0142                 DO 660 I=1,NSTG
0143                 DO 660 J=1,3
```

```
FORTRAN IV G LEVEL 1, MOD 4           MAIN              DATE = 70113        15/13/01

0144                 IF (MODE(I,J) NE.0) GO TO 660
0145                 PLC(I,J) = ALOG(PLC(I,J))/ALOG2
0146               660 CONTINUE
0147              8030 IF(II LT.0 OR NCI EQ 0)  GO TO 8031
0148                 DO 680 I=1,NCI
0149               680 PLCINT(I) = ALOG(PLCINT(I))/ALOG2
                C
0150              8031 CALL STGNUM (IFLAG1)
                C
0151                 GUESS1 = GUESS
                C
0152               620 CALL CHOOZ
                C
0153                 IF(NEXT.GE.500.OR.GUESS LT .001)  GO TO 1
                C
                C       IFLAG = NUMBER OF TIMES CHOOZ HAS BEEN CALLED
0154                 IFLAG = IFLAG + 1
                C
0155                 CALL OUTPUT
                C
0156                 CALL STGNUM (IFLAG1)
                C
0157                 IF (IFLAG1.EQ 0) GO TO 1
0158                 GUESS = GUESS1
0159                 GO TO 620
0160               1 KNSTG = NSTG
0161                 KNFAM = NFAM
0162                 KNCI = NCI
0163                 KNP = NP
0164                 KNMIS = NMIS
0165                 KNV = NV
0166                 GO TO 11
0167               806 WRITE(6,8032)
0168                 STOP
0169               100 FORMAT (3I5,F12.2,38X,5I2,2X,I2,I1)
0170               101 FORMAT (I2,1X,A4,3F6.3,3F5.3,  3X,2F6.3,7I3,1X,3I1)
0171               102 FORMAT(I2,1X,A4,2F10.0,10X,2F10.0)
0172               103 FORMAT (2X,2I3,4F10.0,10X,2F10.0)
0173               104 FORMAT (1H1)
0174               105 FORMAT (I2,A6,F4.2,6X,F7.0,F6.0,F7.0,I2,20I2)
0175               107 FORMAT (2X,3F10.2,25X,F10.2,4I2,2X,I3)
0176               213 FORMAT (16H STAGE COST DATA/6HOTITLE,3(16H  RECURRING   LC),68H  D
                   1EVELOPMENT  SUSTAINING  AVAILABLE  SHARED COST GROUPS  BATCH FACT/
```

```
                2  9X,12H(ETR OR WTR),
                3       5X,10H(ETR ONLY),6X,10H(WTR ONLY),29X,7HFROM TO//)
0177          214 FORMAT (1HO///17HOSHARED COST DATA/37HONO  TITLE   DEVELOPMENT   S
                1USTAINING//)
0178          215 FORMAT (1HO///22HOINTEGRATION COST DATA/59HOLOWER   UPPER    RECUR
                1RING    LC   DEVELOPMENT   SUSTAINING/14H GROUP   GROUP//)     /
0179          216 FORMAT (2X,A4,4X,A4,F11 2,F7.3,2F13 2)
0180          217 FORMAT (14H1MISSION MODEL/47HO   MISSION   VELOCITY   PAYLOAD
                1PRIORITY  ,20X 15HLAUNCH SCHEDULE/50X,  2014/1H /)
0181          219 FORMAT (1X,I2,1X,A6,2X,2F10 0,F10 2,4X,I2,2X,2014)
0182         2001 FORMAT (17H1NUMBER OF STAGES,8X,I5/19HONUMBER OF VEHICLES,6X,I5/
                1  19HONUMBER OF FAMILIES,6X,I5/28HONUMBER OF INTEGRATION COSTS,I2/
                X  24HONUMBER OF PAD COMPLEXES,4X,I2/
                2  19HONUMBER OF MISSIONS,6X,I5/16HONUMBER OF YEARS,9X,I5/
                3  17HOLAUNCH BASE YEAR,8X,I5/20HOTOTAL COST ESTIMATE,F12 2/14HOOPT
                4ION NUMBER, 11X,I5)
0183         2141 FORMAT (1X,I2,2X,A4,2X,2F13 2)
0184         4020 FORMAT(2013)
0185         4021 FORMAT(36H1ASSIGNMENT FOR CALCULATION OF GUESS//)
0186         4022 FORMAT(1H ,A6,5X,4(A4,1X))
0187         5000 FORMAT(2X,I2,6F6 0,2X,I2,6F6 0)
0188         5002 FORMAT(8X,2I3,3F6 0,2I3,3F6 0,2I3,3F6 0)
0189         5003 FORMAT (27X,A4,1X,5HSTAGE,17X,3(F9.2,F8.2))
0190         5004 FORMAT (27X,A4,1X,6HSHARED,16X,3(F9 2,F8 2))
0191         5005 FORMAT (27X,15HINTEGRATION OF ,A4,5H AND ,A4,8X,F8 2,2(9X,F8 2))
0192         8001 FORMAT
                1  (1X,A4,1X,3(F9 2,F7 3),F13 2,F12.2,2X,I4,1X,I4,2X,4I4,I9)
0193         8003 FORMAT (4X,4F10 3)
0194         8004 FORMAT (3X,19HRECURRING COST TYPE,I2,22H  FOR X LESS THAN OR =,
                1 F6.2,14H, TOTAL COST =,F6.2,19H.FOR X GREATER THAN,F6 2,
                2 14H, TOTAL COST =,F6 2,4H X +,F6 2)
0195         8005 FORMAT(16HOREUSEABLE STAGE, 4X,20HUNIT PURCHASE PRICE=,F7 3)
0196         8032 FORMAT (1HO///5X,26HEND OF DATA - JOB COMPLETE)
0197         9003 FORMAT(1HO///14HOPAD COST DATA/12HOND  COMPLEX,2X,11HLAUNCHES/YR,
                1  37X,5HPAD 1,12X,
                2  5HPAD 2,12X,5HPAD 3/59X,3(11HDEV    SUST,6X)//)
0198         9005 FORMAT (I4,2X,A4,F5 0)
0199         9006 FORMAT (1X,I2,2X,A4,5X,F6.2)
0200         9021 FORMAT (4X,2F10.0,I4,F7 3,F6 0)
0201              END
```

FORTRAN IV G LEVEL 1, MOD 4 MAIN DATE = 70113 15/13/01

TOTAL MEMORY REQUIREMENTS 002004 BYTES

```
F68-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
        VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
EW0000      NAME*HOX02HN(R)
EW0461  IBCOM*
EW0461  CAPABL
EW0461  DECISN
EW0461  AVAIL
EW0461  STGNUM
EW0461  CHOOZ
EW0461  OUTPUT
EW0461  ALOG

                                        MODULE MAP

 CONTROL SECTION                    ENTRY
   NAME     ORIGIN  LENGTH            NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION

 MAIN          00    2004
 NONDIM      20D8      1C
 VEHCLE      20F8     4B0
 MISSON      25A8     A90
 STAGE       3038     6E4
 DSCOST      3720    1838
 INT         4F58     3C0
 PADS        5318    378C
 VEHREC      8AA8    3CF4
 REUSBL      C7A0     190
 CAPMAT      C930     FA4
 FAMILY      08D0     2D0
 INTREC      0BA8     140
 STGREC      0CE8     670
 MISREQ      E358     640
 VEHIN       E998     1F8
 CHOOZE      EB90      84
 SAVEOP      EC18     194
 SCRACH      ED80    53FC

 ENTRY ADDRESS       00
 TOTAL LENGTH     14180

 ***HOX02HN   NOW REPLACED IN DATA SET



(17)          OS/360  FORTRAN H                                DATE  70 113/15 14.29

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002            SUBROUTINE MATCH(HYFLAG)
                  C    MATCH DECISION COSTS WITH VEHICLE
                  C
   ISN 0003            INTEGER*2 NONREC,LYD,VEH,NPAD,NPFAM,NPSTG,NPINTL,NPINTU
   ISN 0004            COMMON/NONDIM/ NUMD,NSTG,MYRS,NFAM,NCI,NP,NV
   ISN 0005            COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
   ISN 0006            COMMON/STAGE/ LSA(40),SNR(41),STG(40),SUS(40),NYS(40),SUSLS(40,2),
                      1  NFS(40,4)
   ISN 0007            COMMON/DSCOST/ DS(100),SUST(100),NYD(100),LYD(100),NONREC(120,20)
   ISN 0008            COMMON/INT/ NFML(40),NFMU(40),DINT(40),SINT(40),SINTLS(40,2)
   ISN 0009            COMMON/PADS/ PFAMD(30,5,3),PFAMS(30,5,3),NPFAM(30,5),
                      1  PSTGD(30,10,3),PSTGS(30,10,3),NPSTG(30,10),PINTS(30,5,3),
                      2  NPINTL(30,51,NPINTU(30,5),PAD(30)
   ISN 0010            COMMON/SCRACH/          MAT(100),LYF(30),NYF(30),MAF(30,3),
                      1  MAIC(40,3),MAPF(30,5),MAPS(30,10),MAPI(30,5),DUMMY(4045),
                      2  MAS(40,3), VM(2,120)
                  C
   ISN 0011            DO 66 I = 1,NV
   ISN 0012            JX = 1
   ISN 0013            KX = 1
   ISN 0014            DO 64 J = 1,20
   ISN 0015            NONREC(I&NV,J) = 0
   ISN 0016         64 NONREC(I,J) = 0
   ISN 0017            DO 65 MS= 1,4
   ISN 0018            K = VEH(MS,I)
   ISN 0019            IF(K EQ 0) GO TO 66
   ISN 0021            IF (MAS(K,1)  EQ  0) GO TO 9050
   ISN 0023            NONREC(I,JX) = MAS(K,1)
   ISN 0024            NONREC(I&NV,KX) = MAS(K,1)
   ISN 0025            JX = JX & 1
   ISN 0026            KX = KX & 1
   ISN 0027            IF(JX GT 20 OR KX GT 20)  GO TO 93
   ISN 0029         9050 IF (MAS(K,2) EQ 0) GO TO 9051
   ISN 0031            NONREC(I,JX) = MAS(K,2)
   ISN 0032            JX = JX & 1
   ISN 0033            IF (JX GT 20) GO TO 93
   ISN 0035         9051 IF (MAS(K,3).EQ.0) GO TO 9052
   ISN 0037            NONREC(I&NV,KX) = MAS(K,3)
   ISN 0038            KX = KX & 1
   ISN 0039            IF (KX GT 20) GO TO 93
   ISN 0041         9052 IF (NPAD(1,I).EQ 0) GO TO 9053
   ISN 0043              N1 = NPAD(1,I)
```

```
ISN 0044          DO 9054 J = 1,10
ISN 0045          IF (NPSTG(N1,J)  NE K) GO TO 9054
ISN 0047          IF (MAPS(N1,J) EQ 0) GO TO 9053
ISN 0049          NONREC(I,JX) = MAPS(N1,J)
ISN 0050          JX = JX & 1
ISN 0051          IF (JX GT 20) GO TO 93
ISN 0053          GO TO 9053
ISN 0054       9054 CONTINUE
ISN 0055       9053 IF (NPAD(2,I) EQ 0) GO TO 63
ISN 0057          N1 = NPAD(2,I)
ISN 0058          DO 9055 J = 1,10
ISN 0059     *    IF (NPSTG(N1,J) NE.K) GO TO 9055
ISN 0061          IF (MAPS(N1,J).EQ 0) GO TO 63
ISN 0063          NONREC(I&NV,KX) = MAPS(N1,J)
ISN 0064          KX = KX & 1
ISN 0065     *    IF (KX GT 20) GO TO 93
ISN 0067          GO TO 63
ISN 0068       9055 CONTINUE
               C     ***   PICK UP SHARED COSTS   ***
ISN 0069          63 DO 885 KY=1,4
ISN 0070          KZ=NFS(K,KY)
ISN 0071          IF(KZ EQ 0) GO TO 885
ISN 0073          IF (MAF(KZ,1) EQ 0) GO TO 9056
ISN 0075          NONREC(I,JX) = MAF(KZ,1)
ISN 0076          NONREC(I&NV,KX) = MAF(KZ,1)
ISN 0077          JX=JX&1
ISN 0078          KX = KX & 1
ISN 0079          IF(JX.GT.20.OR KX GT 20)    GO TO 93
ISN 0081       9056 IF (MAF(KZ,2) EQ.0) GO TO 9057
ISN 0083          NONREC(I,JX) = MAF(KZ,2)
ISN 0084          JX = JX & 1
ISN 0085          IF (JX GT 20) GO TO 93
ISN 0087       9057 IF (MAF(KZ,3).EQ.0) GO TO 9058
ISN 0089          NONREC(I&NV,KX) = MAF(KZ,3)
ISN 0090          KX = KX & 1
ISN 0091          IF (KX.GT 20) GO TO 93
ISN 0093       9058 IF (NPAD(1,I) EQ 0) GO TO 9059
ISN 0095          N1 = NPAD(1,I)
ISN 0096          DO 9060 J = 1,5
ISN 0097          IF (NPFAM(N1,J) NE.KZ) GO TO 9060
ISN 0099     *    IF (MAPF(N1,J) EQ 0) GO TO 9059
ISN 0101          NONREC(I,JX) = MAPF(N1,J)
                  *
```

```
ISN 0102          JX = JX & 1
ISN 0103          IF (JX GT 20) GO TO 93
ISN 0105          GO TO 9059
ISN 0106       9060 CONTINUE
ISN 0107       9059 IF (NPAD(2,I) EQ 0) GO TO 885
ISN 0109          N1 = NPAD(2,I)
ISN 0110     `    DO 9061 J = 1,5
ISN 0111          IF (NPFAM(N1,J) NE KZ) GO TO 9061
ISN 0113          IF (MAPF(N1,J) EQ 0) GO TO 885
ISN 0115          NONREC(I&NV,KX) = MAPF(N1,J)
ISN 0116     *    KX = KX & 1
ISN 0117          IF (KX GT 20) GO TO 93
ISN 0119          GO TO 885
ISN 0120       9061 CONTINUE
ISN 0121       )885 CONTINUE
               C     ***   PICK UP INTEGRATION COSTS   ***
ISN 0122          IF(MS EQ 4) GO TO 65
ISN 0124          IF(VEH(MS&1,I) EQ 0) GO TO 65
ISN 0126          K1=VEH(MS&1,I)
ISN 0127          IF(NCI EQ 0) GO TO 9062
ISN 0129          DO 89 J=1,NCI
ISN 0130          DO 887 KY=1,4
ISN 0131          IF(NFML(J) NE NFS(K,KY)) GO TO 887
ISN 0133          DO 886 KZ=1,4
ISN 0134          IF(NFMU(J) EQ.NFS(K1,KZ)) GO TO 888
ISN 0136       886 CONTINUE
ISN 0137       887 CONTINUE
ISN 0138          GO TO 89
ISN 0139       888 IF (MAIC(J,1) EQ 0) GO TO 9063
ISN 0141          NONREC(I,JX) = MAIC(J,1)
ISN 0142          NONREC(I&NV,KX) = MAIC(J,1)
ISN 0143          JX = JX & 1
ISN 0144          KX = KX & 1
ISN 0145          IF(JX GT 20 OR KX GT 20)    GO TO 93
ISN 0147       9063 IF (MAIC(J,2).EQ 0) GO TO 9064
ISN 0149          NONREC(I,JX) = MAIC(J,2)
ISN 0150          JX = JX & 1
ISN 0151          IF (JX GT.20) GO TO 93
ISN 0153       9064 IF (MAIC(J,3).EQ.0) GO TO 89
ISN 0155          NONREC(I&NV,KX) = MAIC(J,3)
ISN 0156          KX = KX & 1
ISN 0157          IF (KX GT 20) GO TO 93
```

```
ISN 0159        89 CONTINUE
ISN 0160      9062 DO 9065 M = 1,2
ISN 0161           IF (NPAD(M,I).EQ 0) GO TO 9065
ISN 0163           N1 = NPAD(M,I)
ISN 0164           DO 9066 J = 1,4
ISN 0165           IF (NFS(K,J) EQ 0) GO TO 9065
ISN 0167           DO 9067 KY = 1,5
ISN 0168           IF (NPINTL(N1,KY) NE NFS(K,J)) GO TO 9067
ISN 0170           DO 9068 KZ = 1,4
ISN 0171           IF (NPINTU(N1,KY) EQ NFS(K1,KZ)) GO TO 9069
ISN 0173      9068 CONTINUE
ISN 0174           GO TO 9067
ISN 0175      9069 IF (MAPI(N1,KY).EQ 0) GO TO 9067
ISN 0177           IF (M EQ 2) GO TO 9070
ISN 0179           NONREC(I,JX) = MAPI(N1,KY)
ISN 0180           JX = JX + 1
ISN 0181           IF (JX.GT.20) GO TO 93
ISN 0183           GO TO 9067
ISN 0184      9070 NONREC(IENV,KX) = MAPI(N1,KY)
ISN 0185           KX = KX + 1
ISN 0186           IF (KX GT 20) GO TO 93
ISN 0188      9067 CONTINUE
ISN 0189      9066 CONTINUE
ISN 0190      9065 CONTINUE
ISN 0191        65 CONTINUE
ISN 0192        66 CONTINUE
ISN 0193           RETURN
ISN 0194        93 WRITE(6,220) I
ISN 0195           HYFLAG = 1
ISN 0196           RETURN
ISN 0197       220 FORMAT(45HDEXCEEDED 20  NON-RECURRING COSTS FOR VEHICLE,I4)
ISN 0198           END

-**** END OF COMPILATION ******
```

```
F68-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000     NAME MOX02MT(R)
IEW0461  IBCOM*
```

CROSS REFERENCE TABLE

| CONTROL SECTION NAME | ORIGIN | LENGTH | ENTRY NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|---|---|---|
| MATCH | 00 | DB4 | | | | | | | | |
| NONDIM | D88 | 1C | | | | | | | | |
| VEHCLE | D08 | 480 | | | | | | | | |
| STAGE | 1288 | 6E4 | | | | | | | | |
| DSCOST | 1970 | 1038 | | | | | | | | |
| INT | 31A8 | 3C0 | | | | | | | | |
| PADS | 3568 | 378C | | | | | | | | |
| SCRACH | 6CF8 | 53FC | | | | | | | | |

| LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION | LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION |
|---|---|---|---|---|---|
| 128 | NONDIM | NONDIM | 12C | VEHCLE | VEHCLE |
| 130 | STAGE | STAGE | 134 | DSCOST | DSCOST |
| 138 | INT | INT | 13C | PADS | PADS |
| 140 | PADS | PADS | 144 | PADS | PADS |
| 148 | SCRACH | SCRACH | 14C | SCRACH | SCRACH |
| 150 | IBCOM* | *UNRESOLVED | | | |

ENTRY ADDRESS 00
TOTAL LENGTH COF8

****MOX02MT  NOW REPLACED IN DATA SET

DIAGNOSTIC MESSAGE DIRECTORY

IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED

```
(17)          OS/360  FORTRAN H                                  DATE  70 113/15 15 17

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002           SUBROUTINE MATE
 ISN 0003           INTEGER*2 VEH,NPAD
 ISN 0004           REAL ISP,ISPA,LENT
 ISN 0005           COMMON/NONDIM/ NUMD,NSTG,MYRS,NFAM,NCI,NP,NV
 ISN 0006           COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
 ISN 0007           COMMON/MISREQ/ IVEHA(50),VLR(50),WPR(50),NPLS(50),MR(50),NRR(50),
                   1   RPLM(50),NTRIP(50)
 ISN 0008           COMMON/VEHIN/ IV,KNV,IS,NMIS,NOPT,KODE(40),KODEP(30),IP,RPLO(50)
 ISN 0009           COMMON/BATCH/ NRP(60),
                   1          B1(60),B2(60),B3(60),B4(60),KODEV(60),NVS(60),MRV(60)
 ISN 0010           COMMON/SCRACH/ LZ(50),MZ(50),VMW(2,60),NEH(4,60),NPAX(2,60),
                   1 NST(40),THRT(40),DIAM(40),TSL(40),LENT(40),WTFU(40),WTIN(40),
                   2  ISP(40),WINT(3,60),KX,NX,WGHT(40),WF(4),WT(4),ISPA(4),THUT(4),
                   3  PRT(60), DUMMY(3937),VM(2,120)
                C
 ISN 0011           DATA PI,N,VREF,CLI/3 1416,2,25573.,28.5/
 ISN 0012           NV1 = NV & 1
 ISN 0013           DO 34 I = NV1,60
 ISN 0014           VEH(1,I)=0
 ISN 0015           DO 34 J=1,3
 ISN 0016           VEH(J&1,I)=0
 ISN 0017        34 WINT(J,I)=0.0
 ISN 0018           NX = 0
 ISN 0019           IMAX=0
 ISN 0020           JMAX=0
 ISN 0021           KMAX=0
 ISN 0022           LMAX=0
 ISN 0023           DO 35 I=1,NSTG
 ISN 0024           IF(NST(I) EQ 0)   GO TO 36
 ISN 0026        33 WGHT(I)=WTFU(I)&WTIN(I)
 ISN 0027           IF(NST(I) EQ 1) IMAX=I
 ISN 0029           IF(NST(I).EQ 2) JMAX=I
 ISN 0031           IF(NST(I).EQ.3) KMAX=I
 ISN 0033           IF(NST(I) EQ.4) LMAX=I
 ISN 0035        35 CONTINUE
 ISN 0036        36 IF(IMAX EQ 0) GO TO 600
 ISN 0038           IF(JMAX EQ 0) JMAX=IMAX
 ISN 0040           IF(KMAX EQ 0) KMAX=JMAX
 ISN 0042           IF(LMAX.EQ 0) LMAX=KMAX
 ISN 0044           IM1=IMAX&1
 ISN 0045           JM1=JMAX&1
 ISN 0046           DO 500 I=1,IMAX

 ISN 0047           WF(1)=WTFU(I)
 ISN 0048           WT(1)=WTIN(I)
 ISN 0049           ISPA(1)=ISP(I)
 ISN 0050           THUT(1)=THRT(I)
 ISN 0051           DO 400 J=IM1,KMAX
 ISN 0052           WINX=PI*(DIAM(I)&DIAM(J))*(SQRT((LENT(J))**2&((DIAM(I)-DIAM(J))*
                   1  0.5)**2))*5.0*0.5
 ISN 0053            IF (THRT(I) LT.1 2*(WGHT(I)&WGHT(J)&WINX)) GO TO 400
 ISN 0055            IF (THRT(I) GT 3 5*(WGHT(I)&WGHT(J)&WINX)) GO TO 400
 ISN 0057            IF (DIAM(J) GT 1 2*DIAM(I)) GO TO 400
 ISN 0059            IF (DIAM(I).GT 3 5*DIAM(J)) GO TO 400
 ISN 0061           WF(2)=WTFU(J)
 ISN 0062           WT(2)=WTIN(J)
 ISN 0063           ISPA(2)=ISP(J)
 ISN 0064           THUT(2)=THRT(J)
 ISN 0065           M=0
 ISN 0066           PR=0
 ISN 0067           VDES=0
 ISN 0068           CALL PERF(N,M,VDES,WPL,VREF,PR,WF,WT,ISPA,THUT,TSL(1),DIAM(I),
                   1  CLI,IERR)
 ISN 0069            IF (IERR NE 0) GO TO 60
 ISN 0071           NX = NX & 1
 ISN 0072           KX = NV & NX
 ISN 0073           NJ=J
 ISN 0074           VEH(1,KX)=I
 ISN 0075           VEH(2,KX)=J
 ISN 0076           WINT(1,KX)=WINX
 ISN 0077           PRT(KX)=WPL

 ISN 0078           CALL MISMAT

 ISN 0079           IF(KX EQ 100)  GO TO 60
 ISN 0081           IF(KX GE 60)  GO TO 600
 ISN 0083        60 DO 300 K=IM1,LMAX
 ISN 0084           WINY=PI*(DIAM(J)&DIAM(K))*(SQRT((LENT(K))**2&((DIAM(J)-DIAM(K))
                   1  *0 5)**2))*5 0*0 5
 ISN 0085            IF (THRT(J) LT 0 37*(WGHT(J)&WGHT(K)&WINY)) GO TO 300
 ISN 0087            IF (THRT(J).GT 1 25*(WGHT(J)&WGHT(K)&WINY)) GO TO 300
 ISN 0089            IF (THRT(I) LT.1 2*(WGHT(I)&WGHT(J)&WGHT(K)&WINX&WINY)) GO TO 300
 ISN 0091            IF (THRT(I) GT 3.0*(WGHT(I)&WGHT(J)&WGHT(K)&WINX&WINY)) GO TO 300
 ISN 0093            IF (DIAM(K) GT 1 2*DIAM(J)) GO TO 300
 ISN 0095            IF (DIAM(J) GT.3.5*DIAM(K)) GO TO 300
```

```
ISN 0097            WF(3)=WTFU(K)
ISN 0098            WT(3)=WTIN(K)
ISN 0099            ISPA(3)=ISP(K)
ISN 0100            THUT(3)=THRT(K)
ISN 0101            M=1
ISN 0102            PR=0
ISN 0103            VDES=0
ISN 0104            CALL PERF(N,M,VDES,WPL,VREF,PR,WF,WT,ISPA,THUT,TSL(I),DIAM(I),
                   1  CLI,IERR)
ISN 0105            IF (IERR NE.0) GO TO 70
ISN 0107            NX = NX + 1
ISN 0108            KX = NV + NX
ISN 0109            VEH(1,KX)=I
ISN 0110            VEH(2,KX)=J
ISN 0111            VEH(3,KX)=K
ISN 0112            WINT(1,KX)=WINX
ISN 0113            WINT(2,KX)=WINY
ISN 0114            PRT(KX)=WPL
                C
ISN 0115            CALL MISMAT
                C
ISN 0116            IF(KX EQ 100)  GO TO 69
ISN 0118            IF(KX GE 601  GO TO 600
ISN 0120         69 IF(K GT KMAX) GO TO 300
ISN 0122         70 DO 200 L=JM1,LMAX
ISN 0123            IF(L EQ NJ) GO TO  200
ISN 0125            WINZ=PI*(DIAM(K)+DIAM(L))*(SQRT((LENT(L))**2+((DIAM(K)-DIAM(L))
                   1  *0.5)**2))**5.0**0.5
ISN 0126            IF(THRT(K) LT.0.30*(WGHT(K)+WGHT(L)+WINZ)) GO TO 200
ISN 0128            IF(THRT(K).GT.1.25*(WGHT(K)+WGHT(L)+WINZ)) GO TO 200
ISN 0130            IF(THRT(J) LT 0.32*(WGHT(J)+WGHT(K)+WGHT(L)+WINY+WINZ)) GO TO 200
ISN 0132            IF(THRT(J).GT 1 50*(WGHT(J)+WGHT(K)+WGHT(L)+WINY+WINZ)) GO TO 200
ISN 0134            IF(THRT(I) LT.1 20*(WGHT(I)+WGHT(J)+WGHT(K)+WGHT(L)+WINX+WINY+WINZ
                   1  )) GO TO 200
ISN 0136            IF(THRT(I) GT 3 00*(WGHT(I)+WGHT(J)+WGHT(K)+WGHT(L)+WINX+WINY+WINZ
                   1  )) GO TO 200
ISN 0138            IF(DIAM(L) GT 1.2*DIAM(K)) GO TO 200
ISN 0140            IF(DIAM(K) GT 4 0*DIAM(L)) GO TO 200
ISN 0142            WF(4)=WTFU(L)
ISN 0143            WT(4)=WTIN(L)
ISN 0144            ISPA(4)=ISP(L)
ISN 0145            THUT(4)=THRT(L)

ISN 0146            M=2
ISN 0147            PR=0
ISN 0148            VDES=0
ISN 0149            CALL PERF(N,M,VDES,WPL,VREF,PR,WF,WT,ISPA,THUT,TSL(I),DIAM(I),
                   1  CLI,IERR)
ISN 0150            IF(IERR NE 0) GO TO 200
ISN 0152            NX = NX + 1
ISN 0153            KX = NV + NX
ISN 0154            VEH(1,KX)=I
ISN 0155            VEH(2,KX)=J
ISN 0156            VEH(3,KX)=K
ISN 0157            VEH(4,KX)=L
ISN 0158            WINT(1,KX)=WINX
ISN 0159            WINT(2,KX)=WINY
ISN 0160            WINT(3,KX)=WINZ
ISN 0161            PRT(KX)=WPL
                C
ISN 0162            CALL MISMAT
                C
ISN 0163            IF(KX EQ.100)  GO TO 200
ISN 0165            IF(KX GE 60)  GO TO 600
ISN 0167         200 CONTINUE
ISN 0168         300 CONTINUE
ISN 0169         400 CONTINUE
ISN 0170         500 CONTINUE
ISN 0171         600 NV = NV + NX
ISN 0172             L = NV - NX + 1
ISN 0173             DO 601 J = L,NV
ISN 0174             VM(1,J) = VM(1,J)
ISN 0175         601 VM(2,J) = VM(2,J)
ISN 0176             RETURN
ISN 0177             END
```

****** END OF COMPILATION ******

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOXO2MA(R)
IEW0461  PERF
IEW0461  SQRT
IEW0461  MISMAT

                                        CROSS REFERENCE TABLE

  CONTROL SECTION                       ENTRY
    NAME     ORIGIN  LENGTH               NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
  MATE          00     842
  NONDIM       848      1C
  VEHCLE       868     4B0
  MISREQ      1018     640
  VEHIN       1658     1F8
  BATCH       1850     780
  SCRACH      1FD0    53FC


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
      178           NONDIM         NONDIM                   17C           VEHCLE         VEHCLE
      180           MISREQ         MISREQ                   184           VEHIN          VEHIN
      188           BATCH          BATCH                    18C           SCRACH         SCRACH
      190           SCRACH         SCRACH                   194           SCRACH         SCRACH
      198           PERF           $UNRESOLVED              19C           SQRT           $UNRESOLVED
      1A0           MISMAT         $UNRESOLVED               90           SCRACH         SCRACH
       94           SCRACH         SCRACH                    98           SCRACH         SCRACH
       9C           SCRACH         SCRACH
 ENTRY ADDRESS      00
 TOTAL LENGTH     7300

****MOXO2MA   NOW REPLACED IN DATA SET

                                        DIAGNOSTIC MESSAGE DIRECTORY

   IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED



(17)          OS/360  FORTRAN H                                DATE  70 113/15 26 34

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
    ISN 0002            SUBROUTINE MISMAT
    ISN 0003            REAL LENT,ISP,ISPA
    ISN 0004            INTEGER*2 VEH,NPAD
    ISN 0005            COMMON/NONDIM/ NUMD,NSTG,MYRS,NFAM,NC1,NP,NV
    ISN 0006            COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
    ISN 0007            COMMON/MISREQ/ IVEHA(50),VLR(50),WPR(50),NPLS(50),MR(50),NRR(50),
                       1   RPLH(50),NTRIP(50)
    ISN 0008            COMMON/VEHIN/ IV,KNV,IS,NMIS,NOPT,KODE(40),KODEP(30),IP,RPLO(50)
    ISN 0009            COMMON/BATCH/ NRP(60),
                       1          B1(60),B2(60),B3(60),B4(60),KODEV(60),NVS(60),MRV(60)
    ISN 0010            COMMON/SCRACH/ LZ(50),NZ(50),VMH(2,60),NEH(4,60),NPAX(2,60),
                       1 NST(40),THRT(40),DIAM(40),TSL(40),LENT(40),WTFU(40),WTIN(40),
                       2  ISP(40),HINT(3,60),KX,NX,NGHT(40),WF(4),WT(4),ISPA(4),THUT(4),
                       3  PRT(60), DUMMY(3937),VH(2,120)
    ISN 0011            DATA VREF,CLI/25573 ,28 5/
                     C
    ISN 0012            CF1=0.
    ISN 0013            KNS = 1
    ISN 0014            IF(NV.EQ 0)  GO TO 101
    ISN 0016            DO 100 MJ=1,NV
    ISN 0017            IF(VEH(1,KX).NE VEH(1,MJ))  GO TO 100
    ISN 0019            IF(VEH(2,KX) NE VEH(2,MJ))   GO TO 100
    ISN 0021            IF(VEH(3,KX) NE.VEH(3,MJ))   GO TO 20
    ISN 0023            IF(VEH(4,KX) NE.VEH(4,MJ))  GO TO 30
    ISN 0025            KX = 100
    ISN 0026            NX = NX - 1
    ISN 0027            RETURN
    ISN 0028          20 IF(VEH(3,MJ) NE 0)  GO TO 100
    ISN 0030            IF(KNS EQ 3)  GO TO 100
    ISN 0032            CF1 = EXP(B1(MJ))
    ISN 0033            KNS = 2
    ISN 0034            GO TO 100
    ISN 0035          30 IF(VEH(4,MJ) NE.0)  GO TO 100
    ISN 0037            CF1 = EXP(B1(MJ))
    ISN 0038            KNS = 3
    ISN 0039         100 CONTINUE
    ISN 0040         101 GO TO (200,300,400), KNS
    ISN 0041         200 M=0
    ISN 0042            N=2
    ISN 0043            IF(VEH(3,KX) NE 0) N=3
    ISN 0045            IF(VEH(4,KX).NE 0) N=4
    ISN 0047            GO TO 500
```

```
ISN 0048       300 N=2
ISN 0049           M=1
ISN 0050           IF(VEH(4,KX) NE 0) M=2
ISN 0052           GO TO 500
ISN 0053       400 N=3
ISN 0054           M=1
ISN 0055       500 K1=VEH(1,KX)
ISN 0056           DO 501 I=1,4
ISN 0057           IF(VEH(I,KX).EQ 0) GO TO 502
ISN 0059           K=VEH(I,KX)
ISN 0060           WF(I)=WTFU(K)
ISN 0061           WT(I)=WTIN(K)
ISN 0062           ISPA(I)=ISP(K)
ISN 0063           THUT(I)=THRT(K)
ISN 0064       501 CONTINUE
ISN 0065       502 DO 503 I=1,NMIS
ISN 0066           LZ(I) = 0
ISN 0067           PR=CF1
ISN 0068           VDES=VLR(I)
ISN 0069           CALL PERF(N,M,VDES,WPL,VREF,PR,WF,WT,ISPA,THUT,TSL(K1),DIAM(K1),
                  1  CLI,IERR)
ISN 0070           IF(WPL.GT WPR(I).AND.IERR EQ 0) LZ(I) = 1
ISN 0072       503 CONTINUE
ISN 0073           CALL PACK(LZ,VM(1,KX),NMIS,1)
ISN 0074           RETURN
ISN 0075           END

****** END OF COMPILATION ******
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
           VARIABLE OPTIONS USED - SIZE=(126976,245760)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02MH(R)
IEW0461  PACK
IEW0461  PERF
IEW0461  EXP

                                        CROSS REFERENCE TABLE

  CONTROL SECTION                    ENTRY
    NAME     ORIGIN  LENGTH           NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
  MISMAT        00     4C6
  NONDIM       4C8      1C
  VEHCLE       4E8     4B0
  MISREQ       998     640
  VEHIN        FD8     1F8
  BATCH       1100     780
  SCRACH      1950    53FC


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
      140           NONDIM        NONDIM                    144           VEHCLE        VEHCLE
      148           MISREQ        MISREQ                    14C           VEHIN         VEHIN
      150           BATCH         BATCH                     154           SCRACH        SCRACH
      158           SCRACH        SCRACH                    15C           SCRACH        SCRACH
      160           PACK          $UNRESOLVED               164           PERF          $UNRESOLVED
      168           EXP           $UNRESOLVED                A4           SCRACH        SCRACH
       A8           SCRACH         SCRACH                   AC           SCRACH        SCRACH
       B0           SCRACH         SCRACH                   C4           SCRACH        SCRACH
       CC           VEHIN          VEHIN
 ENTRY ADDRESS      00
 TOTAL LENGTH     6050

****MOX02MH   NOW REPLACED IN DATA SET

                                        DIAGNOSTIC MESSAGE DIRECTORY
```

```
(17)          OS/360  FORTRAN H                                DATE  70 113/15.27.03

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002            SUBROUTINE OUTPUT
                     C     ***  PRINT OUT BEST ASSIGNMENT  ***
                     C
   ISN 0003            DOUBLE PRECISION  MISNAM
   ISN 0004            INTEGER*2 LETT,LYR,MIN,NPAD,VEH,NMULT
   ISN 0005            COMMON/SAVEOP/MISNAM(50),IBY
   ISN 0006            COMMON/NONDIM/ NUMO,NSTG,MYRS,NFAM,NCI,NP,NV
   ISN 0007            COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
   ISN 0008            COMMON/MISSON/ LETT(250),LYR(250),LTR(50),YRLM(250),MIN(250),NM
   ISN 0009            COMMON/STAGE/ LSA(40),SNR(41),STG(40),SUS(40),NYS(40),SUSLS(40,2),
                      1  NFS(40,4)
   ISN 0010            COMMON/MISREQ/ IVEHA(50),VLR(50),HPR(50),NPLS(50),MR(50),NRR(50),
                      1   RPLM(50),NTRIP(50)
   ISN 0011            COMMON/VEHREC/ RECUR(60,20,2),NMULT(60,50),IFLAG
   ISN 0012            DATA ETR/1HE/
   ISN 0013            DATA WTR/1HW/
                     C
   ISN 0014            WRITE (6,4010)
   ISN 0015            DO 805 J=1,NM
   ISN 0016            L=LETT(J)
   ISN 0017            K=LYR(J)
   ISN 0018            M=1899+IBY+K
   ISN 0019            IF(YRLM(J).NE 0.0) GO TO 804
   ISN 0021            IF(LETT(J-1) NE L) WRITE(6,206)MISNAM(L),VLR(L),HPR(L),RPLM(L),M,
                      1   YRLM(J)
   ISN 0023            IF(LETT(J-1) EQ L) WRITE(6,2061) M,YRLM(J)
   ISN 0025            GO TO 805
   ISN 0026          804 I1 = MIN (J)
   ISN 0027            IF (I1.LE NV) I = I1
   ISN 0029            IF (I1 GT NV) I = I1 - NV
   ISN 0031            IA=VEH(1,I)
   ISN 0032            IB=VEH(2,I)
   ISN 0033            IC=VEH(3,I)
   ISN 0034            ID=VEH(4,I)
   ISN 0035            X = NMULT(I,L)
   ISN 0036            X = YRLM(J)*X
   ISN 0037            IF (I1 LE NV) TR = ETR
   ISN 0039            IF (I1.GT.NV) TR = WTR
   ISN 0041            IF(LETT(J-1) NE L) WRITE(6,202)MISNAM(L),VLR(L),HPR(L),RPLM(L),M,
                      1 X,STG(IA),STG(IB),STG(IC),STG(ID),TR
   ISN 0043            IF(LETT(J-1) EQ L) WRITE(6,2021) M,X,        STG(IA),STG(IB),
                      1  STG(IC),STG(ID),TR

   ISN 0045          805 CONTINUE
   ISN 0046            RETURN
   ISN 0047          202 FORMAT (1X,A6,6X,F10.0,4X,F10.0,   F10 0,5X,I4,4X,F5.2,9X,5(A4,1X)
                        1)
   ISN 0048          206 FORMAT (1X,A6,6X,F10 0,4X,F10 0,   F10 0,5X,I4,4X,F5 2,9X,
                        1 32HNO LAUNCH VEHICLE CAN ACCOMPLISH)
   ISN 0049         2021 FORMAT (52X,I4,4X,F5.2,9X,5(A4,1X))
   ISN 0050         2061 FORMAT (52X,I4,4X,F5.2,9X,32HNO LAUNCH VEHICLE CAN ACCOMPLISH)
   ISN 0051         4010 FORMAT(8H1MISSION,4X,14HCHARACTERISTIC,4X,7HPAYLOAD,4X,6HRETURN,
                        1 4X,6HLAUNCH,
                        2  4X,6HNUMBER,10X,7HOPTIMUM/7H  TITLE,4X,16HVELOCITY(FT/SEC),4X,
                        3  5H(LBS),4X,7HPAYLOAD,5X,4HYEAR,3X,11HOF LAUNCHES,4X,
                        4  14HLAUNCH VEHICLE//)
   ISN 0052             END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
         VARIABLE OPTIONS USED - SIZE=(126976,245760)              DEFAULT OPTION(S) USED
IEW0000      NAME HOX02OP(R)
IEW0461  IBCOM=
```

CROSS REFERENCE TABLE

| CONTROL SECTION NAME | ORIGIN | LENGTH | ENTRY NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|---|---|---|
| OUTPUT | 00 | 5E4 | | | | | | | | |
| SAVEOP | 5E8 | 194 | | | | | | | | |
| NONDIM | 780 | 1C | | | | | | | | |
| VEHCLE | 7A0 | 4B0 | | | | | | | | |
| MISSON | C50 | A90 | | | | | | | | |
| STAGE | 16E0 | 6E4 | | | | | | | | |
| MISREQ | 1DC8 | 640 | | | | | | | | |
| VEHREC | 2408 | 3CF4 | | | | | | | | |

| LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION | LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION |
|---|---|---|---|---|---|
| 250 | SAVEOP | SAVEOP | 254 | NONDIM | NONDIM |
| 258 | VEHCLE | VEHCLE | 25C | MISSON | MISSON |
| 260 | STAGE | STAGE | 264 | MISREQ | MISREQ |
| 268 | VEHREC | VEHREC | 26C | VEHREC | VEHREC |
| 270 | VEHREC | VEHREC | 274 | IBCOM= | $UNRESOLVED |

ENTRY ADDRESS 00
TOTAL LENGTH 6100

****HOX02OP  NOW REPLACED IN DATA SET

DIAGNOSTIC MESSAGE DIRECTORY

IEW0461 WAR  IBCOM PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED

EXTERNAL SYMBOL DICTIONARY

| SYMBOL | TYPE | ID | ADDR | LENGTH | LD ID |
|---|---|---|---|---|---|
| PACK | SD | 01 | 000000 | 0000E8 | |
| UNPACK | LD | | 000052 | | 01 |
| ITEM | LD | | 000096 | | 01 |

15.27 4/23/70

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01FEB69   4/23/70

                                    1 *       SUBROUTINE  PACK ( L, H, I, N )
                                    2 *
                                    3 *             THIS ROUTINE PACKS I WORDS IN THE L ARRAY  TO THE
                                    4 *             ARRAY H   DATA ITEMS L ARE TRUNCATED ON THE LEFT
                                    5 *             AND ONLY THE N LOW ORDER BITS ARE RETAINED
                                    6 *             PACKED DATA IN H IS LEFT JUSTIFIED WITH 32/N ITEMS
                                    7 *             PER WORD.
                                    8 *
000000                              9 PACK    CSECT
000000                             10         USING  *,15          USE REG 15 FOR BASE
000000 9027 D01C          0001C    11         STM    2,7,28(13)    SAVE REGS
000004 9825 1000          00000    12         LM     2,5,0(1)      LOAD ADDRESSES OF ARGUMENTS
000008 5844 0000          00000    13         L      4,0(4)        I TO REG 4 - NO  OF ITEMS TO BE PACKED
00000C 5875 0000          00000    14         L      7,0(5)        N TO REG 7 - NO  OF BITS/ITEM
000010 4270 F029          00029    15         STC    7,SHIFT+3     MODIFY SHIFT INST WITH NO  OF BITS
000014 1367                        16         LCR    6,7           NO  OF BITS SHIFT FOR DECREMENT
000016 0670                        17         BCTR   7,0           N-1 FOR COMPARAND
000018 1B11                        18         SR     1,1           ZERO REG 1
00001A 4150 0020          00020    19 WORD    LA     5,32          LOAD A 32 TO REG 5 FOR COUNT
00001E 5013 0000          00000    20         ST     1,0(3)        ZERO STORAGE AREA
000022 5802 0000          00000    21 LOOP    L      0,0(2)        LOAD DATA TO REG 0
000026 8C00 0000          00000    22 SHIFT   SRDL   0,0           SHIFT DATA TO REG 1
00002A 1B00                        23         SR     0,0           TRUNCATE ON LEFT FOR MOD 2**N
00002C 8D00 5000 ?        00000    24         SLDL   0,0(5)        SHIFT BACK TO PROPER POSITION
000030 5603 0000          00000    25         O      0,0(3)        OR PACKED WORD TO REG 0
000034 5003 0000          00000    26         ST     0,0(3)        STORE BACK TO PACKED AREA
000038 4122 0004          00004    27         LA     2,4(2)        INCREMENT DATA ADDRESS
00003C 4640 F046          00046    28         BCT    4,NEXT        COUNT DOWN ON NO  OF ITEMS
000040 9827 D01C          0001C    29         LM     2,7,28(13)    RESTORE REGS
000044 07FE                        30         BR     14            RETURN
000046 8656 F022          00022    31 NEXT    BXH    5,6,LOOP      BRANCH BACK IF SPACE LEFT
00004A 4133 0004          00004    32         LA     3,4(3)        OTHERWISE INCREMENT STORAGE ADDRESS
00004E 47F0 F01A          0001A    33         B      WORD          AND CONTINUE
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01FEB69   4/23/70

                                   35 *       SUBROUTINE  UNPACK ( L, H, I, N )
                                   36 *
                                   37 *             THIS ROUTINE UNPACKS I WORDS OF DATA FROM THE H
                                   38 *             ARRAY TO THE L ARRAY   WORDS IN L ARE ZEROED AND N
                                   39 *             BITS ARE PLACED RIGHT JUSTIFIED FROM THE PACKED
                                   40 *             ARRAY H.
                                   41 *
                                   42         ENTRY  UNPACK
000052                             43         USING  *,15          USE REG 15 FOR BASE
000052 9027 D01C          0001C    44 UNPACK  STM    2,7,28(13)    SAVE REGS
000056 9825 1000          00000    45         LM     2,5,0(1)      LOAD ADDRESSES OF ARGUMENTS
00005A 5844 0000          00000    46         L      4,0(4)        I TO REG 4 - NO. OF ITEMS TO BE PACKED
00005E 5875 0000          00000    47         L      7,0(5)        N TO REG 7 - NO. OF BITS/ITEM
000062 4270 F025          00077    48         STC    7,LEFT+3      MODIFY SHIFT INST WITH NO  OF BITS
000066 1367                        49         LCR    6,7           NO  OF BITS SHIFT FOR DECREMENT
000068 0670                        50         BCTR   7,0           N-1 FOR COMPARAND
00006A 4150 0020          00020    51 DATA    LA     5,32          LOAD A 32 TO REG 5 FOR COUNT
00006E 5813 0000          00000    52         L      1,0(3)        LOAD PACKED DATA TO REG 1
000072 1B00                        53 BACK    SR     0,0           ZERO REG 0
000074 8D00 0000          00000    54 LEFT    SLDL   0,0           SHIFT H BITS TO REG 0
000078 5002 0000          00000    55         ST     0,0(2)        STORE IN L
00007C 4122 0004          00004    56         LA     2,4(2)        INCREMENT STORAGE ADDRESS
000080 4640 F038          0008A    57         BCT    4,MORE        COUNT DOWN ON NO. OF ITEMS
000084 9827 D01C          0001C    58         LM     2,7,28(13)    RESTORE REGS
000088 07FE                        59         BR     14            RETURN
00008A 8656 F020          00072    60 MORE    BXH    5,6,BACK      BRANCH BACK IF MORE DATA
00008E 4133 0004          00004    61         LA     3,4(3)        OTHERWISE INCREMENT DATA ADDRESS
000092 47F0 F018          0006A    62         B      DATA          AND CONTINUE
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01FEB69   4/23/70

                                    64 *        FUNCTION  ITEM ( M, I, N )
                                    65 *
                                    66 *              THIS ROUTINE RETRIEVES THE I TH ITEM FROM THE PACKED
                                    67 *              ARRAY M
                                    68 *
                                    69         ENTRY  ITEM
000096                              70         USING  *,15
000096 9025 D01C          0001C     71 ITEM    STM    2,5,28(13)     SAVE REGS
00009A 9824 1000          00000     72         LM     2,4,0(1)       LOAD ADDRESSES OF ARGS TO REGS 2,3,4.
00009E 5833 0000          00000     73         L      3,0(3)         LOAD I TO REG 3
0000A2 0630                         74         BCTR   3,0            SUBTRACT 1 FOR I-1
0000A4 4100 0020          00020     75         LA     0,32           LOAD A 32 TO REG 0
0000A8 8E00 0020          00020     76         SRDA   0,32           SHIFT TO REG 1
0000AC 5D04 0000          00000     77         D      0,0(4)         DIVIDE BY N
0000B0 5010 F04E          000E4     78         ST     1,TEMP         NO  OF ITEMS/WORD
0000B4 1803                         79         LR     0,3            I-1 TO REG 0
0000B6 8E00 0020          00020     80         SRDA   0,32           SHIFT TO REG 1
0000BA 5D00 F04E          000E4     81         D      0,TEMP         DIVIDE I-1 BY NO  ITEMS/WORD
0000BE 1851                         82         LR     5,1            SAVE IN REG 5 TO INDEX ARRAY M
0000C0 8850 0002          00002     83         SLA    5,2            MULTIPLY BY 4
0000C4 8E00 0020          00020     84         SRDA   0,32           REMAINDER TO REG 1
0000C8 5C04 0000          00000     85         M      0,0(4)         MULTIPLY BY N
0000CC 1831                         86         LR     3,1            LOAD TO REG 3 TO INDEX SHIFT
0000CE 5815 2000          00000     87         L      1,0(5,2)       LOAD DATA FROM M ARRAY
0000D2 8910 3000          00000     88         SLL    1,0(3)         LEFT ADJUST PROPER ITEM
0000D6 5844 0000          00000     89         L      4,0(4)         LOAD N TO REG 4
0000DA 8D00 4000          00000     90         SLDL   0,0(4)         SHIFT N BITS TO REG 0
0000DE 9825 D01C          0001C     91         LM     2,5,28(13)     RESTORE REGS
0000E2 07FE                         92         BR     14             RETURN
0000E4                              93 TEMP    DS     F
                                    94         END
```

CROSS-REFERENCE

```
SYMBOL    LEN  VALUE  DEFN     REFERENCES                                                4/23/70

BACK      00002 000072 0053     0060
DATA      00004 00006A 0051     0062
ITEM      00004 000096 0071     0069
LEFT      00004 000074 0054     0048
LOOP      00004 000022 0021     0031
MORE      00004 00008A 0060     0057
NEXT      00004 000046 0031     0028
PACK      00001 000000 0009
SHIFT     00004 000026 0022     0015
TEMP      00004 0000E4 0093     0078  0081
UNPACK    00004 000052 0044     0042
WORD      00004 00001A 0019     0033


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
  120 PRINTED LINES
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME HOX01PK(R)
****HOX01PK   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                              DATE  70 113/15 28 01

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
    ISN 0002              SUBROUTINE PADCST
                        C DETERMINE IF MORE THAN ONE PAD NEEDED AT EACH COMPLEX
                        C
    ISN 0003              REAL NPERPD,NPUSED
    ISN 0004              INTEGER*2 MATCH,JF,JL,NVEH,MORE,NSAVE,   NPAD,VEH,LETT,LYR,MIN,
                         1  NPFAM,NPSTG,NPINTL,NPINTU,KOUT,NONREC,LYD,NMULT
                        C   STORAGE FOR CHOOZ AND PADCST
    ISN 0005              COMMON/NONDIM/ NUMD,NSTG,MYRS,NFAM,NCI,NP,NV
    ISN 0006              COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
    ISN 0007              COMMON/STAGE/ LSA(40),SNR(41),STG(40),SUS(40),NYS(40),SUSLS(40,2),
                         1  NFS(40,4)
    ISN 0008              COMMON/MISSON/ LETT(250),LYR(250),LTR(50),YRLH(250),MIN(250),NM
    ISN 0009              COMMON/PADS/ PFAMD(30,5,3),PFAMS(30,5,3),NPFAM(30,5),
                         1  PSTGD(30,10,3),PSTGS(30,10,3),NPSTG(30,10),PINTS(30,5,3),
                         2  NPINTL(30,5),NPINTU(30,5),PAD(30)
    ISN 0010              COMMON/CHOOZE/ NEXT,GUESS,NPERPD(30),LP
    ISN 0011              COMMON/SAVEPT/MORE(50),KOUT(100),LOUT,EXTRA,SAVS(20)
    ISN 0012              COMMON/SAVELB/ KX,KI,NX,NCOST,K,KZ
    ISN 0013              COMMON/DSCOST/ DS(100),SUST(100),NYD(100),LYD(100),NONREC(120,20)
    ISN 0014              COMMON/VEHREC/ RECUR(60,20,2),NMULT(60,50),IFLAG
    ISN 0015              COMMON/SCRACH/ TDS(500),W2(500),Z(500),W(500),NSAVE(10),
                         1  NODE(5,500),LZ(100),MATCH(20),NPUSED(20),JF(20),JL(20),DUM(20),
                         2   NVEH(20,10),COST(2,250),WR(100)
                        C
    ISN 0016              WRITE (6,404)  NX,W(NX),TDS(NX),Z(NX)
                        C   MORE = NUMBER OF NODE WHICH HAS BEEN CONSIDERED AS OPT. SOLN.
    ISN 0017              DO 355 I = 1,50
    ISN 0018              IF (MORE(I).EQ.NX) GO TO 354
    ISN 0020              IF(MORE(I) .NE. 0)GO TO 355
    ISN 0022              MORE(I) = NX
    ISN 0023              GO TO 356
    ISN 0024             355 CONTINUE
    ISN 0025             357 WRITE (6,358)
    ISN 0026              GUESS = 0.0
    ISN 0027              GO TO 359
    ISN 0028             325 WRITE(6,401) NX,PAD(I),J
    ISN 0029              Z(NX) = 20.0E30
    ISN 0030              GO TO 55
    ISN 0031             356 EXTRA = 0.0
    ISN 0032              IF(NP.EQ.0)  GO TO 1
    ISN 0034              DO 320 I = 1,NP
    ISN 0035              DO 322 K = 1,MYRS
```

```
ISN 0036          NPUSED(K) = 0.0
ISN 0037          DO 322 J = 1,10
ISN 0038        322 NVEH(K,J) = 0
                C  COUNT NUMBER OF LAUNCHES REQUIRED PER PAD PER YEAR
ISN 0039          DO 321 J = 1,NM
ISN 0040          IF (YRLM(J) EQ 0.0) GO TO 321
ISN 0042          K = MIN(J)
ISN 0043          JA = LETT(J)
ISN 0044          L = LTR(JA)
ISN 0045          K1 = K + MIN0(0,ISIGN(NV,NV-K))
ISN 0046          IF (NPAD(L,K1).NE. 1) GO TO 321
ISN 0048          M = LYR(J)
ISN 0049          X = NMULT(K1,JA)
ISN 0050          X = YRLM(J)*X
ISN 0051          NPUSED(M) = NPUSED(M) + X
ISN 0052          DO 323 ME = 1,10
ISN 0053          IF(NVEH(M,ME) EQ K1) GO TO 321
ISN 0055          IF (NVEH(M,ME).NE.0) GO TO 323
ISN 0057          NVEH(M,ME) = K1
ISN 0058          GO TO 321
ISN 0059        323 CONTINUE
ISN 0060        321 CONTINUE
                C   SEE IF SECOND PAD IS REQUIRED AT ANY FACILITY
ISN 0061          DO 331 J = 1,20
ISN 0062        331 MATCH(J) = 0
ISN 0063          DO 324 J = 1,MYRS
ISN 0064          IF (NPUSED(J).GT 2.0*NPERPD(I)) GO TO 325
ISN 0066          IF (NPUSED(J) LE NPERPD(I)) GO TO 324
ISN 0068          DO 326 K = 1,10
ISN 0069          IF (NPSTG(I,K).EQ.0) GO TO 329
ISN 0071          IF (PSTGD(I,K,2) + PSTGS(I,K,2).LT  001) GO TO 326
ISN 0073          DO 327 L = 1,10
ISN 0074          IF (NVEH(J,L) EQ 0) GO TO 326
ISN 0076          LA = NVEH(J,L)
ISN 0077          DO 328 LB = 1,4
ISN 0078          IF(VEH(LB,LA) EQ 0) GO TO 327
ISN 0080          IF (VEH(LB,LA).NE NPSTG(I,K)) GO TO 328
ISN 0082          DO 330 LC = 1,20
ISN 0083          IF (MATCH(LC) EQ.0) GO TO 332
ISN 0085          IF (MATCH(LC) EQ K) GO TO 333
ISN 0087          GO TO 330
ISN 0088        332 MATCH(LC) = K

ISN 0089          JF(LC) = J
ISN 0090        333 JL(LC) = J
ISN 0091          GO TO 326
ISN 0092        330 CONTINUE
ISN 0093        328 CONTINUE
ISN 0094        327 CONTINUE
ISN 0095        326 CONTINUE
ISN 0096        329 DO 334 K = 1,5
ISN 0097          IF (NPFAM(I,K) EQ 0) GO TO 335
ISN 0099          IF (PFAMD(I,K,2) + PFAMS(I,K,2) LT. 001) GO TO 334
ISN 0101          DO 336 L = 1,10
ISN 0102          IF (NVEH(J,L) EQ 0) GO TO 334
ISN 0104          LA = NVEH(J,L)
ISN 0105          DO 337 LB = 1,4
ISN 0106          IF (VEH(LB,LA) EQ 0) GO TO 336
ISN 0108          LD = VEH(LB,LA)
ISN 0109          DO 338 LC = 1,4
ISN 0110          IF (NFS(LD,LC) EQ 0) GO TO 337
ISN 0112          IF (NFS(LD,LC) NE NPFAM(I,K)) GO TO 338
ISN 0114          K1 = -K
ISN 0115          DO 339 LE = 1,20
ISN 0116          IF (MATCH(LE) EQ 0) GO TO 340
ISN 0118          IF (MATCH(LE) EQ K1) GO TO 341
ISN 0120          GO TO 339
ISN 0121        340 MATCH(LE) = K1
ISN 0122          JF(LE) = J
ISN 0123        341 JL(LE) = J
ISN 0124          GO TO 334
ISN 0125        339 CONTINUE
ISN 0126        338 CONTINUE
ISN 0127        337 CONTINUE
ISN 0128        336 CONTINUE
ISN 0129        334 CONTINUE
ISN 0130        335 DO 342 K = 1,5
ISN 0131          IF (NPINTL(I,K).EQ 0) GO TO 324
ISN 0133          IF (PINTS(I,K,2).LT .001) GO TO 342
ISN 0135          DO 343 L = 1,10
ISN 0136          IF (NVEH(J,L).EQ.0) GO TO 342
ISN 0138          LA = NVEH(J,L)
ISN 0139          DO 344 LB = 1,3
ISN 0140          IF (VEH(LB+1,LA).EQ 0) GO TO 343
ISN 0142          LC = VEH(LB,LA)
```

```
ISN 0143            DO 345 LD = 1,4
ISN 0144            IF (NFS(LC,LD) EQ 0) GO TO 344
ISN 0146            IF (NFS(LC,LD) NE NPINTL(I,K)) GO TO 345
ISN 0148            LE = VFI(LD&I,LA)
ISN 0149            DO 346 LF = 1,4
ISN 0150            IF (NFS(LE,LF) EQ 0) GO TO 345
ISN 0152            IF (NFS(LE,LF) EQ NPINTU(I,K)) GO TO 347
ISN 0154          346 CONTINUE
ISN 0155            GO TO 345
ISN 0156          347 K1 = -100 - K
ISN 0157            DO 348 LG = 1,20
ISN 0158            IF (MATCH(LG) EQ 0) GO TO 349
ISN 0160            IF (MATCH(LG) EQ K1) GO TO 350
ISN 0162            GO TO 348
ISN 0163          349 MATCH(LG) = K1
ISN 0164            JF(LG) = J
ISN 0165          350 JL(LG) = J
ISN 0166            GO TO 342
ISN 0167          348 CONTINUE
ISN 0168          345 CONTINUE
ISN 0169          344 CONTINUE
ISN 0170          343 CONTINUE
ISN 0171          342 CONTINUE
ISN 0172          324 CONTINUE
                C     ADD EXTRA PAD COSTS ASSOCIATED WITH THIS SOLUTION
ISN 0173            DO 351 J = 1,20
ISN 0174            IF (MATCH(J) EQ 0) GO TO 320
ISN 0176            IF (MATCH(J) LT.-100) GO TO 352
ISN 0178            IF (MATCH(J) LT 0) GO TO 353
ISN 0180            K = MATCH(J)
ISN 0181            EXTRA = EXTRA & PSTGD(I,K,2) & PSTGS(I,K,2) * FLOAT(JL(J)-JF(J)&1)
ISN 0182            GO TO 351
ISN 0183          353 K = -MATCH(J)
ISN 0184            EXTRA = EXTRA & PFAMD(I,K,2) & PFAMS(I,K,2)*FLOAT(JL(J)-JF(J)&1)
ISN 0185            GO TO 351
ISN 0186          352 K = -MATCH(J) - 100
ISN 0187            EXTRA = EXTRA & PINTS(I,K,2) * FLOAT(JL(J)-JF(J)&1)
ISN 0188          351 CONTINUE
ISN 0189          320 CONTINUE
ISN 0190            1 WRITE (6,403) EXTRA
                C
                C   ADD PREVIOUSLY NEGLECTED SUSTAINING COSTS
```

```
ISN 0191            IF(LOUT EQ 0)  GO TO 11
ISN 0193            DO 10 I = 1,NUMO
ISN 0194            IF(KOUT(I) EQ 0.OR.LZ(I) EQ 0)  GO TO 10
ISN 0196            LZ(I) = 0
ISN 0197            DO 8 J = 1,NH
ISN 0198            LM = MIN(J)
ISN 0199            DO 6 K = 1,20
ISN 0200            IF(NONREC(LM,K) EQ 0)  GO TO 8
ISN 0202            IF(NONREC(LM,K) NE I)  GO TO 6
ISN 0204            LY = LYR(J)
ISN 0205            LZ(I) = MAXO(LZ(I),(LY-NYO(I)&1)/KI)
ISN 0206          6 CONTINUE
ISN 0207          8 CONTINUE
ISN 0208            XX = LZ(I)*KI
ISN 0209            EXTRA = EXTRA & XX*SAVS(LT)
ISN 0210            IF(DS(I) GE..5)  GO TO 10
ISN 0212            EXTRA = EXTRA & DS(I)
ISN 0213         10 CONTINUE
ISN 0214            CALL PACK(LZ,NODE(1,NX),NUMO,4)
ISN 0215            WRITE(6,405) EXTRA
ISN 0216         11 IF (EXTRA LT..001) GO TO 354
ISN 0218            TDS(NX) = TDS(NX) & EXTRA
ISN 0219            Z(NX) = Z(NX) & EXTRA
ISN 0220            GUESS = Z(NX)
ISN 0221            WRITE (6,210) GUESS
ISN 0222            GO TO 55
ISN 0223          354 WRITE(6,402) Z(NX)
ISN 0224          359 EXTRA = 0 0
ISN 0225            RETURN
ISN 0226           55 EXTRA = 100 0
ISN 0227            RETURN
ISN 0228          210 FORMAT (12H NEW GUESS =, F12.2)
ISN 0229          358 FORMAT(36HMORE THAN 50 NODES HAVE BEEN TESTED)
ISN 0230          401 FORMAT (26HPOSSIBLE SOLUTION AT NODE,I4,49H NOT FEASIBLE. MORE TH
                     1AN 2 PADS NEEDED AT COMPLEX,1X,A4,8H IN YEAR, I3)
ISN 0231          402 FORMAT(49HO   THIS ASSIGNMENT IS OPTIMUM AT A TOTAL COST OF,F12 2)
ISN 0232          403 FORMAT (18HOEXTRA PAD COSTS =, F10 2)
ISN 0233          404 FORMAT (1HO,13(1H*),19H POSSIBLE SOLUTION ,12(1H*)/1H ,13,29X,
                     1  3(F9 2,5X))
ISN 0234          405 FORMAT (31HOEXTRA PAD & SMALL SUST COSTS =, F10.2)
ISN 0235            END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,245761)              DEFAULT OPTION(S) USED
IEW0000     NAME MOX02PD(R)
IEW0461  PACK
IEW0461  IBCOM#
```

CROSS REFERENCE TABLE

| CONTROL SECTION NAME | ORIGIN | LENGTH |
|---|---|---|
| PADCST | 00 | 128C |
| NONDIM | 1290 | 1C |
| VEHCLE | 12B0 | 4B0 |
| STAGE | 1760 | 6E4 |
| MISSON | 1E48 | A90 |
| PADS | 28D8 | 378C |
| CHOOZE | 6068 | 84 |
| SAVEPT | 60F0 | 184 |
| SAVELB | 6278 | 18 |
| DSCOST | 6290 | 1838 |
| VEHREC | 7AC8 | 3CF4 |
| SCRACH | B7C0 | 53FC |

| ENTRY NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|

| LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION | LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION |
|---|---|---|---|---|---|
| 298 | NONDIM | NONDIM | 29C | VEHCLE | VEHCLE |
| 2A0 | STAGE | STAGE | 2A4 | MISSON | MISSON |
| 2A8 | PADS | PADS | 2AC | PADS | PADS |
| 2B0 | PADS | PADS | 2B4 | CHOOZE | CHOOZE |
| 2B8 | SAVEPT | SAVEPT | 2BC | SAVELB | SAVELB |
| 2C0 | DSCOST | DSCOST | 2C4 | VEHREC | VEHREC |
| 2C8 | VEHREC | VEHREC | 2CC | VEHREC | VEHREC |
| 2D0 | SCRACH | SCRACH | 2D4 | SCRACH | SCRACH |
| 2D8 | SCRACH | SCRACH | 2DC | PACK | $UNRESOLVED |
| 2E0 | IBCOM# | $UNRESOLVED | 1C4 | SCRACH | SCRACH |
| 1CC | NONDIM | NONDIM | | | |

ENTRY ADDRESS 00
TOTAL LENGTH 10BC0

```
****MOX02PD   NOW REPLACED IN DATA SET
```

DIAGNOSTIC MESSAGE DIRECTORY

IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL HAS SPECIFIED

```
(17)          05/360  FORTRAN H                                DATE  70 113/15 28.41

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002           SUBROUTINE PERF(N,M,VDES,WPL,VREF,PREF,WF,WI,ISP,THR,TSL,DIA,INCL,
                     *   ERR)
   ISN 0003           REAL ISP,INCL
   ISN 0004           INTEGER COUNT,ERR
   ISN 0005           DIMENSION WF(N),WI(N),ISP(N),THR(N)
   ISN 0006           DATA PI,G /3 141593,32 174/
   ISN 0007           K = N
   ISN 0008           KODE = 0
   ISN 0009           ERR = 0
   ISN 0010           V = VDES
   ISN 0011           WPL = 0
   ISN 0012           IF (PREF) 30,30,10
   ISN 0013        10 WPL = PREF
   ISN 0014        30 DO 100 COUNT = 1,50
   ISN 0015           W = WPL
   ISN 0016           DO 40 I=1,K
   ISN 0017        40 W = W&WF(I)&WI(I)
   ISN 0018           WLO = W
   ISN 0019           VIDL = 0
   ISN 0020           DO 50 I=1,K
   ISN 0021           WB = W-WF(I)
   ISN 0022           IF (WB) 45,45,48
   ISN 0023        45 ERR=2
   ISN 0024           GO TO 120
   ISN 0025        48 VIDL = VIDL&G*ISP(I)*ALOG(W/WB)
   ISN 0026        50 W = WB-WI(I)
   ISN 0027           IF (KODE) 90,55,90
   ISN 0028        55 IF (PREF) 70,70,60
   ISN 0029        60 VLOSS = VIDL-VREF
   ISN 0030           GO TO 97
   ISN 0031        70 TB = 0
   ISN 0032           DO 80 I=1,K
   ISN 0033        80 TB = TB&WF(I)*ISP(I)/THR(I)
   ISN 0034           WX = WB * EXP (VDES/ISP(K)/G)
   ISN 0035           TB = TB - (WX-WB)*ISP(K)/THR(K)
   ISN 0036           TOW = TSL/WLO
   ISN 0037           IF (TOW GT 1 5) TOW = 1.5
   ISN 0039           VLOSS = 6800 *(2.-TOW)  &  2800 *(THR(1)/TSL-1.)
                     *  & 5E6*PI*DIA**2/WLO&4 1* EXP(TB/125.)  -1530 *COS(INCL/57 296)
   ISN 0040        90 DELV = VIDL-VLOSS-VREF-V
   ISN 0041           WPL = WPL&(WPL&WI(K))*DELV/ISP(K)/G*(1 &WB/(WB&WF(K)))
   ISN 0042           IF (ABS(DELV)-1 ) 95,95,100

   ISN 0043        95 IF (KODE) 110,96,110
   ISN 0044        96 PREF = WPL
   ISN 0045        97 V = VDES
   ISN 0046           K = NEM
   ISN 0047           KODE = 1
   ISN 0048           WPL = 0.
   ISN 0049       100 CONTINUE
   ISN 0050           ERR = 3
   ISN 0051           GO TO 120
   ISN 0052       110 IF (WPL LT.0.) ERR = 1
   ISN 0054       120 RETURN
   ISN 0055           END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
        VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEH0000      NAME MOX02PF(R)
IEH0461  COS
IEH0461  EXP
IEH0461  ALOG

                                        CROSS REFERENCE TABLE

  CONTROL SECTION                    ENTRY
    NAME    ORIGIN  LENGTH             NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
  PERF          00     5CE

  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
     158            COS           $UNRESOLVED                15C            EXP           $UNRESOLVED
     160            ALOG          $UNRESOLVED
 ENTRY ADDRESS      00
 TOTAL LENGTH      5D0

****MOX02PF   NOW REPLACED IN DATA SET

                                        DIAGNOSTIC MESSAGE DIRECTORY

   IEH0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED
```

```
(17)          OS/360  FORTRAN H                              DATE  70 113/15 29 10

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002            SUBROUTINE PRINT
                  C    ***PRINT OUT DECISION COST CATEGORIES***
                  C
   ISN 0003        -    INTEGER*2 NONREC,LYD,NPFAM,NPSTG,NPINTL,NPINTU
   ISN 0004             COMMON/NONDIM/ NUMD,NSTG,MYRS,NFAM,NCI,NP,NV
   ISN 0005             COMMON/STAGE/ LSA(40),SNR(41),STG(40),SUS(40),NYS(40),SUSLS(40,2),
                       1  NFS(40,4)
   ISN 0006             COMMON/DSCOST/ DS(100),SUST(100),NYD(100),LYD(100),NONREC(120,20)
   ISN 0007             COMMON/INT/ NFML(40),NFMU(40),DINT(40),SINT(40),SINTLS(40,2)
   ISN 0008             COMMON/FAMILY/ XCDEF(30),FMNR(30),FMSUS(30),FMSLS(30,2),FAM(30)
   ISN 0009             COMMON/PADS/ PFAMD(30,5,3),PFAMS(30,5,3),NPFAM(30,5),
                       1  PSTGD(30,10,3),PSTGS(30,10,3),NPSTG(30,10),PINTS(30,5,3),
                       2  NPINTL(30,5),NPINTU(30,5),PAD(30)
   ISN 0010             COMMON/SCRACH/          MAT(100),LYF(30),NYF(30),MAF(30,3),
                       1  MAIC(40,3),MAPF(30,5),MAPS(30,10),MAPI(30,5),DUMMY(4045),
                       2  MAS(40,3), VM(2,120)
                  C
   ISN 0011             WRITE(6,211)
   ISN 0012             DO 925 I = 1,NUMD
   ISN 0013             J=MAT(I)
   ISN 0014             IF (J.LT -400) GO TO 9071
   ISN 0016             IF (J LT -300) GO TO 9072
   ISN 0018             IF (J.LT -200) GO TO 9018
   ISN 0020             IF(J LT -100) GO TO 345
   ISN 0022             IF(J LT 0) GO TO 340
   ISN 0024             DO 9073 K = 1,3
   ISN 0025             IF (MAS(J,K) NE I) GO TO 9073
   ISN 0027             IF(K EQ 1) WRITE(6,208) I,SNR(J),SUS(J),STG(J),NYD(I),LYD(I)
   ISN 0029             IF(K EQ.2) WRITE(6,9074) I,DS(I),SUST(I),STG(J),NYD(I),LYD(I)
   ISN 0031             IF(K.EQ 3) WRITE(6,9075) I,DS(I),SUST(I),STG(J),NYD(I),LYD(I)
   ISN 0033             GO TO 925
   ISN 0034          9073 CONTINUE
   ISN 0035           340 JX=-J
   ISN 0036             DO 9076 K = 1,3
   ISN 0037             IF (MAF(JX,K) NE I) GO TO 9076
   ISN 0039             IF (K EQ 1) WRITE(6,209) I,FMNR(JX),FMSUS(JX),FAM(JX),NYD(I),
                       1  LYD(I)
   ISN 0041             IF (K EQ 2) WRITE(6,9077) I,DS(I),SUST(I),FAM(JX),NYD(I),LYD(I)
   ISN 0043             IF (K EQ.3) WRITE(6,9078) I,DS(I),SUST(I),FAM(JX),NYD(I),LYD(I)
   ISN 0045             GO TO 925
   ISN 0046          9076 CONTINUE
   ISN 0047           345 JX=-J-100
```

```
ISN 0048           JY=NFNL(JX)
ISN 0049           JZ=NFNU(JX)
ISN 0050           DO 9079 K = 1,3
ISN 0051           IF (MAIC(JX,K) NE I) GO TO 9079
ISN 0053           IF (K EQ 1) WRITE(6,210)
                  1  I,DINT(JX),SINT(JX),FAM(JY),FAM(JZ),NYD(I),LYD(I)
ISN 0055           IF (K EQ 2) WRITE(6,9080)
                  1  I,DS(I),SUST(I),FAM(JY),FAM(JZ),NYD(I),LYD(I)
ISN 0057           IF (K EQ 3) WRITE(6,9081)
                  1  I,DS(I),SUST(I),FAM(JY),FAM(JZ),NYD(I),LYD(I)
ISN 0059           GO TO 925
ISN 0060       9079 CONTINUE
ISN 0061       9018 JX = -J - 200
ISN 0062           DO 9082 K = 1,5
ISN 0063           IF (MAPF(JX,K) NE.I) GO TO 9082
ISN 0065      .    KX = NPFAM(JX,K)
ISN 0066           WRITE(6,9083) I,DS(I),SUST(I),FAM(KX),PAD(JX),NYD(I),LYD(I)
ISN 0067           GO TO 925
ISN 0068       9082 CONTINUE
ISN 0069       9072 JX = -J - 300
ISN 0070           DO 9084 K = 1,10
ISN 0071           IF (MAPS(JX,K) NE.I) GO TO 9084
ISN 0073           KX = NPSTG(JX,K)
ISN 0074           WRITE(6,9085) I,DS(I),SUST(I),STG(KX),PAD(JX),NYD(I),LYD(I)
ISN 0075           GO TO 925
ISN 0076       9084 CONTINUE
ISN 0077       9071 JX = -J - 400
ISN 0078           DO 9086 K = 1,5
ISN 0079           IF (MAPI(JX,K) NE I) GO TO 9086
ISN 0081           KX = NPINTL(JX,K)
ISN 0082           KY = NPINTU(JX,K)
ISN 0083           WRITE(6,9087)I,DS(I),SUST(I),FAM(KX),FAM(KY),PAD(JX),NYD(I),LYD(I)
ISN 0084           GO TO 925
ISN 0085       9086 CONTINUE
ISN 0086        925 CONTINUE
ISN 0087           RETURN
ISN 0088        208 FORMAT (I4,6X,2F12.2,5X,A4,1X,14HSTAGE HARDWARE,29X,I3,9X,I3)
ISN 0089        209 FORMAT (I4,6X,2F12 2,5X,A4,1X,15HSHARED HARDWARE,28X,I3,9X,I3)
ISN 0090        210 FORMAT(I4,6X,2F12 2,5X,15HINTEGRATION OF ,A4,5H AND ,A4,
                   1  9H HARDWARE,11X,I3,9X,I3)
ISN 0091        211 FORMAT(25H1QUANTITIES BRANCHED UPON/1HO,11X,11HDEVELOPMENT,2X,
                   1  10HSUSTAINING,50X,10HYEAR AVAIL,2X,9HLAST YEAR//)

ISN 0092       9074 FORMAT (I4,6X,2F12.2,5X,A4,1X,9HSTAGE ETR,34X,I3,9X,I3)
ISN 0093       9075 FORMAT (I4,6X,2F12 2,5X,A4,1X,9HSTAGE WTR,34X,I3,9X,I3)
ISN 0094       9077 FORMAT (I4,6X,2F12 2,5X,A4,1X,10HSHARED ETR,33X,I3,9X,I3)
ISN 0095       9078 FORMAT (I4,6X,2F12 2,5X,A4,1X,10HSHARED WTR,33X,I3,9X,I3)
ISN 0096       9080 FORMAT (I4,6X,2F12.2,5X,15HINTEGRATION OF ,A4,5H AND ,A4,4H ETR,
                   1  16X,I3,9X,I3)
ISN 0097       9081 FORMAT (I4,6X,2F12.2,5X,15HINTEGRATION OF ,A4,5H AND ,A4,4H WTR,
                   1  16X,I3,9X,I3)
ISN 0098       9083 FORMAT (I4,6X,2F12 2,5X,A4,1X,14HSHARED AT PAD ,A4,25X,I3,9X,I3)
ISN 0099       9085 FORMAT (I4,6X,2F12 2,5X,A4,1X,13HSTAGE AT PAD ,A4,26X,I3,9X,I3)
ISN 0100       9087 FORMAT (I4,6X,2F12 2,5X,15HINTEGRATION OF ,A4,5H AND ,A4,1X,
                   1  7HAT PAD ,A4,8X,I3,9X,I3)
ISN 0101           END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL          DEFAULT OPTION(S) USED
        VARIABLE OPTIONS USED - SIZE=(126976,24576)
EW0000      NAME HOX02PR(R)
EW0461  IBCOM#
```

CROSS REFERENCE TABLE

| CONTROL SECTION NAME | ORIGIN | LENGTH | ENTRY NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|---|---|---|
| PRINT | 00 | 0BA | | | | | | | | |
| NONDIM | D90 | 1C | | | | | | | | |
| STAGE | DB0 | 6E4 | | | | | | | | |
| DSCOST | 1498 | 1838 | | | | | | | | |
| INT | 2CD0 | 3C0 | | | | | | | | |
| FAMILY | 3090 | 2D0 | | | | | | | | |
| PADS | 3360 | 378C | | | | | | | | |
| SCRACH | 6AF0 | 53FC | | | | | | | | |

| LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION | LOCATION | REFERS TO SYMBOL | IN CONTROL SECTION |
|---|---|---|---|---|---|
| 368 | NONDIM | NONDIM | 36C | STAGE | STAGE |
| 370 | DSCOST | DSCOST | 374 | INT | INT |
| 378 | FAMILY | FAMILY | 37C | PADS | PADS |
| 380 | PADS | PADS | 384 | PADS | PADS |
| 388 | SCRACH | SCRACH | 38C | SCRACH | SCRACH |
| 390 | IBCOM# | $UNRESOLVED | | | |
| ENTRY ADDRESS | 00 | | | | |
| TOTAL LENGTH | BEF0 | | | | |

```
****HOX02PR   NOW REPLACED IN DATA SET
```

DIAGNOSTIC MESSAGE DIRECTORY

```
IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL HAS SPECIFIED
```

```
FORTRAN IV G LEVEL 1, MOD 4            STGNUM            DATE = 70113        15/45/28

0001              SUBROUTINE STGNUM(IFLAG1)
                C COMPUTE NUMBER OF STAGES ACTUALLY USED AND THEIR RECURRING COSTS
0002              INTEGER*2 LETT,LYR,MIN,NPAD,VEH,NMULT,LYD,NONREC
                C STORAGE FOR MAIN, DECISN, STGNUM, CAPABL, AVAIL, AND CHOOZ
0003              COMMON/NONDIM/ NUMD,NSTG,AYRS,NFAM,NCI,NP,NV
                C STORAGE FOR MAIN, DECISN, STGNUM, CAPABL, AND AVAIL
0004              COMMON/VEHCLE/ NPAD(2,60),VEH(4,60),NYP(2,60)
                C STORAGE FOR MAIN, STGNUM, CAPABL, AVAIL, AND CHOOZ
0005              COMMON/MISSON/ LETT(250),LYR(250),LTR(50),YRLM(250),MIN(250),NM
                C STORAGE FOR MAIN, DECISN, STGNUM, AND AVAIL
0006              COMMON/STAGE/ LSA(40),SNR(41),STG(40),SUS(40),NYS(40),SUSLS(40,2),
                 1  NFS(40,4)
                C STORAGE FOR MAIN, DECISN, AND STGNUM
0007              COMMON/INT/ NFML(40),NFMU(40),DINT(40),SINT(40),SINTLS(40,2)
0008            C    STORAGE FOR MAIN, STGNUM, CHOOZ, AND CAPABL
0009              COMMON/VEHREC/ RECUR(60,20,2),NMULT(60,50),IFLAG
                C STORAGE FOR MAIN AND STGNUM
0010              COMMON/CAPMAT/ VNM(4,250),NV2
0011              COMMON/INTREC/ RINT(40),PLCINT(40)
0012              COMMON/STGREC/ NBY(40),MODE(40,3),SR(40,3),PLC(40,3),POJ(3),
                 1  SRJ(3,3)
0013              COMMON/DSCOST/ DS(100),SUST(100),NYD(100),LYD(100),NONREC(120,20)
0014              COMMON/REUSBL/ NU(50),UPP(50)
                C  STORAGE FOR STGNUM AND REUSE
0015              COMMON/NUMBER/ STGYTR(40,20,2),RINTYR(40,20)
                  COMMON/SCRACH/ STGYHM(40,20),STGMAX(40,20,2),RINTMX(40,20),
                 1  HSAVE(40),ISAVE(40), DUMMY(1735),MAS(40,3),DUM(160),NUS(40),
                 2  KLUE(40)
                C
0016              IF (IFLAG 6E-1) GO TO 621
                C     FIND MAX NUM OF EACH STAGE AND INTEGRATION POSSIBLE
0017              DO 661 I = 1,NSTG
0018              NUS(I) = 0
0019              DO 661 J=1,MYRS
0020              STGYTR(I,J,1) = 0.0
0021            661 STGYTR(I,J,2) = 0 0
0022              IF(NCI EQ 0)  GO TO 665
0023              DO 662 I = 1,NCI
0024              DO 662 J = 1,MYRS
0025            662 RINTYR(I,J) = 0 0
0026            665 DO 650 I = 1,NM
0027              IF(YRLM(I) EQ 0 0)   GO TO 650
0028              J = LYR(I)
```

```
FORTRAN IV G LEVEL 1, MOD 4            STGNUM            DATE = 70113        15/45/28

0029              JX = LETT(I)
0030              K = LTR(JX)
                C   MSAVE & ISAVE INDICATE IF THAT STAGE OR INTEGRATION HAS ALREADY BEEN
                C      COUNTED FOR MISSION NM
0031              DO 9003 IZ = 1,40
0032              MSAVE(IZ) = 0
0033             9003 ISAVE(IZ) = 0
0034              DO 650 IV = 1,NV2
0035              IF(ITEM(VNM(1,I),IV,1) EQ 0)  GO TO 650
0036              I1 = IV + MIN0(0,ISIGN(NV,NV-IV))
0037              X = NMULT(I1,JX)
0038              DO 649 MS = 1,4
0039              IA = VEH(MS,I1)
0040              IF(IA EQ.0)  GO TO 650
0041              IF(MSAVE(IA) EQ 1)  GO TO 644
0042              STGYTR(IA,J,K) = YRLM(I)*X + STGYTR(IA,J,K)
0043              MSAVE(IA) = 1
0044             644 IF(NCI EQ 0)  GO TO 649
0045              IF(MS EQ.4)  GO TO 650
0046              IF(VEH(MS+1,I1).EQ.0)  GO TO 650
0047              L1 = VEH(MS+1,I1)
0048              DO 645 MI = 1,NCI
0049              IF(ISAVE(MI) EQ.1)  GO TO 645
0050              DO 646 KY = 1,4
0051              IF(NFML(MI) NE NFS(IA,KY))  GO TO 646
0052              DO 647 KZ = 1,4
0053              IF(NFMU(MI) EQ NFS(L1,KZ))   GO TO 648
0054             647 CONTINUE
0055             646 CONTINUE
0056              GO TO 645
0057             648 RINTYR(MI,J) = RINTYR(MI,J) + YRLM(I)*X
0058              ISAVE(MI) = 1
0059             645 CONTINUE
0060             649 CONTINUE
0061             650 CONTINUE
0062              DO 668 I = 1,NSTG
0063              DO 668 J = 1,MYRS
0064             668 STGYHW(I,J) = STGYTR(I,J,1) + STGYTR(I,J,2)
0065              GO TO 673
                C DETERMINE NUMBER OF EACH STAGE AND INTEGRATION USED IN LAST ITERATION BY YEAR
0066             621 DO 623 I=1,NSTG
0067              DO 623 J=1,MYRS
0068              DO 623 K=1,2


FORTRAN IV G LEVEL 1, MOD 4            STGNUM            DATE = 70113        15/45/28

0069              STGMAX(I,J,K) = STGYTR(I,J,K)
0070             623 STGYTR(I,J,K) = 0.0
0071              IF (NCI.EQ 0) GO TO 9000
0072              DO 624 I=1,NCI
0073              DO 624 J=1,MYRS
0074              RINTMX(I,J) = RINTYR(I,J)
0075             624 RINTYR(I,J) = 0.0
0076            9000 DO 622 J=1,NM
0077              IF(YRLM(J) EQ 0.0) GO TO 622
0078              I1 = MIN(J)
0079              I = I1 + MIN0(0,ISIGN(NV,NV-I1))
0080              K = LYR(J)
0081              JX = LETT(J)
0082              ITR = LTR(JX)
0083              X = NMULT(I,JX)
0084              DO 625 MS = 1,4
0085              L = VEH(MS,I)
0086              IF (L.EQ 0) GO TO 622
0087              STGYTR(L,K,ITR) = STGYTR(L,K,ITR) + YRLM(J)*X
0088              IF (NCI EQ.0) GO TO 625
0089              IF (MS EQ.4) GO TO 622
0090              IF (VEH(MS+1,I).EQ 0) GO TO 622
0091              L1 = VEH(MS+1,I)
0092              DO 626 MI=1,NCI
0093              DO 627 KY=1,4
0094              IF(NFML(MI) NE NFS(L,KY)) GO TO 627
0095              DO 628 KZ = 1,4
0096              IF (NFMU(MI) EQ NFS(L1,KZ)) GO TO 629
0097             628 CONTINUE
0098             627 CONTINUE
0099              GO TO 626
0100             629 RINTYR(MI,K) = RINTYR(MI,K) + YRLM(J)*X
0101             626 CONTINUE
0102             625 CONTINUE
0103             622 CONTINUE
0104              IF (NCI EQ 0) GO TO 9001
0105              DO 691 I=1,NCI
0106              DO 691 J=1,MYRS
0107             691 IF(RINTYR(I,J) EQ 0.0)RINTYR(I,J) = RINTMX(I,J)
0108            9001 DO 676 I = 1,NSTG
0109              DO 676 J = 1,MYRS
0110              DO 676 K = 1,2
0111              IF(ABS(STGYTR(I,J,K) - STGMAX(I,J,K)).GT 0.001.AND STGYTR(I,J,K).
```

```
FORTRAN IV G LEVEL 1, MOD 4            STGNUM            DATE = 70113          15/45/28

               1GT 0 001)   GO TO 677
0112          676 CONTINUE
0113              WRITE(6,4101)
0114         4101 FORMAT (1HO,4X, 40HTHE OPTIMUM SOLUTION HAS BEEN DETERMINED)
0115              IFLAG1 = 0
0116              RETURN
             C    DETERMINE HARDWARE COSTS BY YEAR BASED ON LAST ITERATION
0117          677 DO 8013 I = 1,NSTG
0118              KLUE(I) = 0
0119              DO 8014 J = 1,MYRS
0120              IF(STGYTR(I,J,1) GT 0 01 OR STGYTR(I,J,2).GT. 01)  KLUE(I) = I
0121              STGYHW(I,J) = STGYTR(I,J,1) + STGYTR(I,J,2)
0122         8014 CONTINUE
0123              IF(KLUE(I) EQ.1)  GO TO 8013
0124              DO 8016 J = 1,MYRS
0125              STGYTR(I,J,1) = STGMAX(I,J,1)
0126              STGYTR(I,J,2) = STGMAX(I,J,2)
0127         8016 STGYHW(I,J) = STGYTR(I,J,1) + STGYTR(I,J,2)
0128         8013 CONTINUE
             C    ADD INITIAL REUSABLE PURCHASE PRICE TO DEV. COST DS
0129          673 DO 710 I = 1,NSTG
0130              IF(NU(I).EQ.0)  GO TO 710
0131              IF(IFLAG GE 1 AND NU(I) GE 0)  GO TO 710
0132              IF(IFLAG EQ 0)  GO TO 709
0133              NUS(I) = NU(I)
             C    THE FOLLOWING CARD IS USED IF THE PROGRAM ITERATES ON INITIAL QUANTITY
             C     TO BE PURCHASED
             C      CALL REUSE
0134          709 NI = MAS(I,1)
0135              X = IABS(NU(I))
0136              Y = NUS(I)
0137              DS(NI) = DS(NI) + (X+Y) * UPP(I)
0138          710 CONTINUE
             C      MAKE ADJUSTMENT FOR BATCHING OVER YEARS
0139              DO 663 I = 1,NSTG
0140              IF (NBY(I) EQ.1) GO TO 663
0141              IA = 2
0142              IB = NBY(I)
0143              IC = 1
0144          666 DO 664 J= IA,IB
0145              IF (J GT MYRS) GO TO 700
0146          664 STGYHW(I,IC) = STGYHW(I,IC) + STGYHW(I,J)
0147          700 DO 667 J = IA,IB


FORTRAN IV G LEVEL 1, MOD 4            STGNUM            DATE = 70113          15/45/28

0148              IF (J GT MYRS) GO TO 663
0149          667 STGYHW(I,J) = STGYHW(I,IC)
0150              IA = IA + NBY(I)
0151              IB = IB + NBY(I)
0152              IC = IC + NBY(I)
0153              GO TO 666
0154          663 CONTINUE
0155              DO 9002 I = 1,NSTG
0156              IF(KLUE(I) EQ 0)  GO TO 9002
0157              DO 9002 J = 1,MYRS
0158              IF(STGYTR(I,J,1) LT..01) STGYTR(I,J,1) = STGMAX(I,J,1)
0159              IF (STGYTR(I,J,2).LT..01) STGYTR(I,J,2) = STGMAX(I,J,2)
0160              IF(STGYHW(I,J) LT..01)  STGYHW(I,J) = STGYTR(I,J,1)+STGYTR(I,J,2)
0161         9002 CONTINUE
             C    DETERMINE VEHICLE RECURRING COSTS BY YEAR AND LAUNCH SITE
0162              DO 632 I=1,NV
0163              DO 635 J=1,MYRS
0164              RECUR(I,J,1) = 0 0
0165          635 RECUR(I,J,2) = 0 0
0166              DO 633 MS = 1,4
0167              K = VEH(MS,I)
0168              IF (K EQ 0) GO TO 632
0169              DO 634 J= I,MYRS
0170              IF (STGYHW(K,J).LT 0 001) GO TO 634
0171              IF (MODE(K,1).NE 0) GO TO 8015
0172              HDWR = SR(K,1) * STGYHW(K,J) ** PLC(K,1)
0173              GO TO 8010
0174         8015 LX = MODE(K,1)
0175              IF (STGYHW(K,J).LE POJ(LX)) HDWR = SRJ(LX,1)/STGYHW(K,J)
0176              IF (STGYHW(K,J) GT.POJ(LX)) HDWR = SRJ(LX,2)+SRJ(LX,3)/STGYHW(K,J)
0177         8010 DO 692 L = 1,2
0178              IF(STGYTR(K,J,L) EQ 0 0) GO TO 692
0179              M = L + 1
0180              IF (MODE(K,M) NE 0) GO TO 8011
0181              RECUR(I,J,L)=RECUR(I,J,L)+HDWR+SR(K,M)*STGYTR(K,J,L)**PLC(K,M)
0182              GO TO 692
0183         8011 LX = MODE(K,M)
0184              IF(STGYTR(K,J ,L) LE.POJ(LX)) RECUR(I,J ,L) = RECUR(I,J ,L)
                1  + SRJ(LX,1)/STGYTR(K,J ,L) + HDWR
0185              IF (STGYTR(K,J ,L) GT POJ(LX)) RECUR(I,J ,L) = RECUR(I,J ,L) +
                1         SRJ(LX,2) + SRJ(LX,3)/STGYTR(K,J ,L) + HDWR
0186          692 CONTINUE
0187          634 CONTINUE
```

```
0188            IF (NCI EQ.0) GO TO 633
0189               IF (MS EQ 4) GO TO 632
0190               IF(VEH(MS+1,I) EQ 0) GO TO 632
0191               K1 = VEH(MS+1,I)
0192               DO 636 L=1,NCI
0193               DO 637 KY=1,4
0194               IF (NFML(L) NE NFS(K,KY)) GO TO 637
0195               DO 638 KZ = 1,4
0196               IF (NFMU(L) EQ NFS(K1,KZ)) GO TO 639
0197            638 CONTINUE
0198            637 CONTINUE
0199               GO TO 636
0200            639 DO 640 J = 1,MYRS
0201               IF(RINTYR(L,J) EQ.0 0) GO TO 640
0202               HDWR = RINT(L)*RINTYR(L,J)**PLCINT(L)
0203               RECUR(I,J,1) = RECUR(I,J,1) + HDWR
0204               RECUR(I,J,2) = RECUR(I,J,2) + HDWR
0205            640 CONTINUE
0206            636 CONTINUE
0207            633 CONTINUE
0208            632 CONTINUE
0209               IFLAG1 = 1
0210               RETURN
0211               END
```

FORTRAN IV G LEVEL 1, MOD 4 STGNUM DATE = 70113 15/45/28

TOTAL MEMORY REQUIREMENTS 001B74 BYTES

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000     NAME MOX02SN(R)
IEW0461   ITEN
IEW0461   IBCOM=
IEW0461   FRXPR=
IEW0461   MIN0
```

MODULE MAP

| CONTROL SECTION | | | ENTRY | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| NAME | ORIGIN | LENGTH | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
| STGNUM | 00 | 1B74 | | | | | | | | |
| NONDIM | 1B78 | 1C | | | | | | | | |
| VEHCLE | 1B98 | 4B0 | | | | | | | | |
| MISSON | 2040 | A90 | | | | | | | | |
| STAGE | 2AD8 | 6E4 | | | | | | | | |
| INT | 31C0 | 3C0 | | | | | | | | |
| VEHREC | 3580 | 3CF4 | | | | | | | | |
| CAPMAT | 7278 | FA4 | | | | | | | | |
| INTREC | 8220 | 140 | | | | | | | | |
| STGREC | 8360 | 670 | | | | | | | | |
| DSCOST | 89D0 | 1838 | | | | | | | | |
| REUSBL | A208 | 190 | | | | | | | | |
| NUMBER | A398 | 2580 | | | | | | | | |
| SCRACH | C918 | 53FC | | | | | | | | |

ENTRY ADDRESS 00
TOTAL LENGTH 11D18

****MOX02SN NOW REPLACED IN DATA SET

DIAGNOSTIC MESSAGE DIRECTORY

IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED

NASA
FORMAL
REPORT