AQPLUS: An Adaptive Random Search Method for Selecting a Best Set of Attributes from a Large Collection of Candidates

Scott Forsburg

1976

Internal Report

Department of Computer Science
University of Illinois
Urbana, Illinois

SYSTEM AQPLUS

This document describes AQPLUS, an interactive, iterative system for inductive inference. AQPLUS is built around an algorithm based on random search with adaptation, which is due to Lbov. The system is written in Fortran for use on the DEC-10, and uses the inductive program AQFOR, written by Jim Larson, which is a Fortran implementation of the algorithm AQ. The reader is referred to the documentation of AQ.FOR for information about that segment of the system. Familiarity with the algorithm is assumed.

The algorithm AQPLUS associates with each feature X an information-content value U(X) in [0,1]. A certain number of features are chosen from the feature set, using a random selection process in which the relative probubility of feature X being chosen is equal to U(X). AQ.FOR is then run on the chosen features, and the information-content values for the chosen features are modified according to the results of the AQ.FOR run.

It should be noted that when the number of description features is reduced that it is possible to have two events from different classes having the same description. In this case it is said that the representation of the events is non-disjoint. If this occurs, the user is notified.

If the number of features that we are working with is DIMTOT, and if NDIM is the number of features used by AQ.FOR, the modification of the various features should depend on the relative values of DIMTOT and NDIM, and on the value of C, which is a real number between zero and one which is a measure of the 'goodness' of the formula produced by AQ.FOR.

In general, the better the rule (the greater the C), the greater should be the increase of the U(X) for those X used in the rule, and the less should be the decrease of the U(X) for those X (among the chosen features) which are not used in the rules.

Also, as NDIM → DIMTOT, the more extreme should be the modification of the information-content values.

To satisfy the above conditions, the following function can be used to dodify the U(X) corresponding to those X's among the chosen features:

For X used in the rule:

$$U(X) \leftarrow U(X) + (1-U(X)) \frac{C \times NDIM}{DIMTOT}$$

For X not used in the rule:

$$U(X) \leftarrow U(X) \times C \frac{DIMTOT-NDIM}{DIMTOT}$$

In addition to the code for the algorithm itself, supporting routines have been written which allow the user to examine the progress of AQ.For, and alter various prameters. These routines are described below:

Routine AQPARM: Allows the user to alter the parameters used by AQ.For. The entry format is 'parmname:value*'. The parameters are:

DIMTOT (≤200) = The total number of features known to the system.

NCT = The total number of events in file in file 'DATA.TOT'.

NDIM (≤DIMTOT)= The number of features to be submitted to AQVAL.

C1 = The number of the class to be covered.

IB = The number of the first event in C1.

NC1 = The number of events in C1.

NUSE1 (≤NC1)= The number of events in C1 to be used by AQ.FOR.

RAND = If 1, then the events from C1 to be used by AQ.FOR will be chosen randomly. If 0, then not.

NC2 = The number of events not in C1 to be used by AQ.FOR.

MAXSTR = The maximum number of complexes in an intermediate star.

NOSEC = IF 1, uses the number of events in C1 covered as an additional criterion for complex selection. If 0, then not.

Routine FORCE: Allows the user to select features to be used by AQVAL. He may choose as many NDIM features. If he chooses less than NDIM features, the remainder will be chosen randomly by the subroutine 'SELECT'. Feature I is 'forced' if FORC(I) = 1, otherwise FORC(I) = 0.

Routine MODIFY: Allows the user to view or modify the information content values of the features. In alter/display mode, a feature number is entered, and the usefulness value is displayed. The user can then either enter a new value, or hit return to select another feature. Hitting return when a feature number is requested displays the usefulness value of the next feature. Altering or displaying feature number 0 alters or displays (respectively) the values for all features.

Routine CREATE: Allows the user to create new features from combinations of existing features. (Not yet implemented.)

Other features used by the system:

Routine GETINT: Accepts and checks numeric data from the user. The resulting Packed-Ascii data is availiable in array M. If M(1) = 'E', then the value accepted was non-numeric.

Routine Select (N):
Chooses N features 'randomly' weighted by the usefulness values of the features. These choices, along with the forced features are put in NSE1 (1) through NSEL (NDIM).

Routine PROJEC: Creates the file 'JIM.DAT' by projecting the events from 'DATA.TOT' onto the selected features.

Routine USEFUL: Alters the usefulness values of the features used in the last run or AQ.FOR. Calculations made:

QUOT = (# of events in C1 classified correctly - # of events classified incorrectly)/ NC1. Then $C = \sqrt{prev}/C$.

The routine also stores the formula in the array PUT: Put (*, last) contains the best formula so far in the form:

Put(1,last)= cost
Put (2, last)= class
Put (3, last)= -1

Then groups of three elements correspond to selectors, with feature number, lower bound, and upper bound. Terms are separated by -1, and -2 designat the end of the formula. The resulting formula, along with old and new information content values are computed using the formulas listed in the description of the algorithm.

Routine PRBEST: Prints the best formula found so far from Put (*, last).

Most of the data areas in the common area are only used by AQ.FOR and its supporting routines, and are explained in the AQ.FOR documentation. Others are:

U(I) = Contains the current value of feature I

FORC(I) = 1 if the user has selected feature I, 0 otherwise.

NFORC = The number of features selected by the user.

PUT (*,last)= Contains the best formula found so far, in the format described in the write-up of the subroutine 'USEFUL'.

NSEL = Is used in selecting the features to be used by AQ.FOR.

PREVC1 = Contains the last value of C1.

PREVC = Contains the best value of C found so far.

DISJNT = Is a flag which is set to 0 by AQ.FOR if the event representation is found to be non-disjoint.

SUGGESTIONS MADE BY PROFESSOR MICHALSKI:

1. List features along with information content values.
2. Print event sets .
3. Choice for cost, usefulness formulas.
4. Temporary storage of usefulness values so other usefulness values may be used.
5. Choice of constant or random selection of events in C2.

1. Introdubtion

This paper describes a sequence of special purpose programs which generate VL_1 formulas using algorithm AQ and test their performance against a set of testing events with emphasis on a large event space, and economical generation of covers for many classes using unquantized real valued variables. The algorithm itself is implemented in Fortran on the DEC - 10 computer and contains the potential for user interaction with the program. The formulas generated by this program can be sent to a PL/1 program on the IBM 360/75 to be printed in a readible format and tested against a set of testing events. A confusion matrix can be generated.

It is assumed that the reader has a basic knowledge of the algorithm AQ and the VVL system VL_1. Although these concepts are not explained here, the modification and simplifications which are made to the algorithm are described in detail.

The fortran program AQ.FOR contains approximately 450 Fortran statements, and occupies between 40 and 60 pages of DEC - 10 memory when executing (each page contains 1000 [octal] 36 bit words). This is equivalent to 100K - 150K bytes on the 360/75. The two PL/1 programs require between 100 and 180K bytes of storage to run on the 360/75 and consist of about 100 to 200 PL/1 statements each.

Section 2 of this paper describes the algorithm simplification used in the fortran program. Sections 3 and 4 contain descriptions of the use of the program AQ.FOR and the program structure. Section 5 describes the PL/1 testing program TCPXIS and its use. Section 6 describes the confusion matrix program PBLOG. The three program listings are given in appendices A, B, and C.

2. Algorithm Description and Simplifications

The program AQ..FOR is an implementation of the algorithm AQ with emphasis on handling a large event space with real valued variables economically. The program contains some restricitons on the basic algorithm in that only interval variables are accepted and only 3 cost functions are currently implemented with a tolerance of 0. Also, all variable values must be integers between 1 and 9999. Only a general star size is maintained instead of the maxstar-cutstar prodedure which is available in other programs. Initial comparisons with the program AQVAL-v.7 (AQ7), however, indicate a 5 fold reduction in time and a 10 fold reduction in cost.

The basic algorithm AQ in unchanged from that described in paper. The sets F^i are to be covered are selected in IC mode only. For each set, the algorithm maintains two sets of events, those events which have not been covered by any previous star (the set E_p), and those events which have not been covered by a prefious L_q or output complex (the set E_1). Events from the set E_p are selected inturn and stars generated against a random sampling of events from other events sets. The number of these events is controlled by an input parameter. (NC2) A new intermediate star is generated only if the randomly selected event is contained in the current intermediate star. The number of complexes in each intermediate star is maintained at a constant number of complexes (MAXSTR). If the number of complexes exceeds MAXSTR, then the best MAXSTR complexes are chosen with respect to the following 3 criteria:

Criterion #	Description
1	largest number of new events covered by the complex of the set F^i not covered by any previous output complexes
2	fewest number of literals in the complex
3	(optional) - the largest number of events from the original event set F^i which are covered by the complex

This selection process is designed for a MAXSTR value of 1 to increase efficiency. If an event from F^i is identical to an event from another event set, the program halts and prints a message indicating those two events which were identical and that an intermediate star was 'wiped out'.

In addition, unspecified values for some variables may be handled by assuming that the value for the variable is the entire range of values for that variable (i.e. the program ignores unspecified variable values in star formation). These variables are coded as the value 0 in the event set data.

3. How to Use the Program AQ.FOR

This section describes the input sequence and recommends optimal parameters to attain the best VL1 formulas ~~economically~~. To execute the program, some preprocessing of the event data is necessary to create the proper binary mode input. The program reads the events from a binary mode random access file called DATA.JEM. This file contains the entire event set where each event is preceeded by the

number of the set F^i to which it belongs. For example, with 50 variables and 1200 events, this is a 1200 X 51 matrix of values. The program REFORM. will accept input for AQVAL - v7 and create the proper input for AQ.FOR as well as accept formatted data (which is more economical to store).

In addition, a file FSIZE which contains the size of each event set (except the first) is necessary. Each successive line of this file should contain the size of the next event set (numbered 2, 3, etc.).

The output appears on the terminal in readable form and also in a file FORM.DAT in a format suitable for subsequent processing by the PL/1 programs.

3.1 Input Specification

Reformat the data if it has not been already done by following the steps in table 1. Then, form VL_1 formulas using the steps in table 2. Before proceeding with table 1, copy the events into a file named EVENT.DAT.

Generate Covers

Step	Terminal Oupput	User Response Example	Explanation
1	.	RUN AQ	Execute the program which generates covers
2	ENTER NUMBER OF DIMENSIONS	50	Enter the number of variables (or dimensions)
3	HOW MANY EVENTS TOTAL	1942	Enter the total number of events in the entire set F
4	ENTER MAXSTAR AND NOSEC I2	∅1∅∅	(∅ means enter blank) Enter the maxstar parameter in I2 format and then enter 0 if the third criterion (total events of the events set covered by a complex) is to be used, enter 1 otherwise in I2 format.
5	ENTER CLASS TO BE COVERED I2	1	Enter the first class number which is to be covered (1 unless this is a restart)
6	HOW MANY EVENTS OF C2	200	Enter the number of events which are to be randomly chosen against which covers are to be generated.
7	ENTER BEGINNING AND SIZE OF CLASS I4,I3	∅∅∅1∅98	Enter the event number which begins the set specified in step 5 and the number of events in this set. The program now generates covers for the sets specified the file FSIZE generated by the program REGORM and prints the covers on the terminal.
8	EXIT		Normal program termination. The covers are now in the file FORM.DAT

Reformat Data

Step	Terminal output	User Response Examples	Explanation
1	.	RUN REFORM	Execute the program which reformats the event data
2	ENTER FORMAT	CR or (3I2,1X,4I2)	Enter the Fortran format in parenthesis if the data is formatted in the file EVENT, otherwise, press CR if it is in list format.
3	ENTER # OF VARIABLES AND # OF EVENTS SETS	7 3	Enter the number of variables and the number of event sets or classes
4	HOW MANY EVENTS IN SET 1	21	Enter the number of events in the set requested.
5	HOW MANY EVENTS ARE TO BE COVERED	13	Enter the number of events which are to be randomly selected to be covered. Go to step 4 until all event set sizes have been entered
6	EXIT		The program has completed execution and the file DATA.JIM now contains the internal form of the event data. A file FSIZE contains the number of events to be covered.

Should it be necessary to restart the program, first copy the useable information out of the file FORM.DAT, then build the file FSIZE with the sizes of the sets and which remain to be covered and execute the steps 1 to 7 of table 2.

3.2 File Structures

FSIZE This is a file containing the size of each event set to be covered (except for the first set which is specified on execution), and the number of events which are actually to be covered. Each line of the file corresponds to one event set. Each line contains one or two numbers in I3 format. The first number indicates how many events of the event set are to actually be covered. The second number is specified only if a subset of the original event set is to actually be covered. In this instance, the second number indicates the total number of events in the events.

Example:
```
b45113        (b = space)
200
300400
```

In this example, the second set to be covered contains 113 events but only 45 are to be actually covered bythe program. The third set to be covered contains 200 events all of which are to be covered. The fourth set contains 400 events, 300 of which are to be covered (Note - the first set to be covered is specified in step 7 of table 2)

FORM.DAT - This file contains the formulas generated by the program AQ.FOR in internal format accepted by the follow-up PL/1 programs. The file consists of a list of formulas, each of which covers one event set (or part of an event set as specified by the file FSIZE). Each such formula is terminated by a -1 in columns 1 and 2. Each such formula is composed of a set of complexes each of which is terminated by a 0. Finally, each complex is a list of literals containing the variable number, the lower bound and the upper bound for the literal. Unrestricted lower and upper bounds are represented as 0 and 10000 respectively.

AQ.FOR FORTRAN source for the program AQ.FOR

AQ.SAV Save module (load module) for the program AQ.FOR

REFORM.FOR Fortran source for the program REFORM

REFORM.SAV Save module for the program REFORM

DATA.JIM Random access file containing a list of events in internal (binary, random access) form. Each event is represented by a list of the variable values and is proceeded by the number i of the events set F^i to which it belongs.

EVENT.DAT File containing event sets in raw form.

4 Program Structure AQ.FOR

The Fortran program AQ.FOR generates VL_1 covers of event set F^i against randomly selected events from other sets F/F^i. The basic files and their structures are described in section 3.2. The program and the data structures within the program are described here.

4.1 Program Data Structures

Event representation -- Events which are to be covered are stored in the array EVENT. Each row represents one event. Each element of a row represents a variable value for that event. The size of this array may be altered to better fit the event set which is to be covered by program modification.

Event set representation -- The sets E_1 and E_2 are represented by the arrays FQ and FP respectively. Each of these arrays contains a pointer to the location in EVENT for each element of the corresponding sets. When an event is removed from one of the sets, the corresponding element of the array is set to 0. An event against which covers are to be formed is stored in an array EVE.

Complex representation -- Each complex is represented by a list of selectors. Each selector contains information about the variable (DIM), the lower and upper bounds (XLB and UB) and a pointer to the next selector (SPTR).. Each complex contains information about the weight of the complex (COST) encoded as (NDIM) * (# new events covered) — (the number of selectors), the number of events covered by the complexx (SEC) and a pointer

to the next complex in the star (PTR). In addition, for garbage collection, each selector node (shown below) contains a mark bit (MARK).

Example:

Node (3)

DIM	PTR
3	20
XLB	UB
3	6
MARK	SPTR
0	15
COST	SEC
113	200

Node (15)

DIM	PTR
2	-
XLB	UB
5	10000
MARK	SPTR
1	0
COST	SEC
-	-

Node (20)

DIM	PTR
25	0
XLB	UB
0	10
MARK	SPTR
1	0
COST	SEC
103	120

VL_1 formula represented above:

$$(3 <= X3 <= 6) \wedge (5 <= X2) \vee (X25 <= 10)$$

Star representation -- Each star is represented by a head pointer and a list of complexes in order of decreasing COST and SEC values. Two subh lists are maintained; the list representing the current intermediate star under construction (with head GLIST*), and the previous intermediate star (with head pointer OLIST). A list of free selector nodes is also maintained with head pointer FREE.

Selector storage maintanence -- Whenever a selector is created, a selector element is obtained from the free list of elements. If this free list should be exhausted, a garbage collection routine is invoked which marks all nodes which are currently in use and stores the remainder in the free list. If more than 90% of the space is actually in use, a message to that effect is printed.

* GLIST contains a dummy first element for better insertion algorithm.

4.2 Program Structure

The program AQ.FOR consists of several routines each of which is briefly described below. Refer to the program listing given in appendix A for further details.

MAIN -- The program AQ.FOR begins with a driving program or main program which selects events which are to be covered based on user input or the data in the file FSIZE. Events are either randomly selected (if only a subset of event set F^1 is to be covered) or sequentially (if the entire event set F^1 is to be covered.

AQ -- This subroutine is the heart of the algorthm. The process proceeds as follows: An event is selected from the set E_p (FP array) or if this is empty, from the set E_1 (the FQ array). Events are then selected from the event space $F = U F^1$ at random. If an event from the event set which is to be covered is selected, from the set F, it is discarded and a new event is picked again at random. Now, for each variable a selector is generated, multiplied by the old intermediate star (pointed to by OLIST), and inserted into the new intermediate star (see the subroutine INSERT).. when the intermediate star is complete variable CLIST is set to point to the current intermediate star and a new event is selected from the event set F as before. If this event is not contained in the current star, it is included in the total number of events of F (i.e. the events against which a cover is to be formed) but no elementary star is formed against this event and no multiplication takes place since it would not change the current intermediate star anyway. Once the correct number of events of F have been used, the complex which is selected as the best of the current star is typed on the terminal and stored in the file FORM.DAT using the subroutine PRT. The arrays FQ and FP are updated and a new event is selected from FP or FQ. When all events of E_1 have been covered, the subroutine returns control to the main program.

PRT(R) -- This subroutine prints the complex which is passed as an argument (R) on the terminal and stores the complex in the file FORM.DAT in the proper format.

CF(R) -- This routine calculates the weight of the complex (R) and returns this value. The weight of a complex is: # of events not previously covered by a complex but covered by this complex times the number of variables in the event space minus the number of selectors in the complex. CF also maintains the total number of events covered if the switch NOSEC is off.

5. TCPXLS - Program to list complexes generated by AQ.FOR

This PL/1 program was written as a special purpose program to list the complexes generated by AQ.FOR. The complexes and the original event sets are read into this program, a listing of the complexes in readible form, [illegible] statistics [illegible] and a restricted confusion matrix is printed. The program structure and the algorithms used to compute the statistics and the confusion matrix are given in section 5.1. A description of the input and modifications necessary to handle different problems is given in section 5.2. Section 5.3 gives an example of the output and describes the proper interpretation of the output. Section 5.4 gives a grief description of the data structures within the program. A complete program listing appears in appendix B.

5.1 The input to the algorithm implemented in this program TCPXLS includes a set of complexes generated by the program AQ.FOR, the set of training events (or learning events) used to create these complexes and a set of testing events. For each complex, covering part of the event set F^i, a trimmed complex TC is created which has the following properties:

i the trimmed complex contains the same events of F^i as the original complex C

ii the range of each of the selectors of C is reduced as much as possible.

The statistics about the complexes are determined as follows:

COV -- the number of events of the set F^i covered by the complex (i.e. the events associated with the current decision class)

NEW -- the number of events of the set F^i not covered by any previous complex but covered by this complex C,

IND -- the number of events of the set F^i not covered by any other complex but covered by the complex C,

TOT -- the number of events of F^i ocvered by the complex and all provious complexes,

Following these calculations, the complexws are printed in a readable form, and finally, the confusion matrix is formed. Approximately the same procedure described in the CONFUS manual is used with the following modifications:

i The definiton of the degree of fit of a selector to a variable value (DS) is changed to be following:

$$DS = \begin{cases} 1 & \text{- if the value is satisfied by the selector of the trimmed complex} \\ .5 & \text{- if the value is satisfied only by the selectors of the original complex} \\ 0 & \text{- otherwise} \end{cases}$$

ii The unspecified values are simply ignored in the calculation of DT (degree of fit of an event to a term).

5.2 Input Specification

i Formula input - This is the file FORM.DAT created by AQ.FOR,

ii a data file LEARN which contains first a list of the number events ineach event set followdng by the actual events in a fixed format defined in the program.

iii a data file TEST which contains first a list of the number of events in each test set followed by the actual events in a fixed format.

Unspecified values are coded as 0. The format part of the EDIT statements must be modified to accept different format events. Different numbers of event sets and the maximum number of events must also involve program modification

COPY(S1,D) -- This routine allocates a new selector node (D) and copies the information from selector S1 into D.

ALLOC(F) -- This routine allocates a new selector node from the list of free selector nodes. If this list is empty, the garbage collection procedure is invoked. Each node is marked with a 0. Then, each intermediate star and temporary selector is marked with a 1. All node marked 0 are collected into the free list. If this list contains less than 10% of the total number of nodes, a message is printed, otherwise, requested selector node is returned in the parameter F

COVER(R,S) -- This function determines whether complex R covers complex S. If it does, then COVER returns a 1, otherwise it returns a 0.

CONT(E,LIST,FLAG) -- The function CONT decides if the event E is contained inthe complex LIST (if FLAG is 1) or in the star LIST (if FLAG is 0). If E is zero, the event to test is assumed to be stored in the array EVE, otherwise, it is in EVENT. CONT returns a 1 if the event E is stored in the list LIST, otherwise, CONT returns 0.

INSERT -- This subroutine inserts the complex Q into the intermediate star LIST in the proper place and then trims the star to MAXSTR elements. The new complex is first compared against each element of the current intermediate star. In such a comparision, if one complex has more weight (COST) than the other, the routine COVER is invoked to determine if one complex covers the other. If it does, the second complex is eliminated from further consideration. The star list is kept in order of decreasing weight so that the first complex in the list is always the best.

XMIN, XMAX -- These routines find the min and max of two elements.

QLQT(I) -- This function selects the best complex from the list I. Currently, the best complex is always the first on the list.

Set 1 [illegible]

```
                    THE FOLLOWING   3  COMPLEXES COVER SET    1                                  COV  NEW  IND  TOT

COMPLEX:  (  31<=X 1<=  74) (X18<=  29) (X19<= 192, 331) (X22<=  57, 108) ( 541, 556<=X23<= 863, 889
                                                                                                  37   37   10   37
          ( 191, 220<=X25) (X27<=   1,   2) (X30<=   1) (X32<=   2) (X33<=   1,   2)
          (X34<=   1,   2) (X39<=   2) (X42<=   1) (X48<=   1) (X50<=   1)

COMPLEX:  (   2,  18<=X 1<=  76,  78) (X14<= 100) (X15<=   6) (X16<=   1) (X17<=   1)            31   14    6   51
          (X18<=  41, 107) (  33,  36<=X19<= 332, 399) (X23<= 863, 889) (X27<=   1)
          (X31<=   1) (X32<=   1) (X33<=   1,   2) (X39<=   2) (X41<=   1,   2)
          (X43<=   2)

COMPLEX:  (  18<=X 1<=  64) (X10<=   1) (X11<=   1) (X12<=   1) (X15<=   4) (   7<=X18)            30    7    7   58
          (X19<= 200, 331) (X21<=  37,  39) (   9,  10<=X22<=  91, 509) (X23<= 844, 863)
          ( 251, 260<=X25) (X27<=   1,   2) (X30<=   1) (X39<=   2) (X43<=   2)
          (X47<=   1)


                    THE FOLLOWING  11  COMPLEXES COVER SET    2                                  COV  NEW  IND  TOT

COMPLEX:  (  28<=X 1<=  71,  74) (   2<=X 3) (X14<= 103) (X17<=   1) (  15<=X18)                 109  109    2  109
          (X19<=1520,1709) (  12,  14<=X21<=  96) ( 471, 480<=X23<= 960, 997)
          (X24<= 420, 449) ( 201, 210<=X25) (   2<=X35) (X41<=   2) (X45<=   2)
          (   2<=X48<=   2) (X50<=   2)

COMPLEX:  (  24,  26<=X 1<=  65) (   2<=X 3) (  13,  15<=X18) (  49,  52<=X19<=5450,5779)         128   45    6  154
          (  20,  22<=X22) (X23<= 852, 869) (X24<= 341, 349) ( 251<=X25<= 645, 838)
          (   2<=X35) (   2<=X48<=   2)

COMPLEX:  (  34,  35<=X 1<=  72) (X14<= 103, 104) (   3<=X15) (X17<=   1) (  37,  38<=X19)         60   16    6  170
          (   3<=X20) (  19<=X21<= 105, 107) (  27<=X22<= 112, 114) (X23<= 874, 879)
          ( 171, 220<=X25) (X28<=   2) (   2<=X33) (   2<=X35) (X42<=   2) (X44<=   2)
          (   2<=X48)

COMPLEX:  (  21,  26<=X 1<=  66) (   2<=X 3) ( 101, 106<=X19<=2164,2249) (  78, 107<=X24<= 380, 419)
                                                                                                  93    5    3  175
          ( 251, 266<=X25) (X32<=   2) (   2<=X33) (X34<=   1) (   2<=X35) (X40<=   1)
          (X41<=   2) (X50<=   2,   3)

COMPLEX:  (  27<=X 1<=  71) (X 5<=   1) (X15<=   8) (   9<=X16<= 255, 335) (  25,  32<=X19<= 520,181
                                                                                                  49    9    5  184
          (X20<=   4) (  19,  20<=X21<=  97, 113) (  26<=X22<= 292, 499) ( 201, 210<=X25<= 483, 489)
          (X28<=   1,   2) (X31<=   2) (X32<=   2) (X40<=   1) (X41<=   2) (X48<=   2)
          (X49<=   2)
```

5.3 Out put Description

The output of the program consists of a list of complexes and statistics. Each complex and its associated trimmed complex is printed along with the generated statistics as described above. The format of each selector is as follows:

$$([a[,b] <=] x \vee [<=[c,] d]) \text{ or } (a,b <: x \vee c: c,d)$$

v is the variable number

a is the lower bound of the generalized (or original) complex if it exists

b is the lower bound of the trimmed complex (b is only printed if it is different from a and a was printed)

c is the upper bound of the trimmed complex (c is only printed if it differs from the generalized upper bound d and d was printed)

d the upper bound of the generalized complex (only printed if it exists).

(The brackets in the formula above indicate that the information inside may be omitted)

The confusion matrix is printed following the complexes. All decisions with decision value within .05 of the highest decision value are printed. Each decision value is calculated as described in CONFUS with the modifications described in section 5.1. A sample output from the program is given on the next few pages.

5.4 Data Structures

The complexes and trimmed complexes are stored in the following 6 arrays:

VAR -- contains the number of the variable associated with the complex

LB,UB -- contain the lower and upper bounds for each selector the generalized complex

TLB,TUB -- contain the lower and upper bounds for each selector of the trimmed complex

SPTR -- contains a pointer to the next complex

CPTR -- is an array which contains x pointers to the complexes stored in the above structure. Each element of row i of CPTR points to a complex which covers a part of the event set F^i

NUMCOV, TOTCOV, IND, NEWCOV -- are arrays which accumulate statistics for corresponding complexes of CPTR

BCOV -- is a binary array which contains a '1'B in row i and column j if the event j was covered by the i^{th} complex.

5.5 Program Structure

The first set of statements read the generalized complexes into the proper arrays and initialize the array CPTR. Next, the trimmed complexes are calculated and the appropriate statistics generated. After all statistics have been generated for one decision class, the complexes and statistics are printed. After all trimmed complexes have been formed and the complexes printed, the confusion matrix for each event set is printed using the calculations described in section 5.1.

The prodedure XMATCH forms an array CSAT which contains the decision values for each complex. The best complex is selected from this array and returned to the confusion matrix section of the program A complete listing of the program appears in appendix B.

```
                (   2<=X35) (X42<=   1) (X43<=   2) (X47<=   1) (   2<=X48<=   2) (X50<=   2,   3)

COMPLEX:  (   2,   9<=X 1<=  84,  85) (X 3<=   1) (X14<= 103) (X15<=   5) (X17<=   1)                    26   17    4   57
          (X18<= 896, 997) (  57,  58<=X19<=1710,2249) (   4<=X20) (  24<=X21<= 168, 195)
          (  12,  16<=X22) ( 157, 160<=X24<= 380, 389) (X25<= 516, 519) (X32<=   2)
          (X39<=   2) (X43<=   2) (   2<=X48)

COMPLEX:  (X 1<=  76,  78) (X15<=   7) (  20<=X18<= 418, 449) (  17,  19<=X21<= 216, 449)                  31   11    0   68
          (  12,  17<=X22) (X23<= 880, 899) ( 191, 194<=X24<= 360, 379) ( 283, 299<=X25)
          (   2<=X26) (X39<=   2) (X40<=   1) (X43<=   2) (X45<=   2) (X47<=   1)
          (   2<=X48<=   2)

COMPLEX:  (   2,   9<=X 1) (X14<= 100) (   2<=X15<=   5) (  10<=X18<= 836, 997)                           26    3    1   71
          (  63,  76<=X19<=1484,1539) (   2<=X20) (  21,  24<=X21) (  14,  16<=X22<= 600, 619)
          ( 108, 150<=X24<= 390, 410) (X25<= 516, 525) (X28<=   1) (   2<=X35)
          (X41<=   2) (X42<=   1) (X43<=   2) (   2<=X48)

COMPLEX:  (   2,   6<=X 1) (X14<= 103, 104) (X18<= 200, 249) (   9,  14<=X21)                             45    7    5   78
          ( 151, 164<=X24<= 330, 339) ( 181, 190<=X25) (X29<=   1) (X34<=   1)
          (   2<=X38) (X43<=   2) (   2<=X48<=   2) (X49<=   2)

COMPLEX:  (   2,  18<=X 1<=  52,  54) (X15<=   5) (  25,  56<=X19) (X21<= 114, 166)                      14    4    4   82
          (X23<= 790, 809) (X28<=   1,   2) (X34<=   1,   2) (   2<=X35<=   2)
          (X36<=   2) (X42<=   1) (   2<=X48<=   2)

COMPLEX:  (X 1<=  69) (X15<=   7) (  10,  11<=X18) (  10,  14<=X21<= 147, 159)                           33    1    1   83
          ( 341, 345<=X25<= 530, 545) (   2<=X26) (X32<=   2) (X42<=   2) (X43<=   2)
          (X45<=   2) (X47<=   1) (   2<=X48<=   2)


                THE FOLLOWING   3  COMPLEXES COVER SET    5                                              COV  NEW  IND  TOT

COMPLEX:  (  40,  41<=X 1<=  77,  78) (X14<= 101, 102) (X17<=   1) (  22,  29<=X18<= 212, 215)           10   10    3   10
          (  49, 104<=X19) (   2<=X20) (X21<= 107, 114) (  21,  27<=X22<=  82,  85)
          (X32<=   2) (   2<=X35) (X40<=   1) (X43<=   2) (X44<=   2) (X46<=   1)
          (X47<=   1)

COMPLEX:  (  24,  27<=X 1<=  69,  70) (X12<=   1) (X14<= 100, 101) (X15<=   4)                            9    5    4   15
          (X17<=   1) (X18<= 184, 483) (  53,  60<=X19) (   2<=X20) (  12,  18<=X21)
          (  14,  17<=X22<=  82) ( 108, 133<=X24<= 380, 415) (   2<=X35) (X42<=   2)
          (X43<=   2) (X48<=   2) (X50<=   1,   3)

COMPLEX:  (  51,  57<=X 1<=  74,  78) (  91, 104<=X19<= 338, 375) (  13,  14<=X21<=  57,  77)             7    1    1   16
          (  27,  28<=X22<=  82, 111) (   2<=X35) (X43<=   2) (X45<=   2)
```

```
COMPLEX:  (   2<=X 3) (X15<=   8) (  13<=X18<= 410, 435) (X21<= 116, 161) (X22<= 144, 146)          49    -    -   49
          ( 251, 260<=X25<= 605, 838) (X34<=   2) (   2<=X35) (   2<=X38) (X41<=   2)
          (X42<=   2) (X45<=   2) (X48<=   2)

COMPLEX:  (  22,  26<=X 1<=  58) (   2<=X 3) (X14<= 103) (   3,   8<=X18) (  65,  70<=X19)          90    4    4  194
          (  12,  14<=X21) (  25<=X22) ( 134, 140<=X24) ( 231, 240<=X25) (   2<=X26)
          (   2<=X35) (X41<=   1) (X42<=   2)

COMPLEX:  (  44<=X 1<=  50) (   2<=X 3) (X14<= 102) (  59,  69<=X19<= 940,1039)                     22    2    2  196
          ( 191, 200<=X25<= 400, 409)

COMPLEX:  (  60<=X 1<=  66) (  15<=X21<= 131, 159) (X22<= 176, 183) (  15, 230<=X23)                25    1    1  197
          ( 301, 310<=X25) (   2<=X35) (X36<=   2) (X44<=   2) (   2<=X48<=   2)

COMPLEX:  (   2,  18<=X 1<=  28,  29) (  23<=X21<=  28)

COMPLEX:  (  31,  32<=X 1<=  73,  77) (  81,  82<=X19<= 292, 299) (  27,  29<=X21<=  52,  56)      23    1    1  199
          (  60<=X22<= 413, 564)

                    THE FOLLOWING   4  COMPLEXES COVER SET    3                                    COV  NEW  IND  TOT

COMPLEX:  (X 1<=  62,  65) (X11<=   1) (   3<=X15) (X17<=   1) (  13,  14<=X18)                    21   21    4   21
          (  75,  80<=X19<=2140.5779) (   7,  16<=X21) (   7,  18<=X22<= 478, 599)
          ( 521, 540<=X23) ( 161, 210<=X24) ( 251, 260<=X25) (X26<=   2) (X32<=   2)
          (X35<=   2) (X36<=   2) (X42<=   2) (X45<=   2) (   2<=X48) (   2<=X49)

COMPLEX:  (X 1<=  68,  71) (X 5<=   1) (X 9<=   1) (X14<=  99) (X15<=   7) (  11,  12<=X18)        20    7    5   28
          (  53,  74<=X19) (  15,  16<=X21) (  31,  35<=X22<= 478, 699) ( 511, 530<=X23)
          ( 210<=X24) (X26<=   2) (X41<=   2) (   2<=X48)

COMPLEX:  (   2,   3<=X 1<=  68,  73) (X 3<=   1) (X 4<=   1) (X14<= 102, 103)                     13    3    3   31
          (   3<=X15<=   7) (   8<=X18) (   9,  10<=X21<= 126, 198) (  25<=X22<= 135, 338)
          (X23<= 780, 789) (X24<= 360, 415) ( 271, 300<=X25) (   2<=X36) (X47<=   1)
          (   2<=X48)

COMPLEX:  (  51,  56<=X 1<=  74,  76) (  19,  26<=X22<=  42,  43) ( 217, 221<=X24<= 225, 267)       2    1    1   32
          (   2<=X48<=   2)

                    THE FOLLOWING   7  COMPLEXES COVER SET    4                                    COV  NEW  IND  TOT

COMPLEX:  (X 1<=  72,  78) (X17<=   1) (  39,  48<=X19<=1130,1137) (   9,  11<=X21)               40   40    3   40
          (X22<= 360, 424) ( 161, 164<=X24<= 410) (   2<=X26) (X34<=   1,   2)
```

6 PBLOG - Program Description

This program prints out a confusion matrix based on VL_1 formulas produced by the program AQ.FOR. The decision values on which decisions are based are obtained by selecting the set of complexes which best describe the given event and then calculating the event set F^i which has the highest probability of being covered by this set of complexes.

More precisely, the probability P(event $\in F^i$ | cpx = j) (i.e. the probability that the event covered by complex j is a member of the set F^i is calculated). This is printed in a table, each row of which corresponds to a different complex (i.e. the element (j,i) is P(event $\in F^i$ | cpx = j)).

For each event in the listing set, those complexes C_j which most closely match the event (DC within .05 of the best match) are selected. The average probability $\underset{C_j}{\text{avg}}$ P(event $\in F^i$ | cpx = C_j) = P_i' taken over all such complexes C_j is computed. Then the resulting figures are normalized to 1 and the best two such values are printed in the appropriate columns. In addition, statistics are maintained about the number of correct decisions and the total number of decisions made altogether. The input is identical to that for the program TCPXLS as described in the previous section and the restriction about program modification for application to a different problem still apply.

```
000000                  IMPLICIT INTEGER (A-Z)
000000                  COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),TA
000000                 1U,FREE,TU,SEC(200),NOSEC,NC1
000000                 2,F1(350),F2(350),FQ(350),FP(350),EVENT(350,50),SIZE,MAXSTR,NDIM
000000                 1,HSLASH(50),COST(200),T,TS,GLIST,OLIST,NC1,NC2,TLIST,C1,EVE(50),IFL
000000                  INTEGER DIM,Z,I=(2),SPTR,PTR,MARK,FREE,F1,Y2,FQ,FP,SIZE,HSLASH
000000                 1,P,Q,R,S,1,TS,TLIST,GLIST,OLIST,CST,COST,SEC
000000                  INTEGER TEMP(30),THRESH(15,30)
000000                  REAL RAN
000000                  CALL OPFILE(21,'FORM.DAT')
000003                  CALL IFILE(22,'FSIZE.')
000006                  LOG=-1
000007                  UU=0
000010          11      TYPE 110
000013          110     FORMAT(' ENTER THE NUMBER OF DIMENSIONS I2')
000024                  TYPE 108
000027          108     FORMAT(' HOW MANY EVENTS TOTAL IN EVENT SET?')
000040                  ACCEPT 107,NC1
000044          107     FORMAT(I)
000046                  ACCEPT 111,NDIM
000052          111     FORMAT(2012)
000055                  TYPE 112
000060                  IF (MAXSTR.GT.0)MAXSTR=MAXSTR-1
000063          112     FORMAT(' ENTER MAXSTAR AND NOSEC  I2')
000073                  ACCEPT 111,MAXSTR,NOSEC
000100                  TYPE 113
000103          113     FORMAT(' ENTER CLASS TOBE COVERED I2')
000113                  ACCEPT 111,C1
000117                  TYPE 114
000122          114     FORMAT(' HOW MANY EVENTS OF C2')
000131                  ACCEPT 109,NC2
000135                  CALL DEFINE FILE(1,NDIM+1,IV,'DATA.JIM',*4113,*1372)
000147                  TYPE 117
000152          117     FORMAT(' ENTER BEGINNING AND SIZE OF CLASS I4,I3')
000164                  ACCEPT 123,IB,NC1,OO
000172          123     FORMAT(I4,2I3)
000175          109     FORMAT(2I3)
000177          31      DO 1 I=IB,IB+NC1-1
000205                  J=1
000206                  IF(OO.NE.0)J=RAN(J)*OO+IB
000223          1       READ(1*J)CLAS,(EVENT(I-IB+1,JJ),JJ=1,50)
000247                  DO 3 I=1,NC1
000251          3       F1(I)=1
000254                  CALL AQ
000255                  WRITE(21,111)LOG
000261                  IF(OO.EQ.0)IB=IB+NC1
000265                  IF(OO.NE.0)IB=IB+OO
000271                  READ(22,109,END=300)NC1,OO
000306                  C1=MOD(C1+1,20)
000307                  GO TO 31
000310                  TYPE 118
000313          118     FORMAT(' GO ON TO NEXT CLASS ?')
000322                  ACCEPT 119,A
```

```
000326                  119   FORMAT(1A1)
000330                        IF(A.EQ.'Y')GO TO 11
000333                  300   STOP
000334                        END
```

CONSTANTS

0	000000000025	1	432372246534	2	422032420100	3	000000000000	4	000000000026		
5	432471155212	6	271004020100	7	000000000000	10	000000000001	11	422032[illegible]		
12	452231520100	13	000000000000	14	000000004113	15	000000001372	16	000000030024		
17	545004020100										

COMMON

DIM	/.COMM./+0	XLB	/.COMM./+310	UB	/.COMM./+620	SPTR	/.COMM./+1130	PTR	/.COMM./+1440
MARK	/.COMM./+1750	TAU	/.COMM./+2260	FREE	/.COMM./+2261	TO	/.COMM./+2262	SEC	/.COMM./+2263
NOSEC	/.COMM./+2573	NCT	/.COMM./+2574	F1	/.COMM./+2575	F2	/.COMM./+3333	FO	/.COMM./+4071
FP	/.COMM./+4627	EVENT	/.COMM./+5365	SIZE	/.COMM./+47521	MAXSTR	/.COMM./+47522	NDIM	/.COMM./+47523
HSLASH	/.COMM./+47524	COST	/.COMM./+47606	T	/.COMM./+50116	TS	/.COMM./+50117	GLIST	/.COMM./+50120
OLIST	/.COMM./+50121	NC1	/.COMM./+50122	NC2	/.COMM./+50123	TLIST	/.COMM./+50124	C1	/.COMM./+50125
EVE	/.COMM./+50126	IF	/.COMM./+50210						

SUBPROGRAMS

FORSE. .JHFF OFILE IFILE INTO. INTI. DEFINE RAN RECNO. RANAC. BINAR. AQ END. MOD ALPHO.
ALPHI. EXIT

SCALARS

LOG	362	OO	363	NCT	2574	NDIM	47523	MAXSTR	47522
NOSEC	2573	C1	50125	NC2	50123	IV	364	IO	365
NC1	50122	I	366	J	367	CLAS	370	JJ	371
A	372	TAU	2260	FREE	2261	TO	2262	SIZE	47521
T	50116	TS	50117	GLIST	50120	OLIST	50121	TLIST	50124
IF	50210								

ARRAYS

DIM	0	XLB	310	UB	620	SPTR	1130	PTR	1440
MARK	1750	SEC	2263	F1	2575	F2	3333	FO	4071
FP	4627	EVENT	5365	HSLASH	47524	COST	47606	EVE	50126
IN	373	TEMP	375	THRESH	433				

```
000000                      SUBROUTINE AQ
000000                      IMPLICIT INTEGER (A-Z)
000000                      COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),TA
000000                     1U,FREE,TU,SEC(200),NOSEC,NCT
000000                     1,F1(350),F2(350),FQ(350),FP(350),EVENT(350,50),SIZE,MAXSTR,NDIM
000000                     1,NSLASH(50),COST(200),T,TS,GLIST,OLIST,NC1,NC2,TLIST,C1,EVE(50),IPL
000000                      INTEGER DIM,GLU,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,NSLASH
000000                     1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
000000                      INTEGER C1,E2,E1,E2P,DELTA,CONT,TAU
000000                      REAL RAN
000000                      SIZE=200
000002                 61   DO 60 I=1,SIZE
000004                 60   PTR(I)=I+1
000011                      PHASE=1
000013                      FREE=1
000015                      PTR(SIZE)=0
000017                      CALL ALLOC(GLIST)
000021                      CALL ALLOC(TLIST)
000023                      DO 46 I=1,NC1
000025                      FP(I)=F1(I)
000027                 48   FQ(I)=F1(I)
000033                 49   DELTA=0
000034                      DO 50 I=1,NC1
000035                      IF(400*PHASE+FP(I).EQ.400)GO TO 50
000045                      IF(400*PHASE+FQ(I).EQ.800)GO TO 50
000054                      DELTA=DELTA+1
000055                      E2P=I
000057                      OLIST=0
000060                      E1=F1(I)
000062                 33   J=RAN(J)*NCT+1
000073                 C      TYPE 1141,J
000073                 1141 FORMAT(' NEXT RANDOM EVENT IS ',I4)
000102                      READ(1#J)IC,EVE
000112                      IF(IC.EQ.C1)GO TO 33
000115                 18   IDIM=1
000117                 C      TYPE 114,E2P
000117                      CALL ALLOC(TS)
000121                 19   DIM(TS)=IDIM
000124                      XLB(TS)=0.000
000126                      UB(TS)=10000
000131                      IF(EVE(IDIM)*EVENT(E1,IDIM).EQ.0)GO TO 23
000140                      IF(EVENT(E1,IDIM)-EVE(IDIM))20,21,22
000146                 21   IDIM=IDIM+1
000151                      IF(IDIM-NDIM)19,19,24
000153                 20   UB(TS)=EVE(IDIM)-1
000162                      GO TO 23
000163                 22   XLB(TS)=EVE(IDIM)+1
000170                 23   CALL COPY(TS,T)
000173                      Q=OLIST
000175                      IF(Q.EQ.0)GO TO 16
000177                 11   IF(IABS(DIM(T)-DIM(Q))+IABS(XLB(T)-XLB(Q))+IABS(UB(T)-UB(Q))+SPTR(
000177                     1Q).EQ.0)GO TO 16
000227                      Q=PTR(Q)
```

```
000232                      IF(Q.NE.0)GO TO 11
000234                      Q=QLIST
000236                10    R=Q
000240                      T=0
000241                      CALL COPY(TS,TQ)
000244                      FLG=0
000245                12    CALL COPY(R,S)
000250                      IF(DIM(TS).EQ.DIM(R))GO TO 13
000255                14    SPTR(S)=T
000260                      T=S
000262                      R=SPTR(R)
000265                      IF(R.NE.0)GO TO 12
000267                      IF(FLG.EQ.0)SPTR(TQ)=T
000274                      IF(FLG.EQ.0)T=TQ
000300                      COST(T)=CF(T)
000304                      CALL INSERT(T,QLIST)
000307                15    Q=PTR(Q)
000312                      IF(Q)10,21,10
000313                13    XLB(S)=AMAX(XLB(TS),XLB(R))
000330                      FLG=1
000332                      UB(S)=AMIN(UB(TS),UB(R))
000345                      IF(XLB(S)-UB(S))14,14,15
000350                16    COST(T)=CF(T)
000356                      CALL INSERT(T,QLIST)
000361                      GO TO 21
000362                24    QLIST=PTR(QLIST)
000365                25     E2P=E2P+1
000366                114   FORMAT(' NEXT E2 IS',I4)
000373                      PTR(QLIST)=0
000375                      IF(QLIST.EQ.0)GOTO 56
000377                      IF(E2P.GT.NC2)GO TO 47
000402                34    J=RAN(J)*NC1+1
000413                      READ(1#J)IC,EVE
000423                      IF(IC.EQ.C1)GO TO 34
000426                      IF(CONT(0,QLIST,0))25,25,16
000432                47    CMAX=COST(QLIST)
000437                      P=QLIST
000441                65    IF(COST(P)+NDIM.GT.CMAX)GO TO 63
000446                      PTR(Q)=0
000450                      GO TO 64
000451                63      Q=P
000453                      P=PTR(P)
000456                      IF(P.NE.0)GO TO 65
000460                64    JJ=QLIST
000462                      IF(PTR(QLIST).NE.0)JJ=QLG(QLIST)
000470                      CALL PRT(JJ)
000472                      DO 51 J=1,NC1
000474                      IF(FO(J).EQ.0) GO TO 51
000476                      IF(CONT(FO(J),JJ,1).NE.1) GO TO 52
000507                55    FO(J)=0
000511                      FP(J)=0
000513                      GO TO 51
000514                52    IF(FP(J).EQ.0) GO TO 51
```

MARK	1750	SEC	2263	F1	2575	F2	3333	FU	4071
FP	4027	EVENT	5305	HSLASH	47524	COST	4/000	EVE	50120

```
000000                      SUBROUTINE PRT(R)
000000                      IMPLICIT INTEGER (A-Z)
000000                      COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),TA
000000                     1U,FREE,TO,SEC(200),NOSEC,NCT
000000                     1,F1(350),F2(350),FU(350),FP(350),EVENT(350,50),SIZE,MAXSTR,NDIM
000000                     1,MSLASH(50),COST(200),I,TS,GLIST,OLIST,NC1,NC2,TLIST,C1,EVE(50),IPT
000000                      INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FU,FP,SIZE,MSLASH
000000                     1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
000000                      INTEGER A(200)
000000                      DATA X/'X'/
000000                      K=0
000001                      P=R
000003                      IF(P.EQ.0) RETURN
000006                    4 Q=P
000010                      TYPE 101, COST(P)
000015                      J=1
000017                  102 FORMAT(I3,2I7)
000022                    5 A(J)=X
000025                      A(J+1)=DIM(Q)
000031                      A(J+2)=XLB(Q)
000035                      A(J+3)=UB(Q)
000041                      J=J+4
000043                      WRITE(21,102)DIM(Q),XLB(Q),UB(Q)
000052                      Q=SPTR(Q)
000055                      IF(Q.NE.0)GO TO 5
000057                      J=J-1
000060                      TYPE 100,(A(J),J=1,J)
000071                      P=PTR(P)
000074                  0     IF(P.NE.0) GO TO 4
000076                  101 FORMAT(' WEIGHT OF THIS TERM ',I10)
000103                  100 FORMAT(3(' ',A1,I3,'*',I7,'**',I7,' '))
000113                      WRITE(21,102)K
000117                      RETURN
000120                      END
```

GLOBAL SUMMIES

R 134

COMMON

DIM	/.COMM./+0	XLB	/.COMM./+310	UB	/.COMM./+620	SPTR	/.COMM./+1130	PTR	/.COMM./+1440		
MARK	/.COMM./+1750	TAU	/.COMM./+2260	FREE	/.COMM./+2261	TO	/.COMM./+2262	SEC	/.COMM./+2263		
NOSEC	/.COMM./+2573	NCT	/.COMM./+2574	F1	/.COMM./+2575	F2	/.COMM./+3333	FU	/.COMM./+4071		
FP	/.COMM./+4627	EVENT	/.COMM./+5365	SIZE	/.COMM./+47521	MAXSTR	/.COMM./+47522	NDIM	/.COMM./+47523		
MSLASH	/.COMM./+47524	COST	/.COMM./+47606	I	/.COMM./+50116	TS	/.COMM./+50117	GLIST	/.COMM./+50120		
OLIST	/.COMM./+50121	NC1	/.COMM./+50122	NC2	/.COMM./+50123	TLIST	/.COMM./+50124	C1	/.COMM./+50125		
EVE	/.COMM./+50126	IPT	/.COMM./+50210								

SUBPROGRAMS

INTU. INTI. ALPHU. ALPHI.

```
000517                      IF(CONT(FP(J),OLIST,0),EQ.1)FP(J)=0
000533                51    CONTINUE
000536                      PTR(JJ)=0
000540                      CALL INSERT(JJ,TLIST)
000543                C       TYPE 116
000543                C       ACCEPT 117 ,A
000543                C       IF(A.NE.'Y')RETURN
000545                116   FORMAT(' CONTINUE?')
000547                117   FORMAT(A1)
000551                50    CONTINUE
000554                      PHASE=PHASE+1
000555                      IF(PHASE.EQ.2) GO TO 49
000560                C       TYPE 102,DELTA
000560                102   FORMAT(' DELTA=',I3)
000564                      RETURN
000565                56    TYPE 115,E1,J
000572                115   FORMAT( 'OOPS, OLD STAR WIPED OUT ','E1 EVENT =',I4,
000572               1' E2 EVENT=',I4)
000610                      STOP
000611                      END
```

CONSTANTS

```
0       000000000000    1       000000000001
```

COMMON

```
DIM     /.COMM./+0      XLB     /.COMM./+310    UB      /.COMM./+620    SPTR    /.COMM./+1130   PTR     /.COMM./+1440
MARK    /.COMM./+1750   IAU     /.COMM./+2260   FREE    /.COMM./+2261   TQ      /.COMM./+2262   SEC     /.COMM./+2263
NOSEC   /.COMM./+2573   NCT     /.COMM./+2574   F1      /.COMM./+2575   F2      /.COMM./+3333   FQ      /.COMM./+40171
FP      /.COMM./+4627   EVENT   /.COMM./+5365   SIZE    /.COMM./+47521  MAXSTR  /.COMM./+47522  NDIM    /.COMM./+47523
HSLASH  /.COMM./+47524  COST    /.COMM./+47600  T       /.COMM./+50116  TS      /.COMM./+50117  OLIST   /.COMM./+50120
OLIST   /.COMM./+50121  NC1     /.COMM./+50122  NC2     /.COMM./+50123  TLIST   /.COMM./+50124  C1      /.COMM./+50125
EVE     /.COMM./+50126  IF      /.COMM./+50210
```

SUBPROGRAMS

```
ALLOC   RAN     INTO.   INTI.   RECRO.  RANAC.  BINWR.  COPY    IABS    CF      INSERT  XMAX    XMIN    CONT    OLD
PRT     ALPHO.  ALPHI.  EXIT
```

SCALARS

```
AG      633             SIZE    47521           I       634             PHASE   635             FREE    2261
GLIST   50120           TLIST   50124           NC1     50122           DELTA   636             E2P     637
OLIST   50121           E1      640             J       641             NCT     2574            IC      642
C1      50125           IDIM    643             TS      50117           NDIM    47523           T       50116
Q       644             M       645             TQ      2262            FLG     646             S       647
NC2     50123           CMAX    650             P       651             JJ      652             IAU     2260
NOSEC   2573            MAXSTR  47522           IF      50210
```

ARRAYS

```
DIM     0               XLB     310             UB      620             SPTR    1130            PTR     1440
```

APPENDIX A

AQ.FOR Program Listing

SCALARS

PRT	135	X	136	K	137	P	140	M	134
Q	141	I	142	J	143	TAU	2260	FREE	2261
TQ	2262	NOSEC	2573	NCT	2574	SIZE	47521	MAXSTK	47522
NDIM	47523	I	50116	TS	50117	GLIST	50120	ULIST	50121
NC1	50122	NC2	50123	TLIST	50124	C1	50125	IP	50210

ARRAYS

DIM	0	XL0	310	UB	620	SPTR	1130	PTR	1440
MARK	1750	SEC	2263	F1	2575	F2	3333	FU	4071
FP	4627	EVENT	5365	HSLASH	47524	COST	47600	EVE	50126
A	144								

```
000000                  INTEGER FUNCTION CF(R)
000000                  IMPLICIT INTEGER (A-Z)
000000                  COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),FA
000000                 1U,FREE,TO,SEC(200),NOSEC,NCT
000000                 1,F1(350),F2(350),FQ(350),FP(350),EVENT(350,50),SIZE,MAXSTK,NDIM
000000                 1,HSLASH(50),COST(200),T,TS,GLIST,OLIST,NC1,NC2,TLIST,C1,EVE(50),IFL
000000                  INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
000000                 1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC,E1
000000                  CF=0
000001                  SEC(R)=0
000003                  DO 1 I=1,NC1
000005                  E1=F1(I)
000007                  IF(E1.EQ.0) GO TO 1
000011                  IF(FQ(I).EQ.0.AND.NOSEC.EQ.1)GO TO 1
000023                  P=R
000025                2 KK=DIM(P)
000030                  IF(EVENT(E1,KK).EQ.0)GO TO 14
000035                  IF(XLB(P).GT.EVENT(E1,KK)) GOTO 1
000044                  IF(UB(P).LT.EVENT(E1,KK)) GO TO 1
000053                 14 P=SPTR(P)
000056                  IF(P.NE.0)GO TO 2
000060                  SEC(R)=SEC(R)+NDIM
000063                  IF(FQ(I).NE.0)CF=CF+NDIM
000070                1 CONTINUE
000073                  P=R
000075                4 CF=CF-1
000076                  P=SPTR(P)
000101                  IF(P.NE.0) GO TO 4
000103                  RETURN
000104                  END
```

GLOBAL DUMMIES

R 143

COMMON

DIM	/.COMM./+0	XLB	/.COMM./+310	UB	/.COMM./+620	SPTR	/.COMM./+1130	PTR	/.COMM./+1440
MARK	/.COMM./+1750	TAU	/.COMM./+2260	FREE	/.COMM./+2261	TO	/.COMM./+2262	SEC	/.COMM./+2263
NOSEC	/.COMM./+2573	NCT	/.COMM./+2574	F1	/.COMM./+2575	F2	/.COMM./+3333	FQ	/.COMM./+4071
FP	/.COMM./+4027	EVENT	/.COMM./+5305	SIZE	/.COMM./+47521	MAXSTK	/.COMM./+47522	NDIM	/.COMM./+47523
HSLASH	/.COMM./+47524	COST	/.COMM./+47606	T	/.COMM./+50116	TS	/.COMM./+50117	GLIST	/.COMM./+50120
OLIST	/.COMM./+50121	NC1	/.COMM./+50122	NC2	/.COMM./+50123	TLIST	/.COMM./+50124	C1	/.COMM./+50125
EVE	/.COMM./+50126	IF	/.COMM./+50210						

SCALARS

CF	144	R	143	I	145	NC1	50122	E1	146
NOSEC	2573	P	147	KK	150	NDIM	47523	TAU	2260
FREE	2261	TO	2262	NCT	2574	SIZE	47521	MAXSTK	47522
T	50116	TS	50117	GLIST	50120	OLIST	50121	NC2	50123
TLIST	50124	C1	50125	IF	50210				

```
000000                          SUBROUTINE COPY(S1,D)
000000                          IMPLICIT INTEGER (A-Z)
000000                          COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),TA
000000                         1O,FREE,TQ,SEC(200),NOSEC,NCT
000000                         1,F1(350),F2(350),FQ(350),FP(350),EVENT(350,50),SIZE,MAXSTR,NDIM
000000                         1,HSLASH(50),COST(200),T,TS,GLIST,OLIST,NC1,NC2,TLIST,C1,EVE(50),IF
000000                          INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
000000                         1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
000000                          INTEGER D,S1
000000                          CALL ALLOC(D)
000002                          DIM(D)=DIM(S1)
000006                          XLB(D)=XLB(S1)
000012                          UB(D)=UB(S1)
000016                          RETURN
000017                          END
```

GLOBAL DUMMIES

S1	36	D	37

COMMON

DIM	/.COMM./+0	XLB	/.COMM./+310	UB	/.COMM./+620	SPTR	/.COMM./+1130	PTR	/.COMM./+1440		
MARK	/.COMM./+1750	TAU	/.COMM./+2260	FREE	/.COMM./+2261	TQ	/.COMM./+2262	SEC	/.COMM./+2263		
NOSEC	/.COMM./+2573	NCT	/.COMM./+2574	F1	/.COMM./+2575	F2	/.COMM./+3333	FQ	/.COMM./+4071		
FP	/.COMM./+4627	EVENT	/.COMM./+5365	SIZE	/.COMM./+47521	MAXSTR	/.COMM./+47522	NDIM	/.COMM./+47523		
HSLASH	/.COMM./+47524	COST	/.COMM./+47606	T	/.COMM./+50116	TS	/.COMM./+50117	GLIST	/.COMM./+50120		
OLIST	/.COMM./+50121	NC1	/.COMM./+50122	NC2	/.COMM./+50123	TLIST	/.COMM./+50124	C1	/.COMM./+50125		
EVE	/.COMM./+50126	IF	/.COMM./+50210								

SUBPROGRAMS

ALLOC

SCALARS

COPY	40	D	37	S1	36	TAU	2260	FREE	2261
TQ	2262	NOSEC	2573	NCT	2574	SIZE	47521	MAXSTR	47522
NDIM	47523	T	50116	TS	50117	GLIST	50120	OLIST	50121
NC1	50122	NC2	50123	TLIST	50124	C1	50125	IF	50210

ARRAYS

DIM	0	XLB	310	UB	620	SPTR	1130	PTR	1440
MARK	1750	SEC	2263	F1	2575	F2	3333	FQ	4071
FP	4627	EVENT	5365	HSLASH	47524	COST	47606	EVE	50126

```
000000                          SUBROUTINE ALLOC(F)
000000                          IMPLICIT INTEGER (A-Z)
000000                          COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),TA
000000                         1U,FREE,TD,SEC(200),NOSEC,NCT
000000                         1,F1(350),F2(350),FQ(350),FP(350),EVENT(350,50),SIZE,MAASTR,NDIM
000000                         1,MSLASH(50),COST(200),I,TS,GLIST,OLIST,NC1,NC2,TLIST,C1,EVE(50),IYE
000000                          INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,MSLASH
000000                         1,P,Q,R,SST,TS,TLIST,GLIST,OLIST,CST,COST,SEC
000000                          INTEGER F,A(6)
000000                          F=FREE
000002                          FREE=PTR(FREE)
000005                          SPTR(F)=0
000007                          PTR(F)=0
000011                          IF(PTR(FREE).NE.0)RETURN
000015                          REI=REI+1
000016                          DO 1 I=1,SIZE
000020                     1    MARK(I)=0
000023                          MARK(F)=1
000026                          A(1)=OLIST
000030                          A(2)=TLIST
000032                          A(3)=GLIST
000034                          A(4)=TS
000036                          A(5)=I
000040                          A(6)=TD
000042                          DO 10 I=1,6
000044                          P=A(I)
000046                          IF(P.EQ.0)GO TO 10
000050                     3    Q=P
000052                     2    IF  (MARK(Q).EQ.1) GO TO 4
000056                          MARK(Q)=1
000061                          Q=SPTR(Q)
000064                          IF(Q.NE.0) GO TO 2
000066                     4    P=PTR(P)
000071                          IF(P.NE.0)GO TO 3
000073                    10    CONTINUE
000076                          J=0
000077                          DO 11 I=1,SIZE
000101                          IF(MARK(I).EQ.1) GO TO 11
000104                          PTR(I)=FREE
000106                          FREE=I
000107                          J=J+1
000110                    11    CONTINUE
000112                          IF(J.LT..1*SIZE)TYPE 100
000123                          RETURN
000124                    100   FORMAT('  USING 90% OF SPACE')
000132                          END

CONSTANTS

0       175631463146

GLOBAL DUMMIES
```

ARRAYS

DISK	0	XLB	310	UB	620	SPTR	1130	PTR	1640
MARK	1750	SEC	2263	F1	2575	F2	3333	FU	4071
FP	4627	EVENT	5365	HSLASH	47524	COST	47600	EVE	50120

```
000000                      INTEGER FUNCTION COVER(S,R)
000000                      IMPLICIT INTEGER (A-Z)
000000                      COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),TA
000000                     1U,FREE,TO,SEC(200),NOSEC,NCT
000000                     1,F1(350),F2(350),FQ(350),FP(350),EVENT(350,50),SIZE,MAXSTR,NDIM
000000                     1,HSLASH(50),COST(2007),T,TS,GLIST,OLIST,NC1,NC2,TLIST,C1,EVE(50),IFQ
000000                      INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
000000                     1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
000000                      COVER=0
000001                      P=S
000003                  3   Q=R
000005                  1   IF(DIM(P).EQ.DIM(Q)) GO TO 2
000012                      Q=SPTR(Q)
000015                      IF(Q.NE.0) GO TO 1
000017                      RETURN
000020                  2   IF(XLB(P).GT.XLB(Q))RETURN
000026                      IF(UB(P).LT.UB(Q))RETURN
000034                      P=SPTR(P)
000037                      IF(P.NE.0)GO TO 3
000041                      COVER=1
000043                      RETURN
000044                      END
```

GLOBAL DUMMIES

S	102	R	103

COMMON

DIM	/.COMM./+0	XLB	/.COMM./+310	UB	/.COMM./+620	SPTR	/.COMM./+1130	PTR	/.COMM./+1440
MARK	/.COMM./+1750	TAU	/.COMM./+2260	FREE	/.COMM./+2261	TO	/.COMM./+2262	SEC	/.COMM./+2263
NOSEC	/.COMM./+2573	NCT	/.COMM./+2574	F1	/.COMM./+2575	F2	/.COMM./+3333	FQ	/.COMM./+4071
FP	/.COMM./+4627	EVENT	/.COMM./+5365	SIZE	/.COMM./+47521	MAXSTR	/.COMM./+47522	NDIM	/.COMM./+47523
HSLASH	/.COMM./+47524	COST	/.COMM./+47606	T	/.COMM./+50116	TS	/.COMM./+50117	GLIST	/.COMM./+50120
OLIST	/.COMM./+50121	NC1	/.COMM./+50122	NC2	/.COMM./+50123	TLIST	/.COMM./+50124	C1	/.COMM./+50125
EVE	/.COMM./+50126	IF	/.COMM./+50210						

SCALARS

COVER	104	P	105	S	102	Q	106	R	103
TAU	2260	FREE	2261	TO	2262	NOSEC	2573	NCT	2574
SIZE	47521	MAXSTR	47522	NDIM	47523	T	50116	TS	50117
GLIST	50120	OLIST	50121	NC1	50122	NC2	50123	TLIST	50124
C1	50125	IF	50210						

ARRAYS

DIM	0	XLB	310	UB	620	SPTR	1130	PTR	1440
MARK	1750	SEC	2263	F1	2575	F2	3333	FQ	4071
FP	4627	EVENT	5365	HSLASH	47524	COST	47606	EVE	50126

```
000000                  INTEGER FUNCTION CONT(E,LIST,FLAG)
000000                  IMPLICIT INTEGER (A-Z)
000000                  COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),TA
000000                 1U,FREE,TQ,SEC(200),NOSEC,NCT
000000                 1,F1(350),F2(350),FQ(350),FP(350),EVENT(350,50),SIZE,MAXSTR,NDIM
000000                 1,NSLASH(50),COST(200),T,TS,GLIST,OLIST,NC1,NC2,TLIST,C1,EVE(50),IFL
000000                  INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,NSLASH
000000                 1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
000000                  INTEGER E,FLAG
000000                  CONT=0
000001                  P=LIST
000003               3  Q=P
000005               2  IF(E.NE.0)XT=EVENT(E,DIM(Q))
000015                  IF(E.EQ.0)XT=EVE(DIM(Q))
000023                  IF(XT.EQ.0)GO TO 12
000025                  IF(XT.LT.XLB(Q))GO TO 1
000031                  IF(XT.GT.UB(Q)) GO TO 1
000035              12  Q=SPTR(Q)
000040                  IF(Q.NE.0) GO TO 2
000042                  CONT=1
000044                  RETURN
000045               1  IF(FLAG.EQ.1)RETURN
000051                  P=PTR(P)
000054                  IF(P.NE.0)GO TO 3
000056                  RETURN
000057                  END
```

GLOBAL DUMMIES

E	116	LIST	117	FLAG	120

COMMON

DIM	/.COMM./+0	XLB	/.COMM./+310	UB	/.COMM./+620	SPTR	/.COMM./+1130	PTR	/.COMM./+1440
MARK	/.COMM./+1750	TAU	/.COMM./+2260	FREE	/.COMM./+2261	TQ	/.COMM./+2262	SEC	/.COMM./+2263
NOSEC	/.COMM./+2573	NCT	/.COMM./+2574	F1	/.COMM./+2575	F2	/.COMM./+3333	FQ	/.COMM./+4071
FP	/.COMM./+4627	EVENT	/.COMM./+5305	SIZE	/.COMM./+47521	MAXSTR	/.COMM./+47522	NDIM	/.COMM./+47523
NSLASH	/.COMM./+47524	COST	/.COMM./+47606	T	/.COMM./+50116	TS	/.COMM./+50117	GLIST	/.COMM./+50120
OLIST	/.COMM./+50121	NC1	/.COMM./+50122	NC2	/.COMM./+50123	TLIST	/.COMM./+50124	C1	/.COMM./+50125
EVE	/.COMM./+50126	IF	/.COMM./+50210						

SCALARS

CONT	121	P	122	LIST	117	Q	123	E	116
XT	124	FLAG	120	TAU	2260	FREE	2261	TQ	2262
NOSEC	2573	NCT	2574	SIZE	47521	MAXSTR	47522	NDIM	47523
T	50116	TS	50117	GLIST	50120	OLIST	50121	NC1	50122
NC2	50123	TLIST	50124	C1	50125	IF	50210		

ARRAYS

DIM	0	XLB	310	UB	620	SPTR	1130	PTR	1440
MARK	1750	SEC	2263	F1	2575	F2	3333	FQ	4071

FP 4027 EVENT 5305 HSLASH 47524 COST 47600 EVE 50120

```
000000                      SUBROUTINE INSERT(Q,LIST)
000000                      IMPLICIT INTEGER (A-Z)
000000                      COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),FA
000000                     1U,FREE,TO,SEC(200),NOSEC,ECT
000000                     1,F1(350),F2(350),FQ(350),FP(350),EVENT(350,50),SIZE,MAXSTR,NDIM
000000                     1,NSLASH(50),COST(200),T,TS,GLIST,OLIST,NC1,NC2,TLIST,C1,EVE(50),IFG
000000                      INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,NSLASH
000000                     1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
000000                      INTEGER BP,COVER
000000                      IF(XLB(Q).EQ.0.AND.UB(Q).EQ.10000)RETURN
000014                      K=0
000015                      IF (Q.EQ.0)RETURN
000020                      J=LIST
000022                      P=J
000024                1     BP=P
000026                      P=PTR(P)
000031                      IF(P.EQ.0)GO TO 5
000033                      IF(COST(Q)-COST(P))2,3,4
000037                2     J=BP
000043                      IF(COVER(P,Q))1,1,6
000046                3     IF(COVER(P,Q).EQ.1)GO TO 6
000056                      IF(COVER(Q,P).NE.1)GO TO 1
000064                7     PTR(BP)=PTR(P)
000070                      P=BP
000072                      GO TO 1
000073                4     IF(COVER(Q,P))1,1,7
000076                5     P=LIST
000102                9     BP=P
000104                      P=PTR(P)
000107                      IF(P.EQ.0)GO TO 10
000111                      IF(COST(Q).LT.COST(P))GO TO 9
000116                      IF(SEC(Q).LE.SEC(P))GO TO 9
000123               10     PTR(BP)=Q
000126                      PTR(Q)=P
000131                      P=LIST
000133                      K=0
000134               11     P=PTR(P)
000137                      IF(P.EQ.0)GO TO 6
000141                      K=K+1
000142                      IF(K.LE.MAXSTR)GO TO 11
000145                      PTR(P)=0
000147                6     CONTINUE
000147                      RETURN
000150                8     PTR(P)=0
000152                      GO TO 5
000153                      END

GLOBAL DUMMIES

Q     174           LIST    175

COMMON
```

DIM	/.COMM./+0	XLB	/.COMM./+310	UB	/.COMM./+620	SPTR	/.COMM./+1130	PTR	/.COMM./+1440
MARK	/.COMM./+1750	TAU	/.COMM./+2260	FREE	/.COMM./+2261	TO	/.COMM./+2262	SEC	/.COMM./+2263
NOSEC	/.COMM./+2573	NCT	/.COMM./+2574	F1	/.COMM./+2575	F2	/.COMM./+3333	FO	/.COMM./+4071
FP	/.COMM./+4627	EVENT	/.COMM./+5365	SIZE	/.COMM./+47521	MAXSTR	/.COMM./+47522	NDIM	/.COMM./+47523
HSLASH	/.COMM./+47524	COST	/.COMM./+47606	T	/.COMM./+50116	TS	/.COMM./+50117	GLIST	/.COMM./+50120
OLIST	/.COMM./+50121	NC1	/.COMM./+50122	NC2	/.COMM./+50123	TLIST	/.COMM./+50124	C1	/.COMM./+50125
EVE	/.COMM./+50126	IF	/.COMM./+50210						

SUBPROGRAMS

COVER

SCALARS

INSERT	176	Q	174	K	177	J	200	LIST	175
P	201	BP	202	MAXSTR	47522	TAU	2260	FREE	2261
TO	2262	NOSEC	2573	NCT	2574	SIZE	47521	NDIM	47523
T	50116	TS	50117	GLIST	50120	OLIST	50121	NC1	50122
NC2	50123	TLIST	50124	C1	50125	IF	50210		

ARRAYS

DIM	0	XLB	310	UB	620	SPTR	1130	PTR	1440
MARK	1750	SEC	2263	F1	2575	F2	3333	FO	4071
FP	4627	EVENT	5365	HSLASH	47524	COST	47606	EVE	50126

```
000000                  INTEGER FUNCTION XMIN(XI,XJ)
000000                  IMPLICIT INTEGER(A-Z)
000000                  XMIN=XI
000002                  IF (XJ.LT.XI)XMIN=XJ
000007                   RETURN
000010                  END
```

GLOBAL DUMMIES

XI 46 XJ 47

SCALARS

XMIN 50 XI 46 XJ 47

APPENDIX B

TCPXLS Program Listing

```
000000                          INTEGER FUNCTION XMAX(XI,XJ)
000000                          IMPLICIT INTEGER(A-Z)
000000                          XMAX=XI
000002                           IF (XJ.GT.XI)XMAX=XJ
000007                          RETURN
000010                          END
```

GLOBAL DUMMIES

XI 46 XJ 47

SCALARS

XM 50 XI 46 XJ 47

```
000000                  INTEGER FUNCTION QLQ(I)
000000                  QLQ=I
000002                  RETURN
000003                  END
```

GLOBAL DUMMIES

I 23

SCALARS

QLQ 24 I 23

```
                END;
                XMX=0;
                DO I=1 TO NC;
                        XMX=MAX(BST(I),XMX);
                        END;
                DO I=1 TO NC;
                        IF XMX-BST(I)>.05 THEN
                        BST(I)=0;
                        END;
                PUT SKIP; L=TSIZE(IC)/2; IF L=EVE THEN
                        PUT EDIT(' D ',IC)(A,F(5));
                IF BST(IC)¬=0 THEN ICORR=ICORR+1;
                DO I=1 TO NC;
                IF BST(I)¬=0 THEN PUT EDIT(BST(I))(COL(10+5*I),F(4,2));
                END;
        END;
        PUT SKIP DATA(ICORR);
        END;
        XMATCH: PROC(C)RETURNS(FLOAT);
        DCL (C,CPX)FIXED BIN;
                DO CPX=1 TO 20 WHILE(CPTR(C,CPX)¬=0);
                ICX=CPTR(C,CPX);
                TOT=0; NV=0;
                DO WHILE(ICX¬=0);
        /*      CENTER = (TUB(ICX)+TLB(ICX))/2;
                L=EVENT(EVE,VAR(ICX));
                IF L¬=0 THEN
                IF L<CENTER THEN
                DO; NV=NV+1;
                        IF LB(ICX)=0 THEN TOT=TOT+1;
                        ELSE IF LB(ICX)<=L THEN
                        TOT=TOT+(L-LB(ICX))/(CENTER-LB(ICX));
                        END;
                ELSE IF L=CENTER THEN DO;TOT=TOT+1; NV=NV+1; END;
                ELSE DO;        NV=NV+1;
                        IF UB(ICX)=10000 THEN TOT=TOT+1;
                        ELSE IF UB(ICX)>=L THEN
                        TOT=TOT+(UB(ICX)-L)/(UB(ICX)-CENTER);
                        END; */
        L=EVENT(EVE,VAR(ICX));
        IF L>=0 THEN
        DO; IF LB(ICX)<=L & UB(ICX)>=L THEN TOT=TOT+.5;
            IF TLB(ICX)<=L & TUB(ICX)>=L THEN TOT=TOT+.5;
            NV=NV+1;
            END;
                        ICX=SPTR(ICX);  END;
                IF NV¬=0 THEN TOT=TOT/NV;
                DCL CSAT(20)STATIC;
                CSAT(CPX)=TOT;
```

```
STMT LEV NT

  1       0  |TCPXLST: PROC OPTIONS(MAIN);                                                    |
  2   1   0  |        PUT PAGE EDIT(' FORMULAS FROM INC EVENTS')(COL(20),A);                  |
  3   1   0  |        NC=20; NEVE=200;                                                        |
  5   1   0  |        BEGIN;                                                                  |
  6   2   0  |        DCL ((SPTR,VAR,LB,UB,TLB,TUB)(3000),CPTR(NC+1,20)                      |
             |        ,EVENT(NEVE,50),(LSIZE,TSIZE)(NC),CPX,                                  |
             |        (NUMCOV,TOTCOV,IND,NEWCOV)(NC,20),BCOV(20,NEVE),NEW(NEVE)               |
             |        ,PTR(50))FIXED BIN, LINE CHAR(100)VAR ,NUMB CHAR(4)VAR;                 |
  7   2   0  |        COV: PROC(CPX,L)RETURNS(FIXED BIN);                                     |
  8   3   0  |        DCL I STATIC, CPX FIXED BIN;                                            |
  9   3   0  |        I=CPTR(IC,CPX);                                                         |
 10   3   0  |        DO WHILE(I¬=0);                                                         |
 11   3   1  |        IF LB(I)>EVENT(L,VAR(I)) | UB(I)<EVENT(L,VAR(I)) THEN RETURN(0);       |
 12   3   1  |        I=SPTR(I);                                                              |
 13   3   1  |        END;                                                                    |
 14   3   0  |        RETURN(1); END COV;                                                     |
             |        /* READ IN COMPLEXES */                                                |
 16   2   0  |        CPTR=0;                                                                 |
 17   2   0  |        NUMCOV,TOTCOV,IND,NEWCOV=0;                                             |
 18   2   0  |        IC,CPX=1; CPTR(1,1)=1; NEXT=1;                                          |
 21   2   0  |S0:     GET LIST(VAR(NEXT));                                                   |
 22   2   0  |        IF VAR(NEXT)<=0 THEN GO TO S1;                                          |
 23   2   0  |        GET LIST(LB(NEXT),UB(NEXT));                                            |
 24   2   0  |        J=NEXT; NEXT=NEXT+1; SPTR(NEXT-1)=NEXT; GO TO S0;                       |
 28   2   0  |S1:     CPX=CPX+1; SPTR(J)=0;                                                  |
 30   2   0  |        IF VAR(NEXT)=-1 THEN DO;CPTR(IC,CPX-1)=0; IC=IC+1; CPX=1; END;         |
 35   2   0  |        CPTR(IC,CPX)=NEXT;                                                      |
 36   2   0  |        IF IC<=NC THEN GO TO S0;                                                |
             |                                                                                |
             |/*   CALCULATE TRIMMED COMPLEXES   */                                        |
             |                                                                                |
 37   2   0  |        GET FILE(LEARN) LIST(LSIZE);                                            |
 38   2   0  |TCPX: DO IC=1 TO NC;                                                            |
 39   2   1  |        BCOV=0;                                                                 |
 40   2   1  |        GET FILE(LEARN) EDIT((EVENT(J,*) DO J=1 TO LSIZE(IC)))                 |
             |        (COL(3),F(3),12 F(1),F(3),3 F(1),F(3),F(4),F(1),5 F(3),25 F(1)         |
             |        );                                                                      |
 41   2   1  |        DO CPX=1 TO 20 WHILE(CPTR(IC,CPX)¬=0);                                  |
 42   2   2  |                DO L=1 TO LSIZE(IC); BCOV(CPX,L)=COV(CPX,L); END;               |
 45   2   2  |                END;                                                            |
             |                                                                                |
 46   2   1  |        DO CPX=1 TO 20 WHILE(CPTR(IC,CPX)¬=0);                                  |
 47   2   2  |        FIRST=1;                                                                |
 48   2   2  |                DO J=1 TO LSIZE(IC);                                            |
 49   2   3  |                IF BCOV(CPX,J)=1 THEN DO;                                       |
 50   2   4  |                ICX=CPTR(IC,CPX);                                               |
```

```
 51   2  4  |                DO WHILE(ICX¬=0);                                                    |
 52   2  5  |                IF FIRST=1 THEN                                                     |
             |                TUB(ICX),TLB(ICX)=EVENT(J,VAR(ICX));                                |
 53   2  5  |                ELSE DO;                                                            |
 54   2  6  |                TUB(ICX)=MAX(TUB(ICX),EVENT(J,VAR(ICX)));                           |
 55   2  6  |                TLB(ICX)=MIN(TLB(ICX),EVENT(J,VAR(ICX)));                           |
 56   2  6  |                END;                                                                |
 57   2  5  |                ICX=SPTR(ICX);                                                      |
 58   2  5  |                END;                                                                |
 59   2  4  |                FIRST=0;                                                            |
 60   2  4  |                END;                                                                |
 61   2  3  |                END;                                                                |
 62   2  2  |                END;                                                                |
 63   2  1  |                NCPX=CPX-1; NEW=0;                                                  |
 65   2  1  |                DO CPX=1 TO NCPX;                                                   |
 66   2  2  |                DO L=1 TO LSIZE(IC);                                                |
 67   2  3  |                IF BCOV(CPX,L)=1 THEN                                               |
 68   2  4  |                DO; IF NEW(L)=0 THEN NEWCOV(IC,CPX)=NEWCOV(IC,CPX)+1;               |
 69   2  4  |                NUMCOV(IC,CPX)=NUMCOV(IC,CPX)+1;                                    |
 70   2  4  |                NEW(L)=1;                                                           |
 71   2  4  |                END;                                                                |
 72   2  3  |                END;                                                                |
 73   2  2  |                END;                                                                |
 74   2  1  |        TOTCOV(IC,1)=NEWCOV(IC,1);                                                  |
 75   2  1  |        DO CPX=2 TO NCPX;                                                           |
 76   2  2  |                TOTCOV(IC,CPX)=TOTCOV(IC,CPX-1)+NEWCOV(IC,CPX);                     |
 77   2  2  |                END;                                                                |
 78   2  1  |        DO CPX=1 TO NCPX;                                                           |
 79   2  2  |                DO L=1 TO LSIZE(IC);                                                |
 80   2  3  |                IF BCOV(CPX,L)¬=1 THEN GO TO L1; DO K=1 TO NCPX;                    |
 82   2  4  |                IF CPX¬=K & BCOV(K,L)=1 THEN GO TO L1;END;                          |
 84   2  3  |                IND(IC,CPX)=IND(IC,CPX)+1;                                          |
 85   2  3  |        L1:      END;                                                               |
 86   2  2  |                END;                                                                |
 87   2  1  |        PUT SKIP(5) EDIT(' THE FOLLOWING',NCPX,                                    |
             |                ' COMPLEXES COVER SET ',IC,'COV  NEW  IND  TOT')                    |
             |        (COL(20),A,F(4),A,F(4),COL(100),A);                                         |
 88   2  1  |LST:    DO CPX=1 TO NCPX; M=0;                                                      |
 90   2  2  |        PUT SKIP(2);                                                                |
 91   2  2  |                LINE='COMPLEX: ';                                                   |
 92   2  2  |                ICPX=CPTR(IC,CPX);                                                  |
 93   2  2  |                DO NV=1 TO 50 WHILE(ICPX¬=0);                                       |
 94   2  3  |                PTR(NV)=ICPX; ICPX=SPTR(ICPX); END;                                 |
 97   2  2  |                NV=NV-1;                                                            |
 98   2  2  |                DO I=1 TO NV-1;                                                     |
 99   2  3  |                        DO J=I+1 TO NV;                                             |
100   2  4  |                        IF VAR(PTR(I))>VAR(PTR(J)) THEN                             |
101   2  5  |                        DO; L=PTR(I); PTR(I)=PTR(J); PTR(J)=L; END;                 |
105   2  4  |                        END;                                                        |
```

```
106   2  3  |                     END;                                                                  |
107   2  2  |                  DO J=1 TO NV; LINE=LINE || ' ('; I=PTR(J);                               |
110   2  3  |                     IF LB(I)>0 THEN                                                       |
111   2  4  |                     DO; PUT STRING(NUMB)EDIT(LB(I))(F(4));                                |
112   2  4  |                     LINE=LINE || NUMB;                                                    |
113   2  4  |                     IF TLB(I)¬=LB(I) THEN DO;                                             |
114   2  5  |                     PUT STRING(NUMB)EDIT(TLB(I))(F(4));                                   |
115   2  5  |                     LINE=LINE || ',' || NUMB;                                             |
116   2  5  |                     END;                                                                  |
117   2  4  |                     LINE=LINE||'<=';                                                      |
118   2  4  |                     END;                                                                  |
119   2  3  |                     PUT STRING(NUMB)EDIT(VAR(I))(F(2));                                   |
120   2  3  |                     LINE=LINE || 'X' || SUBSTR(NUMB,1,2);                                 |
121   2  3  |                     IF UB(I)<10000 THEN                                                   |
122   2  4  |                     DO; LINE=LINE || '<=';                                                |
123   2  4  |                     IF TUB(I)¬=UB(I) THEN DO;                                             |
124   2  5  |                     PUT STRING(NUMB)EDIT(TUB(I))(F(4));                                   |
125   2  5  |                     LINE=LINE|| NUMB || ',';                                              |
126   2  5  |                     END;                                                                  |
127   2  4  |                     PUT STRING(NUMB)EDIT(UB(I))(F(4));                                    |
128   2  4  |                     LINE=LINE || NUMB;                                                    |
129   2  4  |                     END;                                                                  |
130   2  3  |                     LINE=LINE || ')';                                                     |
131   2  3  |                     IF LENGTH(LINE)>75 THEN DO;                                           |
132   2  4  |                             PUT SKIP EDIT(LINE)(A);                                       |
133   2  4  |                                     LINE='          ';                                   |
134   2  4  |                             IF M=0 THEN DO;                                               |
135   2  5  |                             M=1;                                                          |
136   2  5  |                     PUT EDIT(NUMCOV(IC,CPX),NEWCOV(IC,CPX),                               |
      |     |                  IND(IC,CPX),TOTCOV(IC,CPX))(COL(100),4(F(3),X(2)));                      |
137   2  5  |                     END;                                                                  |
138   2  4  |                     END; END; PUT SKIP EDIT(LINE)(A);                                     |
141   2  2  |                        END LST;                                                           |
142   2  1  |                     END TCPX;                                                             |
143   2  0  |        CONF:;  DCL BST(NC);                                                               |
145   2  0  |        GET FILE(TEST) LIST(TSIZE);                                                        |
146   2  0  | DO IC=1 TO NC; GET FILE(TEST) EDIT((EVENT(J,*)DO J=1 TO TSIZE(IC)))                       |
      |     |        (COL(3),F(3),12 F(1),F(3),3 F(1),F(3),F(4),F(1),5 F(3),25 F(1)                    |
      |     |        );                                                                                 |
148   2  1  |        PUT PAGE;                                                                          |
149   2  1  |        PUT EDIT(('D',II DO II=1 TO NC))(COL(5*II+11),A,F(2));                             |
150   2  1  |        DO J=1,2;                                                                          |
151   2  2  |        PUT SKIP EDIT(('_' DO I=1 TO 120))(A);                                             |
152   2  2  |        END;                                                                               |
153   2  1  |        ICORR=0;                                                                           |
154   2  1  |        DO EVE=1 TO TSIZE(IC);                                                             |
155   2  2  |                BST=0;                                                                     |
156   2  2  |                DO C=1 TO NC;                                                              |
157   2  3  |                BST(C)=XMATCH(C);                                                          |
```

```
.MT LEV NT

 1       0  |       PBLOG: PROC OPTIONS(MAIN);
 2    1  0  |       NC=20; NEVE=1112;
 4    1  0  |       BEGIN;
 5    2  0  |       DCL ((SPTR,VAR,LB,UB)(4000),(LSIZE,TSIZE)(NC),
               |       CPTR(120),CLASS(120),EVE(50),CPX)FIXED BIN,CCPROB(NC,120);
 6    2  0  |       DCL XM(120),BST(NC),(DTOT,DECIS)FIXED BIN;
 7    2  0  |       ITOT=0; DTOT=0;
 9    2  0  |       MATCH: PROC;
10    3  0  |               DO CPX=1 TO NCPX;
11    3  1  |                       J=CPTR(CPX);
12    3  1  |                       X=0; NV=0;
14    3  1  |                       DO WHILE(J¬=0);
15    3  2  |                       IF EVE(VAR(J))>0 THEN NV=NV+1;
15    3  2  |                       IF LB(J)<=EVE(VAR(J))&UB(J)>=EVE(VAR(J)) THEN
17    3  2  |                       X=X+1; J=SPTR(J); END;
19    3  1  |                       IF NV=0 THEN XM(CPX)=0;
20    3  1  |                               ELSE XM(CPX)=X/NV; END MATCH;
22    2  0  |       IC=1; CPTR=0;CPTR(1)=1; CCPROB=0; NEXT=1; CPX=1;
28    2  0  |       S0:       GET LIST(VAR(NEXT));
29    2  0  |               IF VAR(NEXT)<=0 THEN GO TO S1;
30    2  0  |               GET LIST(LB(NEXT),UB(NEXT));J=NEXT; NEXT=NEXT+1;
33    2  0  |               SPTR(J)=NEXT; GO TO S0;
35    2  0  |       S1: SPTR(J)=0; CLASS(CPX)=IC; IF VAR(NEXT)=0 THEN CPX=CPX+1;
38    2  0  |               CPTR(CPX)=NEXT; IF VAR(NEXT)=-1 THEN IC=IC+1;
40    2  0  |               IF IC<=NC THEN GO TO S0;
41    2  0  |               NCPX=CPX-1; GET FILE(LEARN) LIST(LSIZE);
43    2  0  |       DO IEVE=1 TO NEVE;
44    2  1  |               GET FILE(LEARN) EDIT(IC,EVE)
               |       (COL(1),F(2),F(3),12 F(1),F(3),3 F(1),F(3),F(4),F(1),5 F(3)
               |       ,25 F(1));
45    2  1  |       IF IC=0 THEN IC=20;
46    2  1  |               DO I=1 TO NCPX;
47    2  2  |                       J=CPTR(I); DO WHILE(J¬=0);
49    2  3  |                       IF LB(J)>EVE(VAR(J))|UB(J)<EVE(VAR(J)) THEN
50    2  3  |                       GO TO S4; J=SPTR(J); END;
52    2  2  |                       CCPROB(IC,I)=CCPROB(IC,I)+1;
53    2  2  |                       S4: END;
54    2  1  |                END;
55    2  0  |       DO CPX=1 TO NCPX;
56    2  1  |       XX=SUM(CCPROB(*,CPX));
57    2  1  |       CCPROB(*,CPX)=CCPROB(*,CPX)/XX;END;
59    2  0  |       PUT PAGE EDIT(('D',I DO I=1 TO NC))
               |               (COL(10+5*I),A,F(2));
60    2  0  |       DO CPX=1 TO NCPX;
61    2  1  |               PUT SKIP EDIT((CCPROB(I,CPX) DO I=1 TO NC))
               |                       (COL(10+5*I),F(4,3));
```

```
62   2  1  |              PUT SKIP(0) EDIT('____')(COL(  5*CLASS(CPX)),A);
63   2  1  |              END;
64   2  0  |       GET FILE(TEST) LIST(TSIZE);
65   2  0  |       CCPROB=(CCPROB=0)*2E-3+CCPROB;
66   2  0  |       DO IC=1 TO NC;
67   2  1  |              PUT PAGE EDIT(('D',I DO I=1 TO NC))(COL(10+5*I),A,F(2));
68   2  1  |              PUT EDIT(('_' DO I=1 TO 240))(SKIP,120 A);
69   2  1  |              IC2=TSIZE(IC)/2; ICORR=0; DECIS=0;
72   2  1  |              DO IE=1 TO TSIZE(IC);
73   2  2  |              GET FILE(TEST) EDIT(J,EVE)
                        |       (COL(1),F(2),F(3),12 F(1),F(3),3 F(1),F(3),F(4),F(1),5 F(3)
74   2  2  |       ,25 F(1)); CALL MATCH; X=0;
76   2  2  |              DO I=1 TO NCPX; X=MAX(X,XM(I)); END;
79   2  2  |              BST=0;
80   2  2  |              DO J=1 TO NCPX;
81   2  3  |              IF X-XM(J)<.05 THEN
82   2  4  |              DO L=1 TO NC; BST(L)=BST(L)+CCPROB(L,J); END;
84   2  3  |              END;
85   2  2  |              X=0;
86   2  2  |              DO I=1 TO NC; X=MAX(X,BST(I)); END;
89   2  2  |       Y=0;DO I=1 TO NC;IF BST(I)~=X THEN Y=MAX(Y,BST(I));END;
93   2  2  |              PUT SKIP; DO I=1 TO NC; IF BST(I)>Y-.01 THEN DO;
96   2  4  |                      PUT EDIT(BST(I)/X)(COL(10+5*I),F(4,2));
97   2  4  |               DECIS=DECIS+1;
98   2  4  |                      IF I=IC THEN ICORR=ICORR+1; END; END;
101  2  2  |              IF IC2=IE THEN PUT SKIP(0) EDIT('D',IC)(A,F(6));
     2  2  |              END; PUT SKIP DATA (ICORR,DECIS); DTOT=DTOT+DECIS;
105  2  1  |              ITOT=ITOT+ICORR; END;    PUT DATA(ITOT,DTOT);
108  2  0  |              END PBLOG;
```

APPENDIX C

PBLOG Program Listing

```
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
******* AQPL      *** 101 ***** BIN 23 ***** [4113,6574] FORSBURG ***** BIN 23 *
```

AQPLUS 5/13/76

```
****************************************  H A S P  (3.1)  S Y S T E M  L O G
```

Scott Forsburg

```
$ 13.19.41 JOB 101 AQPL      ON PRINTER2 BIN=23
```

Log 4113, 6574 Borg

```
****************************  O S / M V T  (21.7)  S Y S T E M  M E S S A G E
```

Ex AQPLUS

```
LPTSPL VERSION 100(2016) RUNNING ON XX0003
 *START* USER FORSBURG [4113,6574] JOB  AQPL SEQ. 2758 DATE 13-MAY-76 13:19:18
```

```
          IMPLICIT INTEGER (A-Z)
          COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
         1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
         2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
         3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
         5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
         4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
          REAL U
          TYPE 100
    100   FORMAT(//,' WELCOME TO THE WONDERFUL WORLD OF AQPLUS.')
          CALL DEFINE FILE(1,31,NEXT1,'JIM.DAT',"4113,"6574)
          CALL DEFINE FILE(20,51,NEXT20,'DATA.TOT',"4113,"6574)
          DO 10 I=1,200
          U(I)=0.5
     10   FORC(I)=0
          PREVC1=0
          LAST=1
          PUT(1,1)=-2
          CALL AQPARM
    101   TYPE 102
    102   FORMAT(///' AQPLUS COMMAND LEVEL -- CHOOSE OPTION, OR H FOR HELP'
          GO TO 115
    105   TYPE 110
    110   FORMAT(' CHOOSE ONE (BY FIRST LETTER):',/,16X,
         1'MODIFY USEFULNESS VALUES',/,16X,'AQVAL PARAMETERS',/,16X,
         2'FORCE CHOICES OF FEATURES',/,16X,
         3'RUN AQVAL',/,16X,'PRINT BEST FORMULA',/,16X,'QUIT')
    115   ACCEPT 120,OPTION
    120   FORMAT(A1)
```

```
01500           IF(OPTION.EQ.'H')GO TO 109
01600           IF(OPTION.EQ.'M')CALL MODIFY
01700           IF(OPTION.EQ.'A')CALL AQPARM
01800           IF(OPTION.EQ.'F')CALL FORCE
01900           IF(OPTION.EQ.'C')CALL CREATE
01950           IF(OPTION.EQ.'P')CALL PRBEST
02000           IF(OPTION.EQ.'R')GO TO 180
02100           IF(OPTION.EQ.'Q')GO TO 900
02200           GO TO 101
02300     180   CALL SELECT(NDIM-NFORC)
02350           CALL PROJEC
02400           CALL AQ
02500           IF(DISJNT.EQ.1)GO TO 300
02600           TYPE 260
02700     260   FORMAT(/,' REPRESENTATION IS NON-DISJOINT.  NO CHANGES MADE.')
02800           GO TO 101
02900     300   CALL USEFUL
03000           GO TO 101
03100     900   CLOSE(UNIT=1)
03200           CLOSE(UNIT=20)
03300           TYPE 999
03400     999   FORMAT(//,' Y''ALL COME BACK NOW, HEAR?',25(/))
03500           END
```

```
00100           SUBROUTINE AQPARM
00120           IMPLICIT INTEGER (A-Z)
00140           COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
00160          1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
00180          2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
00200          3IEL,DISJNT,U(200),USE1(200),N(15),M(3),FORC(200),PREVC,
00220          5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
00240          4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
00350           DOUBLE PRECISION HI
00400           DOUBLE PRECISION PNAME(12,2), PACK
00500           INTEGER CHAR(20), COMP(10), VALUE(12)
00600           EQUIVALENCE(VALUE, DIMTOT)
00750           DATA(VALUE(I),I=1,12)/50,204,5,1,1,1,10,1,112,10,0,0/
00800           DATA((PNAME(I,K),K=1,2),I=1,12)
00900          1/'DIMTOT','<= 200','NCT    ',' ','NDIM   ',' ','<= 30 ','MAXSTR'
01000          2'      ','NOSEC ','0 OR 1','C1     ',' ','NC2    ','<= 100',
01100          3'IB     ',' ','<= NCT','NC1    ',' ','<= NCT','NUSE1  ',' ','<= NC1&100',
01200          4'RAND   ',' ','0 OR 1',' ','     ',' ','     '/
01250           NUMPAR=11
01260           TYPE 110
01270     110   FORMAT(/,' ROUTINE TO ACCEPT AQVAL PARAMETERS',/)
01300     120   TYPE 125
01400     125   FORMAT(' ENTER (PARM:VALUE*), (S)HOW VALUES, OR (Q)UIT')
01500     126   ACCEPT 130,CHAR
01600     130   FORMAT(20A1)
01700           IF(CHAR(1).EQ.'Q')GO TO 999
01800           IF(CHAR(1).EQ.'S')GO TO 100
01900           DO 140 I=1,11
01950           NUMB=I-1
02000     140   IF(CHAR(I).EQ.':')GO TO 150
02100           TYPE 145
02200     145   FORMAT(' '':'' NOT FOUND IN SCAN RANGE.  TRY AGAIN.')
02300           GO TO 120
02400     150   ENCODE(10,130,PACK)(CHAR(I),I=1,NUMB)
02500           DO 160 J=1,NUMPAR
02550           JNUMB=J
02600     160   IF(PACK.EQ.PNAME(J,1))GO TO 170
02700           TYPE 165
02800     165   FORMAT(' PARAMETER NAME INVALID.  TRY AGAIN.')
02900           GO TO 120
03000     170   NUMB=NUMB+2
03025           KNUMB=NUMB
```

```
03050           DO 180 K=NUMB,20
03100           IF(CHAR(K).EQ.'*')GO TO 195
03200           IF(CHAR(K).EQ.' ')GO TO 180
03300           IF(CHAR(K).GT.'9')GO TO 220
03400           IF(CHAR(K).LT.'0')GO TO 220
03500     180   KNUMB=K+1
03600     195   IF(KNUMB-NUMB.GT.6.OR.KNUMB-NUMB.LT.1)GO TO 220
03700           ENCODE(10,130,PACK)(CHAR(I),I=NUMB,KNUMB-1)
03800           DECODE(6,200,PACK)HOLD
03900     200   FORMAT(I6)
04000           VALUE(JNUMB)=HOLD/(10**(6+NUMB-KNUMB))
04100           TYPE 210,PNAME(JNUMB,1),VALUE(JNUMB)
04200     210   FORMAT(1X,A10,' = ',I6)
04300           GO TO 126
04400     220   TYPE 230
04500     230   FORMAT(' PARAMETER VALUE INVALID.  TRY AGAIN.')
04600           GO TO 120
04700     100   TYPE 105

04800     105   FORMAT(/,' CURRENT PARAMETER VALUES ARE:',//)
04900           TYPE 115,((PNAME(J,1),VALUE(J),PNAME(J,2)),J=1,NUMPAR)
05000     115   FORMAT(1X,A10,' = ',I6,10X,'(',A10,')')
05040           TYPE 116
05080     116   FORMAT(//)
05100           GO TO 120
05200     999   MAXSTR=0
05300           IF(MAX1.GT.0)MAXSTR=MAX1-1
05400           RETURN
05500           END


00100           SUBROUTINE FORCE
00120           IMPLICIT INTEGER (A-Z)
00140           COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
00160          1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
00180          2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
00200          3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
00220          5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
00240          4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
00260           TYPE 90
00280      90   FORMAT(/,' ROUTINE TO FORCE THE SELECTION OF FEATURES',/)
00400      10   TYPE 100
00500     100   FORMAT(/,' YOU HAVE SELECTED THE FOLLOWING FEATURES:')
00600           DO 150 I=1,DIMTOT
00700     150   IF(FORC(I).EQ.1)TYPE 110,I
00800     110   FORMAT(2X,I3)
00900     115   TYPE 120
01000     120   FORMAT(/,' ENTER FEATURE NUMBER (I3) TO CHANGE SELECTION ',
01100          1'STATUS, L TO LIST.',/,' Q TO QUIT, OR C TO CLEAR.',/)
01140     125   TYPE 126
01180     126   FORMAT($,' ENTER FEATURE I3 OR OPTION =')
01200           CALL GETINT(1)
01300           IF(M(1).NE.'E    ')GO TO 128
01320           IF(N(1).EQ.'L')GO TO 10
01340           IF(N(1).EQ.'Q')GO TO 330
01360           IF(N(1).EQ.'C')GO TO 220
01380           GO TO 200
01400     128   DECODE(3,130,M)I
01500     130   FORMAT(I3)
01700           IF(I.GT.DIMTOT.OR.I.EQ.0)GO TO 200
01800           FORC(I)=1-FORC(I)
01840           TYPE 140,I
01880     140   FORMAT($,' FEATURE ',I3,' CHANGED.  ')
01900           GO TO 125
02000     200   TYPE 210
02100     210   FORMAT(' THAT FEATURE DOESN''T EXIST.  ')
```

```
02200           GO TO 125
02300      220  DO 230 I=1,DIMTOT
02400      230  FORC(I)=0
02500           GO TO 115
02600      330  NFORC=0
03300           DO 350 I=1,DIMTOT
03400           NSEL(I)=FORC(I)
03500      350  NFORC=NFORC+FORC(I)
03600           TYPE 360,NFORC
03700      360  FORMAT(/,' YOU HAVE SELECTED ',I3,' FEATURES.')
03800           IF(NFORC.LE.NDIM)RETURN
03900           TYPE 370,NDIM
04000      370  FORMAT(' YOU ARE ONLY ALLOWED ',I3,' FEATURES. HOWEVER. EITHER '
04100          1'INCREASE NDIM.',/,' OR RE-ENTER THIS ROUTINE TO CHANGE YOUR ',
04200          2'CHOICES.')
04300           RETURN
04400           END
```

```
00100           SUBROUTINE PROJEC
00120           IMPLICIT INTEGER (A-Z)
00140           COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
00160          1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
00180          2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
00200          3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
00220          5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
00240          4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
00300           INTEGER IN(200),OUT(30),IC,I,J,NEXT1,NEXT20,DIMTOT
00400           REAL RAN
01200      C    PROJECT EVENTS
01250           TYPE 90
01275       90  FORMAT(//,' EVENT PROJECTION IN PROGRESS.  PLEASE WAIT.',//)
01300           DO 20 J=1,NCT
01400           READ(20#J)IC,(IN(K),K=1,50)
01500           DO 10 I=1,NDIM
01600       10  OUT(I)=IN(NSEL(I))
01800       20  WRITE(1#J)IC,OUT
02300           DO 30 I=IB,IB+USE1-1
02400           J=I
02500           IF(RAND.EQ.1)J=RAN(J)*NC1+IB
02600       30  READ(1#J)IC,(EVENT(I-IB+1,JJ),JJ=1,30)
02700           DO 40 I=1,NC1
02800       40  F1(I)=I
02850           TYPE 101
02860      101  FORMAT(//,' EVENT PROJECTION COMPLETED.',///,
02870          1' AQVAL PROCEDURE BEGUN.      PATIENCE IS A VIRTUE.',/)
02900           RETURN
03000           END
```

```
00100           SUBROUTINE MODIFY
00110           IMPLICIT INTEGER (A-Z)
00120           COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
00130          1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
00140          2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
00150          3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
00160          5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
00170          4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
00200           REAL U,VALUE
00300           INTEGER DIMTOT,MESS1,FNO,MODE
00400           FNO=0
00420           TYPE 100
00440      100  FORMAT(/,' ROUTINE TO MODIFY USEFULNESS VALUES',/)
00500           TYPE 110,DIMTOT
00600      110  FORMAT(/,' REMINDER: YOU ARE WORKING WITH ',I3,' FEATURES.')
00700      590  TYPE 600
```

```
00800     600  FORMAT(/,' ALTER/DISPLAY MODE, INCREMENT, HELP, OR QUIT?',
00820         1/,$,' A/ /H/Q     ')
00900          ACCEPT 610,MODE
01000     610  FORMAT(A5)
01100          IF(MODE.EQ.'H')GO TO 200
01200          IF(MODE.EQ.' ')GO TO 525
01300          IF(MODE.EQ.'Q')GO TO 999
01400          IF(MODE.NE.'A')GO TO 590
01500     500  TYPE 510
01600     510  FORMAT(/,$,' ENTER FEATURE NUMBER    (I3)  =')
01700          ACCEPT 60,N
01800          DO 154 I=1,3
01900          IF(N(I).EQ.' ')GO TO 154
02000          IF(N(I).GT.'9')GO TO 540
02100          IF(N(I).LT.'0')GO TO 540
02200     154  CONTINUE
02300          ENCODE(3,30,M)N
02400     30   FORMAT(3A1)
02500          DECODE(3,520,M)FNO
02600     520  FORMAT(I3)
02700          IF(FNO.EQ.0)GO TO 312
02800          GO TO 530
02900     525  FNO=FNO+1
03000     530  IF(FNO.LE.DIMTOT)GO TO 45
03100     540  TYPE 550
03200     550  FORMAT(' FEATURE NUMBER INVALID.  CHOOSE ANOTHER. ')
03300          GO TO 500
03400     312  TYPE 314
03500     314  FORMAT(/,$,' DISPLAY OR ALTER ALL VALUES?   D/A  ')
03600          ACCEPT 316,MODE
03700     316  FORMAT(A1)
03800          IF(MODE.NE.'A')GO TO 320
03900          TYPE 425
04000     425  FORMAT($,/,' ENTER VALUE TO REPLACE ALL OTHERS =')
04100          GO TO 52
04200     320  IF(MODE.NE.'D')GO TO 312
04250          TYPE 333
04300     320  TYPE 330,((I,U(I)),I=1,DIMTOT)
04400     330  FORMAT(4(' U(',I3,')=',F6.4,4X))
04440          TYPE 333
04480     333  FORMAT(/)
04500          GO TO 590
04600     45   TYPE 50,FNO,U(FNO)
04700     50   FORMAT(/,$,' PRESENTLY U(',I3,')=',F6.4,',  NEW VALUE =')

04800     52   ACCEPT 60,N
04900     60   FORMAT(15A1)
05000          NO=0
05100          NS=0
05200          DO 54 I=1,6
05300          IF(N(I).EQ.',')GO TO 56
05400          IF(N(I).EQ.' ')GO TO 58
05500          IF(N(I).GT.'9')GO TO 290
05600          IF(N(I).LT.'0')GO TO 290
05700          GO TO 54
05800     58   NS=NS+1
05900          GO TO 54
06000     56   NO=NO+1
06100     54   CONTINUE
06200          IF(NO.GT.1)GO TO 290
06300          IF(NS.EQ.6)GO TO 590
06400          ENCODE(6,60,M)N
06500          DECODE(6,62,M)VALUE
06600     62   FORMAT(F6.5)
06700          IF(VALUE.GT.1.0)GO TO 290
06800          IF(FNO.EQ.0)GO TO 70
06900          U(FNO)=VALUE
```

```
07000           TYPE 65,FNO,U(FNO)
07100      65   FORMAT(/,' CHANGE MADE: U(',I3,')= ',F6.4)
07200           GO TO 590
07300      70   DO 80 I=1,DIMTOT
07400      80   U(I)=VALUE
07500           TYPE 85,VALUE
07600      85   FORMAT(/,' CHANGE MADE: ALL VALUES SET TO ',F6.4)
07700           GO TO 590
07800     300   IF(FNO.EQ.0)GO TO 320
07900           TYPE 310,FNO,U(FNO)
08000     310   FORMAT(/,' U(',I3,')= ',F6.4)
08100           GO TO 500
08200     200   TYPE 210
08300     210   FORMAT(/,' HELPFUL HINTS:',/,
08400          1'   FEATURE NUMBER 0 = ALL FEATURES.',/,
08500          2'   YOU WILL BE NOTIFIED WHENEVER (%) A CHANGE IS MADE.',/,
08540          2'   INCREMENT ACCESSES THE NEXT FEATURE NUMBER.',/,
08600          3'   VALUE = [RETURN] CHANGES NOTHING.',/)
08700           GO TO 590
08800     290   TYPE 295
08900     295   FORMAT(' VALUE INVALID.  NO CHANGES MADE.')
09000           GO TO 590
09100     999   RETURN
09200           END


00100           SUBROUTINE SELECT(NN)
00200           IMPLICIT INTEGER (A-Z)
00210           COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
00220          1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
00230          2SIZE,HSLASH(30),COST(200),T,TS,GLIST,QLIST,TLIST,EVE(30),
00240          3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
00250          5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
00260          4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
00300           REAL R(100),RAN,RAND,U,RAND1,GTOT,TOT
00330           DO 5 I=1,DIMTOT
00360           IF(FORC(I).EQ.1)GO TO 3
00390           NSEL(I)=0
00420           GO TO 5
00450        3  NSEL(I)=1
00480        5  CONTINUE
00490           IF(NN.LE.0)RETURN
00500           GTOT=0.
00600           DO 10 I=1,DIMTOT
00700     10    IF(NSEL(I).EQ.0)GTOT=GTOT+U(I)
00800           DO 20 I=1,NN
00900     20    R(I)=RAN(0.0)
01000           DO 30 I=1,NN-1
01100           DO 30 J=1,NN-I
01200           JP=J+1
01250           IF(R(J).LE.R(JP))GO TO 30
01300           RAND=R(J)
01400           R(J)=R(JP)
01600           R(JP)=RAND
01900     30    CONTINUE
02000           TOT=0.
02100           I=0
02200           DO 70 J=1,NN
02300           RAND=R(J)*GTOT
02400     40    IF(RAND.GE.TOT)GO TO 50
02500           IF(NSEL(I).EQ.0)TOT=TOT-U(I)
02600           I=I-1
02700           GO TO 40
03000     50    IF(RAND.LE.TOT)GO TO 60
03100           I=I+1
03200           IF(NSEL(I).EQ.0)TOT=TOT+U(I)
03300           GO TO 50
```

```
03400      60   NSEL(I)=1
03500           TOT=TOT-U(I)
03600           GTOT=GTOT-U(I)
03700      70   CONTINUE
03800      C    CONDENSE ARRAY NSEL
03900           K=1
04000           DO 100 J=1,DIMTOT
04100           IF(NSEL(J).EQ.0)GO TO 100
04200           NSEL(K)=J
04300           K=K+1
04400     100   CONTINUE
04500           TYPE 200
04600     200   FORMAT(/,' THE FEATURES CHOSEN ARE:',/)
04700           TYPE 220,(NSEL(I),I=1,NDIM)
04800     220   FORMAT(10(4X,I3))
04900           RETURN
05000           END


00100           SUBROUTINE USEFUL
00200           IMPLICIT INTEGER (A-Z)
00220           COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
00240          1TAU,FREE,TO,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
00260          2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
00280          3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
00300          5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
00320          4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
00400           REAL C,PREVC,U
00440           TYPE 50
00480      50   FORMAT(/,' USEFULNESS VALUES BEING COMPUTED')
00500           NCLASS=0
00600           DO 10 I=1,NCT
00700           READ(1#I)IC,EVE
00800           IF(CONT(0,TLIST,0).EQ.1)GO TO 20
00900           IF(IC.EQ.C1)NCLASS=NCLASS-1
01000           GO TO 10
01100      20   IF(IC.EQ.C1)GO TO 30
01200           NCLASS=NCLASS-1
01300           GO TO 10
01320      30   NCLASS=NCLASS+1
01340      10   CONTINUE
01400     100   IF(NCLASS.LE.0)NCLASS=1
01450           J=3
01500           SELCNT=0
01600           CPXPTR=0
01700           EVE(1)=0
01800           NFEAT=0
01900           X=TLIST
01920      C    SEE JIM LARSON'S AQ.FOR DOCUMENTATION FOR AN EXPLANATION OF THE
01940      C         DATA STRUCTURES USED HERE.
01960      C
01980      C    PUT FORMULA INTO ARRAY 'PUT'.
01990           IF(SPTR(X).EQ.0.AND.PTR(X).EQ.0)GO TO 500
02000      120  SELCNT=SELCNT+1
02100           IF(J.GT.95)GO TO 130
02105           DIMENS=NSEL(DIM(X))
02110           PUT(J,3-LAST)=DIMENS
02120           PUT(J+1,3-LAST)=XLB(X)
02130           PUT(J+2,3-LAST)=UB(X)
02140           IF(X.NE.TLIST)J=J+3
02150     130   DO 150 I=1,NFEAT
02200     150   IF(EVE(I).EQ.DIMENS)GO TO 200
02400           NFEAT=NFEAT+1
02500           EVE(NFEAT)=DIMENS
02700     200   IF(CPXPTR.EQ.0)CPXPTR=PTR(X)
02800           IF(SPTR(X).EQ.0)GO TO 300
02900           X=SPTR(X)
```

```
03000           GO TO 120
03100     300   IF(CPXPTR.EQ.0)GO TO 500
03200           X=CPXPTR
03210           CPXPTR=0
03220           PUT(J,3-LAST)=-1
03230           IF(J.LT.96)J=J+1
03300           GO TO 120
03310     500   PUT(J,3-LAST)=-2
03400           C=NFEAT*SELCNT*NC1/NCLASS
03500           PUT(1,3-LAST)=C
03510           PUT(2,3-LAST)=C1
03520           IF(C1.EQ.PREVC1)GO TO 550

03530           PREVC1=C1
03540           GO TO 600
03550     550   IF(C.LE.PREVC)GO TO 600
03600           C=PREVC/C
03625           C=C**0.5
03650           IF(C.LT.0.1)C=0.1
03700           GO TO 610
03800     600   PREVC=C
03900           C=1.0
04000     610   DO 800 I=1,NDIM
04100           DIMENS=NSEL(I)
04140           TYPE 621,DIMENS,U(DIMENS)
04180     621   FORMAT($,' X',I3,10X,'OLD VALUE=',F6.4,8X,'NEW VALUE=')
04200           DO 620 J=1,NFEAT
04300     620   IF(EVE(J).EQ.DIMENS)GO TO 630
04350     C     IF FEATURE NOT USED IN RULE:
04400           U(DIMENS)=U(DIMENS)*C*FLOAT(DIMTOT-NDIM)/FLOAT(DIMTOT)
04500           GO TO 626
04550     C     IF FEATURE USED IN RULE:
04600     630   U(DIMENS)=U(DIMENS)+(1.-U(DIMENS))*C*FLOAT(NDIM)/FLOAT(DIMTOT)
04720     626   TYPE 622,U(DIMENS)
04740     622   FORMAT(F6.4)
04745     800   CONTINUE
04750     C     TYPE FORMULA.
04800           INDX=3-LAST
04900           TYPE 710,PUT(2,INDX),PUT(1,INDX)
05000     710   FORMAT(//,' CLASS= ',I4,'      COST= ',I6)
05100           J=3
05200     720   IF(PUT(J,INDX).GE.0)GO TO 740
05300           IF(PUT(J,INDX).EQ.-2)GO TO 790
05400           J=J+1
05500           TYPE 730
05600     730   FORMAT(/)
05700     740   TYPE 750,(PUT(I,INDX),I=J,J+2)
05800     750   FORMAT($,' (X',I3,'=',I5,':',I5,')')
05900           J=J+3
06000           GO TO 720
06100     790   IF(C.EQ.1.0)LAST=3-LAST
06200           RETURN
06300           END


00100           SUBROUTINE PRBEST
00110           IMPLICIT INTEGER (A-Z)
00120           COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
00130          1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
00140          2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
00150          3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
00160          5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
00170          4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
00200           TYPE 100
00300     100   FORMAT(/,' BEST FORMULA SO FAR:')
00400           INDX=LAST
```

```
00500           IF(PUT(I,INDX).NE.-2)GO TO 700
00600           TYPE 110
00700     110   FORMAT(/,' NO FORMULA')
00800           GO TO 790
04900      700  TYPE 710,PUT(2,INDX),PUT(1,INDX)
05000     710   FORMAT(' CLASS= ',I4,'      COST= ',I6)
05100           J=3
05200     720   IF(PUT(J,INDX).GE.0)GO TO 740
05300           IF(PUT(J,INDX).EQ.-2)GO TO 790
05400           J=J+1
05500           TYPE 730
05600     730   FORMAT(/)
05700     740   TYPE 750,(PUT(I,INDX),I=J,J+2)
05800     750   FORMAT($,' (X',I3,'=',I5,':',I5,')')
05900           J=J+3
06000           GO TO 720
06200     790   RETURN
06300           END
```

```
00100           SUBROUTINE CREATE
00200           TYPE 100
00300     100   FORMAT(/,' SORRY, CREATE DOESN''T EXIST YET.')
00400           RETURN
00500           END
```

```
00100           SUBROUTINE GETINT(NUM)
00120           IMPLICIT INTEGER (A-Z)
00140           COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
00160          1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
00180          2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
00200          3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
00220          5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
00240          4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
00400           ACCEPT 510,N
00500     510   FORMAT(15A1)
00600           DO 530 I=1,15
00700           IF(N(I).EQ.' ')GO TO 530
00800           IF(N(I).EQ.'+')GO TO 530
00900           IF(NUM.GT.0.AND.N(I).EQ.'-')GO TO 590
01100           IF(N(I).GT.'9')GO TO 590
01200           IF(N(I).LT.'0')GO TO 590
01500     530   CONTINUE
01600           ENCODE(15,510,M)(N(I),I=1,15)
01700           RETURN
01800     590   M(1)='E      '
01900           RETURN
02000           END
```

```
05800             SUBROUTINE AQ
05900           IMPLICIT INTEGER (A-Z)
05940           COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
05980          1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
06020          2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
06060          3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
06100          5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
06140          4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
06400             INTEGER DIM,QLQ,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
06500             1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
06600             INTEGER CF,E2,E1,E2P,DELTA,CONT,TAU
06700             REAL RAN
06750             DISJNT=1
06800             SIZE=200
```

```
06900     61        DO 60 I=1,SIZE
07000     60        PTR(I)=I+1
07100               PHASE=1
07200               FREE=1
07300               PTR(SIZE)=0
07400               CALL ALLOC(GLIST)
07500               CALL ALLOC(TLIST)
07600               DO 48 I=1,USE1
07700               FP(I)=F1(I)
07800     48        FQ(I)=F1(I)
07900     49        DELTA=0
08000               DO 50 I=1,USE1
08100               IF(400*PHASE+FP(I).EQ.400)GO TO 50
08200               IF(400*PHASE+FQ(I).EQ.800)GO TO 50
08300               DELTA=DELTA+1
08400               E2P=1
08500               OLIST=0
08600               E1=F1(I)
08700     33        J=RAN(J)*NCT+1
08800     C         TYPE 1141,J
08900     1141      FORMAT(' NEXT RANDOM EVENT IS ',I4)
09000               READ(1#J)IC,EVE
09100               IF(IC.EQ.C1)GO TO 33
09200     18        IDIM=1
09300     C         TYPE 114,E2P
09400               CALL ALLOC(TS)
09500     19        DIM(TS)=IDIM
09600               XLB(TS)=0.000
09700               UB(TS)=10000
09800               IF(EVE(IDIM)*EVENT(E1,IDIM).EQ.0)GO TO 23
09900               IF(EVENT(E1,IDIM)-EVE(IDIM))20,21,22
10000     21        IDIM=IDIM+1
10100               IF(IDIM-NDIM)19,19,24
10200     20        UB(TS)=EVE(IDIM)-1
10300               GO TO 23
10400     22        XLB(TS)=EVE(IDIM)+1
10500     23        CALL COPY(TS,T)
10600               Q=OLIST
10700               IF(Q.EQ.0)GO TO 16
10800     11        IF(IABS(DIM(T)-DIM(Q))+IABS(XLB(T)-XLB(Q))+IABS(UB(T)-UB(Q))+SPT
10900               1Q).EQ.0)GO TO 16
11000               Q=PTR(Q)
11100               IF(Q.NE.0)GO TO 11
11200               Q=OLIST
11300     10        R=Q
11400               T=0

11500               CALL COPY(TS,TQ)
11600               FLG=0
11700     12        CALL COPY(R,S)
11800               IF(DIM(TS).EQ.DIM(R))GO TO 13
11900     14        SPTR(S)=T
12000               T=S
12100               R=SPTR(R)
12200               IF(R.NE.0)GO TO 12
12300               IF(FLG.EQ.0)SPTR(TQ)=T
12400               IF(FLG.EQ.0)T=TQ
12500               COST(T)=CF(T)
12600               CALL INSERT(T,GLIST)
12700     15        Q=PTR(Q)
12800               IF(Q)10,21,10
12900     13        XLB(S)=XMAX(XLB(TS),XLB(R))
13000               FLG=1
13100               UB(S)=XMIN(UB(TS),UB(R))
13200               IF(XLB(S)-UB(S))14,14,15
13300     16        COST(T)=CF(T)
13400               CALL INSERT(T,GLIST)
```

```
13500                GO TO 21
13600     24         QLIST=PTR(QLIST)
13700     25           E2P=E2P+1
13800     114        FORMAT(' NEXT E2 IS',I4)
13900                PTR(QLIST)=0
14000                IF(QLIST.EQ.0)GOTO 56
14100                IF(E2P.GT.NC2)GO TO 47
14200     34         J=RAN(J)*NCT+1
14300                READ(1#J)IC,EVE
14400                IF(IC.EQ.C1)GO TO 34
14500                IF(CONT(0,QLIST,0))25,25,18
14600     47         CMAX=COST(QLIST)
14700                P=QLIST
14800     65         IF(COST(P)+NDIM.GT.CMAX)GO TO 63
14900                PTR(Q)=0
15000                GO TO 64
15100     63           Q=P
15200                P=PTR(P)
15300                IF(P.NE.0)GO TO 65
15400     64         JJ=QLIST
15500                IF(PTR(QLIST).NE.0)JJ=QLQ(QLIST)
15600     C          CALL PRT(JJ)
15700                DO 51 J=1,USE1
15800                IF(FQ(J).EQ.0) GO TO 51
15900                IF(CONT(FQ(J),JJ,1).NE.1) GO TO 52
16000     55         FQ(J)=0
16100                FP(J)=0
16200                GO TO 51
16300     52         IF(FP(J).EQ.0) GO TO 51
16400                IF(CONT(FP(J),QLIST,0).EQ.1)FP(J)=0
16500     51         CONTINUE
16600                PTR(JJ)=0
16700                CALL INSERT(JJ,TLIST)
16800     C          TYPE 116
16900     C          ACCEPT 117 ,A
17000     C          IF(A.NE.'Y')RETURN
17100     116        FORMAT(' CONTINUE?')
17200     117        FORMAT(A1)
17300     50         CONTINUE
17400                PHASE=PHASE+1

17500                IF(PHASE.EQ.2) GO TO 49
17600     C          TYPE 102,DELTA
17700     102        FORMAT(' DELTA=',I3)
17800                RETURN
17900          56    DISJNT=0
18000                RETURN
18300                END
18400                SUBROUTINE PRT(R)
18500              IMPLICIT INTEGER (A-Z)
18550              COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
18600             1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
18650             2SIZE,HSLASH(30),COST(200),T,TS,GLIST,QLIST,TLIST,EVE(30),
18700             3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
18750             5LAST,PREVC1,NFORC,OUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
18800             4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
19000                INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
19100                1,P,Q,R,S,T,TS,TLIST,GLIST,QLIST,CST,COST,SEC
19200                INTEGER A(200)
19300                DATA X/'X'/
19400                K=0
19500                P=R
19600                IF(P.EQ.0) RETURN
19700     4          Q=P
19800                TYPE 101, COST(P)
19900                I=1
20000     102        FORMAT(I3,2I7)
```

```
20100    5          A(I)=X
20200               A(I+1)=DIM(Q)
20300               A(I+2)=XLB(Q)
20400               A(I+3)=UB(Q)
20500               I=I+4
20600               WRITE(21,102)DIM(Q),XLB(Q),UB(Q)
20700               Q=SPTR(Q)
20800               IF(Q.NE.0)GO TO 5
20900               I=I-1
21000               TYPE 100,(A(J),J=1,I)
21100               P=PTR(P)
21200    0          IF(P.NE.0) GO TO 4
21300    101        FORMAT('  WEIGHT OF THIS TERM ',I10)
21400    100        FORMAT(3(' ',A1,I3,'=',I7,':',I7,' '))
21500               WRITE(21,102)K
21600               RETURN
21700               END
21800               INTEGER FUNCTION CF(R)
21900             IMPLICIT INTEGER (A-Z)
21950             COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
22000            1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
22050            2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
22100            3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
22150            5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
22200            4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
22400               INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
22500               1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC,E1
22600               CF=0
22700               SEC(R)=0
22800               DO 1 I=1,USE1
22900               E1=F1(I)
23000               IF(E1.EQ.0) GO TO 1
23100               IF(FQ(I).EQ.0.AND.NOSEC.EQ.1)GO TO 1
23200               P=R

23300    2          KK=DIM(P)
23400               IF(EVENT(E1,KK).EQ.0)GO TO 14
23500               IF(XLB(P).GT.EVENT(E1,KK)) GOTO 1
23600               IF(UB(P).LT.EVENT(E1,KK)) GO TO 1
23700    14         P=SPTR(P)
23800               IF(P.NE.0)GO TO 2
23900               SEC(R)=SEC(R)+NDIM
24000               IF(FQ(I).NE.0)CF=CF+NDIM
24100    1          CONTINUE
24200               P=R
24300    4          CF=CF-1
24400               P=SPTR(P)
24500               IF(P.NE.0) GO TO 4
24600               RETURN
24700               END
24800               SUBROUTINE COPY(S1,D)
24900             IMPLICIT INTEGER (A-Z)
24950             COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
25000            1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
25050            2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
25100            3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
25150            5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
25200            4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
25400               INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
25500               1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
25600               INTEGER D,S1
25700               CALL ALLOC(D)
25800               DIM(D)=DIM(S1)
25900               XLB(D)=XLB(S1)
26000               UB(D)=UB(S1)
26100               RETURN
26200               END
```

```
      SUBROUTINE ALLOC(F)
     IMPLICIT INTEGER (A-Z)
     COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
    1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
    2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
    3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
    5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
    4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
        INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
        1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
        INTEGER F,A(6)
        F=FREE
        FREE=PTR(FREE)
        SPTR(F)=0
        PTR(F)=0
        IF(PTR(FREE).NE.0)RETURN
        REF=REF+1
        DO 1 I=1,SIZE
1       MARK(I)=0
        MARK(F)=1
        A(1)=OLIST
        A(2)=TLIST
        A(3)=GLIST
        A(4)=TS
        A(5)=T
        A(6)=TQ
        DO 10 I=1,6
        P=A(I)

        IF(P.EQ.0)GO TO 10
3       Q=P
2       IF  (MARK(Q).EQ.1) GO TO 4
        MARK(Q)=1
        Q=SPTR(Q)
        IF(Q.NE.0) GO TO 2
4       P=PTR(P)
        IF(P.NE.0)GO TO 3
10      CONTINUE
        J=0
        DO 11 I=1,SIZE
        IF(MARK(I).EQ.1) GO TO 11
        PTR(I)=FREE
        FREE=I
        J=J+1
11      CONTINUE
        IF(J.LT..1*SIZE)TYPE 100
        RETURN
100     FORMAT('  USING 90% OF SPACE')
        END
        INTEGER FUNCTION COVER(S,R)
     IMPLICIT INTEGER (A-Z)
     COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
    1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
    2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
    3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
    5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
    4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,IB,NC1,USE1,RAND,DUMMY
        INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
        1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
        COVER=0
        P=S
3       Q=R
1       IF(DIM(P).EQ.DIM(Q)) GO TO 2
        Q=SPTR(Q)
        IF(Q.NE.0) GO TO 1
        RETURN
2       IF(XLB(P).GT.XLB(Q))RETURN
```

```
32500               IF(UB(P).LT.UB(Q))RETURN
32600               P=SPTR(P)
32700               IF(P.NE.0)GO TO 3
32800               COVER=1
32900               RETURN
33000               END
33100               INTEGER FUNCTION CONT(E,LIST,FLAG)
33200             IMPLICIT INTEGER (A-Z)
33250             COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
33300            1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
33350            2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
33400            3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
33450            5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
33500            4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,I8,NC1,USE1,RAND,DUMMY
33700               INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
33800               1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
33900               INTEGER E,FLAG
34000               CONT=0
34100               P=LIST
34200     3         Q=P
34300     2         IF(E.NE.0)XT=EVENT(E,DIM(Q))
34400               IF(E.EQ.0)XT=EVE(DIM(Q))

34500               IF(XT.EQ.0)GO TO 12
34600               IF(XT.LT.XLB(Q)) GO TO 1
34700               IF(XT.GT.UB(Q)) GO TO 1
34800    12         Q=SPTR(Q)
34900               IF(Q.NE.0) GO TO 2
35000               CONT=1
35100               RETURN
35200    1          IF(FLAG.EQ.1)RETURN
35300               P=PTR(P)
35400               IF(P.NE.0)GO TO 3
35500               RETURN
35600               END
35700               SUBROUTINE INSERT(Q,LIST)
35800             IMPLICIT INTEGER (A-Z)
35850             COMMON DIM(200),XLB(200),UB(200),SPTR(200),PTR(200),MARK(200),
35900            1TAU,FREE,TQ,SEC(200),F1(100),F2(100),FQ(100),EVENT(100,30),
35950            2SIZE,HSLASH(30),COST(200),T,TS,GLIST,OLIST,TLIST,EVE(30),
36000            3IFL,DISJNT,U(200),NSEL(200),N(15),M(3),FORC(200),PREVC,
36050            5LAST,PREVC1,NFORC,PUT(100,2),NEXT1,NEXT20,FP(100),MAXSTR,
36100            4DIMTOT,NCT,NDIM,MAX1,NOSEC,C1,NC2,I8,NC1,USE1,RAND,DUMMY
36300               INTEGER DIM,SPTR,PTR,MARK,FREE,F1,F2,FQ,FP,SIZE,HSLASH
36400               1,P,Q,R,S,T,TS,TLIST,GLIST,OLIST,CST,COST,SEC
36500               INTEGER BP,COVER
36600               IF(XLB(Q).EQ.0.AND.UB(Q).EQ.10000)RETURN
36700               K=0
36800               IF (Q.EQ.0)RETURN
36900               J=LIST
37000               P=J
37100    1          BP=P
37200               P=PTR(P)
37300               IF(P.EQ.0)GO TO 5
37400               IF(COST(Q)-COST(P))2,3,4
37500    2          J=BP
37600               IF(COVER(P,Q))1,1,6
37700    3          IF(COVER(P,Q).EQ.1)GO TO 6
37800               IF(COVER(Q,P).NE.1)GO TO 1
37900    7          PTR(BP)=PTR(P)
38000               P=BP
38100               GO TO 1
38200    4          IF(COVER(Q,P))1,1,7
38300    5          P=LIST
38400    9          BP=P
38500               P=PTR(P)
38600               IF(P.EQ.0)GO TO 10
```

```
38700                IF(COST(Q).LT.COST(P))GO TO 9
38800                IF(SEC(Q).LE.SEC(P))GO TO 9
38900    10          PTR(BP)=Q
39000                PTR(Q)=P
39100                P=LIST
39200                K=0
39300    11          P=PTR(P)
39400                IF(P.EQ.0)GO TO 6
39500                K=K+1
39600                IF(K.LE.MAXSTR)GO TO 11
39700                PTR(P)=0
39800    6           CONTINUE
39900                RETURN
40000    8           PTR(P)=0
40100                GO TO 5
40200                END

40300                INTEGER FUNCTION XMIN(XI,XJ)
40400                IMPLICIT INTEGER(A-Z)
40500                XMIN=XI
40600                IF (XJ.LT.XI)XMIN=XJ
40700                 RETURN
40800                END
40900                INTEGER FUNCTION XMAX(XI,XJ)
41000                IMPLICIT INTEGER(A-Z)
41100                XMAX=XI
41200                 IF (XJ.GT.XI)XMAX=XJ
41300                RETURN
41400                END
41500                INTEGER FUNCTION QLQ(I)
41600                QLQ=I
41700                RETURN
41800                END
```

******* AOPI *** 101 ***** RIN 23 ***** [A113,6574] FORSE