NASA TECHNICAL NOTE

NASA TN D-6545

c. 1

NASA TN D-6545

NAMER - A FORTRAN IV PROGRAM FOR USE IN OPTIMIZING DESIGNS OF TWO-LEVEL FACTORIAL EXPERIMENTS GIVEN PARTIAL PRIOR INFORMATION

by Steven M. Sidik

Lewis Research Center
Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MARCH 1972

1. Report No. NASA TN D-6545	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle NAMER - A FORTRAN IV PROGRAM FOR USE IN OPTIMIZING DESIGNS OF TWO-LEVEL FACTORIAL EXPERIMENTS GIVEN PARTIAL PRIOR INFORMATION		5. Report Date March 1972
		6. Performing Organization Code
7. Author(s) Steven M. Sidik		8. Performing Organization Report No. E-6329
		10. Work Unit No. 132-80
9. Performing Organization Name and Address Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Note
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

Under certain specified conditions, the Bayes procedure for designing two-level fractional factorial experiments is that which maximizes the expected utility over all possible choices of parameter-estimator matchings, physical-design variable matchings, defining parameter groups, and sequences of telescoping groups. NAMER computes the utility of all possible matchings of physical variables to design variables and parameters to estimators for a specified choice of defining parameter group or groups. The matching yielding the maximum expected utility is indicated, and detailed information is provided about the optimal matchings and utilities. Complete documentation is given; and an example illustrates input, output, and usage.

17. Key Words (Suggested by Author(s)) Experimental design; Factorial design; Fractional factorial designs; Optimal design of experiments; Bayesian design of experiments		18. Distribution Statement Unclassified - unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 68	22. Price* $3.00

* For sale by the National Technical Information Service, Springfield, Virginia 22151

CONTENTS

NAMER - A FORTRAN IV PROGRAM FOR USE IN OPTIMIZING DESIGNS OF TWO-LEVEL FACTORIAL EXPERIMENTS GIVEN PARTIAL PRIOR INFORMATION

by Steven M. Sidik

Lewis Research Center

SUMMARY

NAMER can be used to find the Bayes procedure for designing two-level fractional factorial experiments given partial prior information. The required prior information is

(1) A statement for each parameter giving a prior probability that it is not zero

(2) A statement of the probability of stopping at each contemplated stopping point

(3) A statement of the value to the experimenter of an unbiased estimate for each parameter

The steps of the design and performance of the experiment may be represented as a finite discrete game between the experimenter and nature. The decision space E for the experimenter consists of the choice of initial defining parameter group, the choice of the sequence or sequences of subgroups that define the telescoping, the choice of physical-design variable matching, and the choice of parameter-estimator matching. The decision space N for nature consists of the choice of which of the parameters are zero and the choice of the stopping point of the experiment. The Bayes procedure maximizes the expected utility over all possible distinct choices of parameter-estimator matchings, physical-design variable matchings, and defining parameter groups for an assumed strategy for nature.

This report presents an algorithm and a computer program entitled NAMER which computes the expected utility of all possible physical-design variable matchings and parameter-estimator matchings for a specified choice of defining parameter groups. The matchings which maximize the expected utilities are saved and printed out. The computational procedure utilizes the group properties of the parameters and the standard ordering. Complete program documentation is presented including sample input and output and a sample problem illustrating the usage (appendix A), program listings (appendix B), a program symbol table (appendix C), and a general flow diagram of the computer program (appendix D).

INTRODUCTION

The two-level fractional factorial designs represent a class of designs of experiments which yield estimates of first-degree effects and interactions for a small amount of experimentation. The main disadvantage of this class of designs is that the estimates (using linear least-squares estimators) are always estimates of aliased combinations of parameters. To make conclusions about single parameters it is necessary to have some information about the parameters from a source other than the experiment. If such information is available before the experiment is performed, it may be incorporated into the design of the experiment.

There are many situations in practice in which an experimenter may have varying amounts of information concerning the variables he wishes to investigate. Sidik and Holms (ref. 1) have developed some optimal design procedures when the prior information is

(1) A statement for each parameter giving a prior probability that it is not zero

(2) A statement of the probability of stopping at each contemplated stopping point

(3) A statement of the value to the experimenter of an unbiased estimate for each parameter

The steps of the design and performance of the experiment may be represented as a finite discrete game between the experimenter and nature. The decision space E for the experimenter consists of the choice of initial defining parameter group, the choice of the sequence or sequences of subgroups that define the telescoping, the choice of physical-design variable matching, and the choice of parameter-estimator matching. The decision space N for nature consists of the choice of which parameters are zero and the choice of the stopping point of the experiment.

The Bayes procedure maximizes the expected utility over all possible distinct choices of parameter-estimator matchings, physical-design variable matchings, and defining parameter groups for an assumed strategy for nature.

This report presents an algorithm and a computer program entitled NAMER which computes the expected utility of all possible physical-design variable matchings and parameter-estimator matchings for a specified choice of defining parameter group or groups. The computational procedure utilizes the group properties of the parameters and the standard ordering of the parameters.

The program can handle experiment designs for as many as nine factors and 32 stopping points. The relation among the stopping points is arbitrary so that, by proper input of data, multiply telescoping designs may be considered or as many as 32 single-stage designs may be analyzed simultaneously. The program output gives the physical-design variable matchings and the parameter-estimator matchings which are the Bayes decisions. Also those matchings which maximize the expected utility at each individual stopping point are printed out so that a security strategy may be specified.

If an experimental program has already begun so that a physical-design variable matching is specified, NAMER may still be used to change the choices of telescoping options based upon revised prior probabilities of stopping at each stopping point not yet reached.

The algorithm and program are fully described. Listings, sample input and output, and a sample problem illustrating the usage are given.

SYMBOLS

B	full parameter group
$B(h)$	subgroup of B used at the h^{th} stopping point of experiment
h	denotes stopping point of experiment
$i \otimes B(h)$	set of standard-order subscripts of elements of $\beta_i \otimes B(h)$
n	number of factors (independent variables)
$P[A]$	permutation of ordering A
$Pr(A)$	probability of event A
p_b	prior probability of a block effect not being zero
p_i	probability that β_i is not equal to zero, $Pr(\beta_i \neq 0)$
p_{sh}	probability experiment will stop exactly at h^{th} stopping point
U	maximized expected utility over stopping points for a given defining parameter group and matching of variables
$U(h)$	maximized expected utility of h^{th} stage for a given defining parameter group and matching of variables
$U(i,k)$	expected utility gained by assigning estimator for alias set $\beta_i \otimes B(h)$ to β_k, $k \epsilon i \otimes B(h)$
$u_i(h)$	utility assigned to an unbiased estimate of β_i at h^{th} stopping point
$X_A, X_B, \cdots$	independent variables (design)
$X_1, X_2, \cdots$	independent variables (physical)
$\otimes$	group operation
Y	dependent (response) variable
$\beta_I, \beta_A, \beta_B, \cdots$	parameters in a model equation relating design variables to dependent variable

$\beta_i \otimes B(h)$ coset (alias set) obtained by multiplying all elements of $B(h)$ by β_i

$\beta_0, \beta_1, \cdots$ parameters in a model equation relating physical variables to dependent variable

δ random variable with mean zero and finite variance

ϵ element of

REVIEW OF BAYES PROCEDURE AND STATEMENT OF COMPUTING PROBLEM

In a full factorial experiment with n independent variables $X_A, X_B, \cdots$, each restricted to assuming only two values, there are 2^n possible distinct combinations of values. It is common practice to say the independent variables can assume either a "high" level or a "low" level. Each of the 2^n distinct combinations of levels is called a treatment combination. From such an experiment it is possible to estimate the β's in an equation of the form

$$Y = \beta_I + \beta_A X_A + \beta_B X_B + \beta_{BA} X_B X_A + \beta_C X_C + \beta_{CA} X_C X_A + \beta_{CB} X_C X_B + \beta_{CBA} X_C X_B X_A + \cdots + \beta_{\ldots CBA} \cdots X_C X_B X_A + \delta \qquad (1)$$

where δ is a random variable with mean zero and finite variance. (Note that the ordering of the subscripts is the reverse of that normally used. The reason for this will be explained shortly.)

A regular fractional replicate of the full factorial design does not allow separate estimation of all the β's. Certain linear combinations of them can be estimated, however. The particular set of linear combinations which can be estimated depends upon the treatment combinations composing the fractional replicate or, equivalently, upon the choice of the design of the experiment. For example, a one-half replicate experiment on four independent variables would provide eight estimators which might be estimators of (depending upon the particular fraction):

$$\left.\begin{array}{ll} (\beta_I + \beta_{DCBA}) & (\beta_C + \beta_{DBA}) \\ (\beta_A + \beta_{DCB}) & (\beta_{CA} + \beta_{DB}) \\ (\beta_B + \beta_{DCA}) & (\beta_{CB} + \beta_{DA}) \\ (\beta_{BA} + \beta_{DC}) & (\beta_{CBA} + \beta_D) \end{array}\right\} \qquad (2)$$

From such estimators, nothing can be inferred about any single parameter without making some assumptions about the other parameter in the alias set.

The set of all 2^n contrasts which provide estimators of the parameters in a full factorial form a group under the appropriate operation. There is a one-to-one mapping from the group of contrasts onto the group B of parameters. Since the point of view of this report is based upon knowledge about parameters, it is more convenient for the development to be in terms of the parameter group. The operation defining a group with respect to the parameters is analogous to that used in the group of contrasts.

With every regular fractional replicate there is associated a defining parameter group (d. p. g.) which can be used to determine the aliased sets of parameters that can be estimated. Conversely, given a d. p. g., there is a regular fractional replicate associated with it.

Holms (ref. 2) and Holms and Sidik (ref. 3) present a technique called telescoping sequences of blocks. This allows an experimenter to perform a factorial experiment in stages, where the starting stage is a small fractional replicate and the final stage is some larger fraction. Each succeeding stage adds treatment combinations to those run in the preceding stages. In order to retain the orthogonality and the orthogonal blocking, each stage must be a power of two times the size of the preceding stage and all the treatments run must form a regular fractional replicate. In what follows, we will restrict ourselves to single telescoping and consider the h^{th} stopping point to be the h^{th} stage. In the case of multiple telescoping, this would not be true, in general, for there could be many stopping points in a stage and the relations between groups are more complex. This is not essential to the discussion, however; and we consider single telescoping only to keep the notation simple.

Denote the d. p. g. at the starting stage as B(1) and the d. p. g. at the h^{th} stage as B(h). If the d. p. g. 's are such that B(h + 1) is a subgroup of B(h), the sequence of regular fractional replicates corresponding to them will form a telescoping sequence of blocks under the rules established in Holms and Sidik (ref. 3). At the h^{th} stage, the treatment combinations run should form the fractional replicate defined by B(h). The fractional replicate at the h + 1 stage can be achieved by adding to the treatment combinations defined by B(h) those treatments in the replicate defined by B(h + 1) but not yet performed.

As the experiment progresses through the stages, the number of alias sets increases, while the number of β's in each alias set decreases. If the final stage is the full factorial, each parameter is separately estimable except that certain of the parameters are confounded with blocks. Whether block effects physically exist is a question the experimenter must answer.

It will be convenient at this point to introduce an alternate notation for equation (1). Let the n independent variables be denoted as $X_1, \cdots, X_n$. Number the 2^n β's of equation (1) from β_0 to β_{2^n-1} and consider the following equation which is similar to equation (1):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_2 X_1 + \beta_4 X_3 + \cdot \cdot \cdot + \beta_{2^n-1} X_n X_{n-1} \cdot \cdot \cdot X_1 + \delta \tag{3}$$

Equation (1) and equation (3) are both written in what is called the standard order. If the subscripts of the β's are rewritten as n-digit binary numbers, it becomes quite obvious how the terms and coefficients of equation (3) are related. For example, let $n = 4$ and consider the following equation, where the subscripts on the β's are written as binary numbers:

$$\begin{aligned} Y = \beta_0 + \beta_1 X_1 + \beta_{10} X_2 + \beta_{11} X_2 X_1 + \beta_{100} X_3 + \beta_{101} X_3 X_1 + \beta_{110} X_3 X_2 \\ + \beta_{111} X_3 X_2 X_1 + \beta_{1000} X_4 + \beta_{1001} X_4 X_1 + \beta_{1010} X_4 X_2 + \beta_{1011} X_4 X_2 X_1 \\ + \beta_{1100} X_4 X_3 + \beta_{1101} X_4 X_3 X_1 + \beta_{1110} X_4 X_3 X_2 + \beta_{1111} X_4 X_3 X_2 X_1 \end{aligned} \tag{4}$$

In general, a β whose subscript in binary notation has ones in the $i_1, i_2, \cdot \cdot \cdot, i_k$ locations from the right is the coefficient of the $X_{i_1} X_{i_2} \cdot \cdot \cdot X_{i_k}$ interaction.

The set of all 2^n coefficients or parameters form a group B under the appropriate operation. In the alphabetic notation this operation $\otimes$ is simply commutative multiplication of the letter subscripts with the exponents reduced modulo 2. In the binary notation the operation may also be denoted $\otimes$ and defined as

$$\beta_m \otimes \beta_k = \beta_{m_n m_{n-1} \cdot \cdot \cdot m_1} \otimes \beta_{k_n k_{n-1} \cdot \cdot \cdot k_1} = \beta_{d_n d_{n-1} \cdot \cdot \cdot d_1} \tag{5}$$

where $d_i = (k_i + m_i)(\mathrm{mod}\ 2)$. Thus $\beta_{CBA} \otimes \beta_{DCB} = \beta_{DC^2B^2A} = \beta_{DA}$, and $\beta_{0111} \otimes \beta_{1110} = \beta_{1001}$. The defining parameter groups that define the fractional replicates at the various stopping points are subgroups of B. The aliased sets of parameters at each stopping point are the cosets of B(h), which will be denoted $\beta_i \otimes B(h)$.

The principal reason for introducing these notations is that one major problem of finding an optimal design is one of finding an optimal matching of design variables to the physical variables of the particular experiment. The physical variables in an experiment will be denoted as $X_1, X_2, \cdot \cdot \cdot, X_n$. It is assumed that the experimenter decides that these are the only independent variables to be investigated. Each X_i represents one of the physical variables and is fixed for the remainder of the experiment. For example,

$$X_1 = \text{Temperature}$$

$$X_2 = \text{Time}$$

$$\vdots$$

$$X_n = \text{Velocity}$$

The design variables will be denoted as X_A, X_B, X_C, . . . and so forth. These variables represent abstractions, and tables exist which tabulate experimental designs in terms of these design variables. When an experimenter consults one of these tables and chooses a design, he must then determine a matching of the design variables and the physical variables. Ordinarily the choice is arbitrary because the experimenter usually does not have prior information available which would indicate that one matching might be preferred to another. A combination of choices of d. p. g. 's, physical-design variable matching, and parameter-estimator matching completely specifies for the experimenter how to proceed with his experiment and estimation of parameters. Hence, such a combination of choices will be called a DESIGN.

Sidik and Holms (ref. 1) present an analysis of choosing a best DESIGN under the following conditions:

(1) For each β_i of equation (3), the experimenter can specify the probability that β_i is not equal to zero, $p_i = \Pr(\beta_i \neq 0)$.

(2) For each h denoting a possible stopping point of the experiment, the experimenter can specify the probability of stopping exactly at the h^{th} stopping point, p_{sh}.

(3) For each β_i of equation (3) and each h, the experimenter can specify the value to him of obtaining an unbiased estimate of β_i. This is denoted by $u_i(h)$.

None of the β's may be separately estimated from a fractional factorial experiment unless some assumptions about certain of the β's are introduced. Conditions (1) and (3) provide assumptions that will enable the experimenter to assign the estimator for an alias set to a single parameter from the alias set and evaluate the consequences of this.

Changing the matching of physical and design variables will usually change the alias sets. For example, if the matching for n = 4 is

$$X_1 = X_A$$

$$X_2 = X_B$$

$$X_3 = X_C$$

$$X_4 = X_D$$

then the alias set $(\beta_A, \beta_B, \beta_{DBA}, \beta_D)$ is mapped into $(\beta_{0001}, \beta_{0010}, \beta_{1011}, \beta_{1000}) = (\beta_1, \beta_2, \beta_{11}, \beta_8)$. But the matching

$$X_1 = X_B$$

$$X_2 = X_A$$

$$X_3 = X_D$$

$$X_4 = X_C$$

maps $(\beta_A, \beta_B, \beta_{DBA}, \beta_D)$ into $(\beta_{0010}, \beta_{0001}, \beta_{0111}, \beta_{0100}) = (\beta_2, \beta_1, \beta_7, \beta_4)$.

Before considering how best to match physical and design variables, let us assume that some such matching has been made. Then consider the problem of matching estimators and parameters at the h^{th} stopping point. The d.p.g. is B(h) and the alias sets are all those distinct cosets of the form $\beta_i \otimes B(h) = \left\{\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_m}\right\}$. If the parameter $\beta_k \epsilon \beta_i \otimes B(h)$ and the estimator for that alias set is assigned to β_k, then, assuming independence, the prior probability that the estimator will be unbiased is

$$\prod_{\substack{j \epsilon i \otimes B(h) \\ j \neq k}} (1 - p_j)$$

Since $u_i(h)$ is the utility of an unbiased estimate of β_i at the h^{th} stopping point,

$$U(i,k) = u_k(h) \prod_{\substack{j \epsilon i \otimes B(h) \\ j \neq k}} (1 - p_j) \tag{6}$$

is the expected utility of the decision to assign the estimator for the alias set $\beta_i \otimes B(h)$ to the parameter β_k. Thus the Bayes strategy is to assign the estimator to the parameter of the alias set which maximizes this expected utility. One case deserves special mention.

Suppose an estimator is confounded with a block effect. It may be safely assumed that an unbiased estimate of a block effect has no utility to the experimenter. Let the prior probability of the block effect being nonzero be denoted by p_b. Then this information can be incorporated into the decision procedure by computing the expected utility as

$$U(i,k) = u_k(h)\left[\prod_{\substack{j \in i \otimes B(h) \\ j \neq k}} (1 - p_j)\right](1 - p_b) \tag{7}$$

where p_b = Pr (the block effect does not equal 0).

With respect to block effects, it is important to note that, depending upon how the block parameters are defined, the estimator for the d. p. g. may also be confounded with blocks. Let $U(i, k_{max}) = \max[U(i,k){:}k \epsilon i \otimes B(h)]$, where the $U(i, k)$ are computed as in equation (6) or equation (7), as appropriate. Then, for the assumed physical-design variable matching and the given d. p. g., the maximized expected utility at the h^{th} stopping point may be denoted by

$$U(h) = \sum U(i, k_{max}) \tag{8}$$

where the summation is over all the distinct cosets at the h^{th} stopping point. By condition (2) (p. 7) it is also assumed that the experimenter can specify the probabilities of stopping exactly at each of the stopping points. Thus,

$$U = \sum_h p_{sh} U(h) \tag{9}$$

represents the maximized expected utility of the resulting DESIGN.

The Bayes procedure for choosing an optimal DESIGN is to compute the expected utility for each choice of DESIGN and then use any one which yields a maximum expected utility. This can be done by computing U as defined in equation (9) for each choice of physical-design variable matching and all possible distinct (that is, not equivalent under a permutation of letters) choices of d. p. g. 's. NAMER computes U for all possible physical-design variable matchings for a specified set of d. p. g. 's. Repeated application of NAMER to different choices of d. p. g. 's would then allow the experimenter to carry out the full Bayes procedure if he wished. Thus the computing problem is that of mechanizing the evaluation of U for all the matchings of design variables to physical variables.

ALGORITHM

The generation and evaluation of all the matchings of the physical and design variables present two computing problems. The first problem is the generation of all the matchings. This really amounts to computing all permutations of the design variables. The second problem is that of evaluating any given permutation.

The information necessary to evaluate a particular matching is (1) what parameters are in the alias sets, (2) the prior probabilities of the parameters, (3) the utility of unbiased estimates of each parameter, (4) the alias sets confounded with blocks, and (5) the prior probabilities of each block parameter not being zero. The computing procedure used by NAMER uses the group properties of the parameters and binary notation for the subscripts of the parameters. The parameters arranged in the standard order are uniquely identified by the standard-order subscript. Thus, arrays called PROB and UTIL may be set up such that the J^{th} entries are p_{J-1} and u_{J-1}, respectively. Also, arrays called BLOCK and PBLOCK may be set up which indicate the alias sets confounded with blocks and the probabilities associated with them. Then when information about β_J is needed to compute the expected utilities of equation (6) or (7), it can be immediately retrieved.

For the remainder of this discussion consider only a single stopping point since the following procedure will simply be repeated for each stopping point: To determine the alias sets, the d.p.g. must be known. Suppose the d.p.g. is stored in an array called DPG. The numbers in the array DPG are the standard-order subscripts of the parameters in the d.p.g. when the standard order is computed with respect to the design variables. For example, suppose the d.p.g. for the stopping point under consideration is $\left\{\beta_I, \beta_{CBA}, \beta_{DCB}, \beta_{DA}\right\}$, and the matching to be evaluated is

$$X_1 = X_A$$

$$X_2 = X_C$$

$$X_3 = X_B$$

$$X_4 = X_D$$

Then the set of standard-order subscripts would be { 0000, 0111, 1110, 1001 } or { 0, 7, 14, 9 }. Since the d.p.g. must always contain the identity or β_I, this is redundant information to store. Hence, the numbers which should be stored in DPG are 7, 14, and 9.

The operation $\otimes$ defined by equation (5) can be defined by the Exclusive Or (IEXOR) function (defined on p. 24), which is available in almost all computing languages. Thus to identify the alias set corresponding to any specified parameter, say β_J, all that is needed is to compute the Exclusive Or between J and each number in DPG. The result will be the standard-order subscripts of the parameters aliased with β_J.

If we specify that the numbers in BLOCK are the standard-order subscripts (with respect to the design variables) of one parameter from each of the alias sets confounded with blocks and that the respective elements of PBLOCK are the prior probabilities of the block parameters, then the same use of Exclusive Or can be applied. For example, suppose the d.p.g. under consideration is that given previously, $\{\beta_I, \beta_{CBA}, \beta_{DCB}, \beta_{DA}\}$. Also suppose it is known that $\{\beta_{BA}, \beta_C, \beta_{DCA}, \beta_{DB}\}$ and $\{\beta_{DC}, \beta_{DBA}, \beta_B, \beta_{CA}\}$ are each confounded with block parameters with prior probabilities of 0.50. Then one element from each of the two alias sets may be chosen to represent it. Suppose they are β_{BA} and β_{DC}. Then BLOCK(1) should be set to $11)_2 = 3$, and BLOCK(2) should be set to $1100)_2 = 12$; and PBLOCK(1) = PBLOCK(2) = 0.50.

It is now an easy task to compute the expected utilities of equations (6) and (7). It remains to find all the distinct alias sets in some economical manner. To do so, set up an array denoted T1 which is 2^n words long. This will be used as an indicator array to indicate if a parameter has been found in an alias set so far. Begin the computation for the stopping point by setting U(h) = 0.0 and initializing the T1 array to some value, say zero.

For each block effect set the element given by BLOCK in T1 to some indicator value not equal to the initialization value, say IRUN; and compute the Exclusive Or of that element and every value in DPG. This will yield the standard-order subscripts of all the parameters in the alias set. To indicate that these parameters have been identified as members of an alias set, set the locations of T1 corresponding to these parameters equal to IRUN. These standard-order subscripts and the value in PBLOCK indicate where to find the probabilities and utilities necessary for making the optimal estimator-parameter matching according to equation (7). Compute the expected utilities, identify the maximum, and add this value to U(h).

Now begin searching T1 until a value not equal to IRUN is found. Suppose the subscript of this value is K. Then compute the EXOR of K - 1 with each number in DPG. Along with K - 1 itself, this will yield the standard-order subscripts of all parameters in the alias set containing β_{K-1}. To indicate that these parameters have been identified as members of an alias set, set the locations of T1 corresponding to these parameters equal to IRUN.

These standard-order subscripts provide the information needed to find the probabilities and utilities necessary for making the optimal estimator-parameter matching. Compute the expected utilities, identify the maximum, and add this value to U(h).

Now continue searching T1 from the location $K + 1$ for another value not equal to IRUN. This will find the next parameter in the standard order which has not yet appeared in an alias set. Thus, the preceding evaluation procedure should be repeated until the end of the T1 array is reached. At that point, all the distinct alias sets will have been identified and evaluated once and only once. The value of $U(h)$ will then be the total maximized expected utility for the h^{th} stopping point corresponding to the optimal estimator-parameter matchings for the current physical-design variable matching. This same procedure is simply repeated for each stopping point and then $U = \sum p_{sh} U(h)$ may be calculated. The matchings of physical-design variables which provide the largest values of $U(h)$ and U may be kept updated in several arrays. Then when all the permutations are completed, the optimal matchings will be available.

What must now be developed is a procedure for generating all the matchings of physical to design variables in some economical manner. Since all the permutations are to be evaluated, the result does not depend upon which matching is done first or in what order they are generated. Thus the starting permutation and the order of generation may be whatever is most convenient computationally. A simple convention used in NAMER is to begin with the matching

$$
\begin{aligned}
X_1 &= X_A \\
X_2 &= X_B \\
&\vdots
\end{aligned}
$$

The distinction has been made previously that the alphabetic subscripts are for the design variables and the numeric subscripts for the physical variables. Thus the preceding starting convention has both sets of variables in the standard ordering. Suppose the d.p.g. at a given stopping point is $\left\{\beta_I, \beta_{CBA}, \beta_{DCB}, \beta_{DA}\right\}$. Then for the matching

$$
\left.
\begin{aligned}
X_1 &= X_A \\
X_2 &= X_B \\
X_3 &= X_C \\
X_4 &= X_D
\end{aligned}
\right\} \tag{10}
$$

the DPG array contains 0111 for (β_{CBA}), 1110 for (β_{DCB}), and 1001 for (β_{DA}). To evaluate a different matching, say

$$\left.\begin{aligned} X_1 &= X_A \\ X_2 &= X_C \\ X_3 &= X_D \\ X_4 &= X_B \end{aligned}\right\} \tag{11}$$

the DPG array should contain 1011 for (β_{CBA}), 1110 for (β_{DCB}), and 0101 for (β_{DA}). The latter DPG can be derived from the former by permuting the binary digits according to the same permutation that gives the ordering X_B, X_D, X_C, X_A starting with X_D, X_C, X_B, X_A. Recall that the binary digits are numbered from right to left. Thus the different matchings of variables can be achieved by constructing all the n! permutations of the rightmost n binary bits in the numbers in DPG. The same procedure applies to the BLOCK array for the same reasons.

Ord-Smith (ref. 4) has presented a survey of a number of possible permutation algorithms. Of these, the best for the purposes of NAMER is the one by Trotter (ref. 5). Trotters' algorithm computes all the permutations as a sequence of adjacent transpositions. To see why this is best, consider how to achieve the permutation of the binary bits by means of arithmetic and logical machine operations. Let M be the number to be changed and express it in binary as $M = m_n m_{n-1} \cdot \cdot \cdot m_j m_{j-1} \cdot \cdot \cdot m_1$ and suppose the digits m_j and m_{j-1} are to be transposed. Compute

$$I = \text{AND}\,(\overset{n}{0}\quad 0\quad 0\ldots \overset{j}{1}\quad \overset{j-1}{0}\ldots \overset{1}{0},\ M)$$

$$J = \text{AND}\,(\overset{n}{0}\quad 0\quad 0\ldots \overset{j}{0}\quad \overset{j-1}{1}\ldots \overset{1}{0},\ M)$$

$$K = \text{AND}\,(\overset{n}{1}\quad 1\quad 1\ldots \overset{j}{0}\quad \overset{j-1}{0}\ldots 1,\ M)$$

$$I = I/2$$

$$J = J * 2$$

$$M = I + J + K = m_n m_{n-1} \cdot \cdot \cdot m_{j-1} m_j \cdot \cdot \cdot m_1$$

Notice that the shifting of the digits m_j and m_{j-1} is accomplished by the multiplication and division by 2. If the permutations were not the result of transpositions of adjacent digits, a more general shift function would be needed or the use of powers of 2 would be needed. These would take more time to execute and/or more logic to control than the current method. This is an important consideration since the computing of the permutations accounts for a substantial portion of the computing job.

A third major problem involved in the program is that of providing the necessary output from the calculations in an economical and useful manner. To explain what NAMER does it will be convenient to introduce some notations for, and properties of, permutations. Let A denote the set of the first n letters of the alphabet arranged in order; that is, $A = \{A, B, C, D, . . . \}$. Let an <u>ordering</u> of A be the set of the first n letters of the alphabet arranged in some arbitrary order. Let a <u>permutation</u> on A or any particular ordering of A be a function denoted as

$$P = \begin{pmatrix} 1\,2\,3 & . . . & n \\ i_1 i_2 i_3 & . . . & i_n \end{pmatrix} \tag{12}$$

which means to take the j^{th} element of the ordering and make it the $i_j{}^{th}$ element for $j = 1, 2, . . . n$. Thus a permutation is a function which maps the set of all possible orderings of A one-to-one onto itself. Since the upper line of equation (12) is redundant, the notation for P is often reduced to $P = (i_1, i_2, . . ., i_n)$.

A <u>transposition</u> is a permutation which interchanges exactly two elements of A. Any permutation can be expressed as a product of transpositions of the form $(1, i_1)$ $(2, i_2) . . . (n, i_n)$. Here (j, i_j) is an abbreviated notation for

$$\begin{pmatrix} 1 & 2 . . . j . . . i_j . . . n \\ 1 & 2 . . . i_j . . . j . . . n \end{pmatrix}$$

The product of transpositions may be expressed as a <u>transposition</u> <u>vector</u> $\langle i_1, . . ., i_n \rangle$. As an illustration note

$$P = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 5 & 2 & 4 \end{pmatrix}$$

$$= (1,\ 3)\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 5 & 2 & 4 \end{pmatrix}$$

$$= (1,\ 3)(2,\ 3)\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 5 & 3 & 4 \end{pmatrix}$$

$$= (1,\ 3)(2,\ 3)(3,\ 5)\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 5 & 4 \end{pmatrix}$$

$$= (1,\ 3)(2,\ 3)(3,\ 5)(4,\ 5)(5,\ 5) = \langle 3,\ 3,\ 5,\ 5,\ 5\rangle$$

Then $P[ABCDE] = [BDAEC]$ directly and

$$\begin{aligned}(1,\ 3)(2,\ 3)(3,\ 5)(4,\ 5)(5,\ 5)[ABCDE] &= (1,\ 3)(2,\ 3)(3,\ 5)(4,\ 5)[ABCDE] \\ &= (1,\ 3)(2,\ 3)(3,\ 5)[ABCED] \\ &= (1,\ 3)(2,\ 3)[ABDEC] \\ &= (1,\ 3)[ADBEC] \\ &= [BDAEC]\end{aligned}$$

as a sequence of transpositions. This illustrates the equivalence of the two ways of expressing **P**. It is obvious that the permutation expressed as a product of transpositions is most convenient for this computer application. Let NDPG be the number of defining parameter groups. Then the orderings $O_1, \ldots, O_{NDPG+1}$ which yield the largest overall expected utility (O_1) and the largest utilities at each stopping point ($O_2, \ldots, O_{NDPG+1}$) are the information the statistician seeks.

The program included in this report does not print all the information for every permutation since this would be much too large a volume of output. All that is saved are the orderings $O_i (i = 1, 2, \ldots, NDPG + 1)$. The initial letter-factor matching is the assignment of the i^{th} letter of the alphabet to i. After all the permutations are performed and evaluated, the program returns the order of the letters and the d.p.g.'s to their original state.

Then permutations P_i must be found to effect reorderings upon $A = \{A, B, C, D, E, \ldots\}$ to achieve $O_1, O_2, \ldots O_{NDCG+1}$ in some efficient manner. Let

$$P_1[A] = O_1$$

$$P_2[A] = P_2\left[P_1^{-1}\left[P_1[A]\right]\right] = O_2$$

$$P_i[A] = P_i\left[P_{i-1}^{-1}\left[P_{i-1}[A]\right]\right] = O_i$$

and define $O_0 = A$, $P_0 = (1, 2, \ldots, M)$. Then the sequence of permutations $P_i\left[P_{i-1}^{-1}\left[O_{i-1}\right]\right]$ should be determined. If $P = (p_1, \ldots, p_m)$, then $P^{-1} = (r_1, \ldots, r_m)$, where $r_{p_i} = i$. That is, r_j is the subscript of the location in P which contains j. If $Q = (q_1, \ldots, q_m)$ and $P^{-1} = (r_1, \ldots, r_m)$, then $QP^{-1} = (s_1, \ldots, s_m)$, where $s_i = q_{r_i}$. That is, to get s_j, find the character in the j^{th} position of P^{-1} and then go to that location in Q to find s_j. Putting the two operations together - if

$$P = (p_1, \ldots, p_m)$$

$$Q = (q_1, \ldots, q_m)$$

then

$$QP^{-1} = (s_1, \ldots, s_m)$$

where $s_i = q_{r_i}$. That is, find the subscript of the location in P which contains i and go to that location of Q to find s_i.

PROGRAM DESCRIPTION

The NAMER program is composed of the main program NAMER and five subroutines: BLOK, LINER, PERMUT, REMACH, and RECT. They will be discussed in some detail after the following brief descriptions:

Program name	Calling name	Purpose
NAMER	Main program	Input; evaluation of each permutation; identifies and and saves optimal matchings; overall program control.
BLOK		Block data subprogram.
PERM	PERMUT	Determines permutations and permutes DPG and BLOCK arrays.
REMAX	REMACH	Achieves rematchings of physical-design variables; outputs detailed description of matchings of physical-design variables and estimators-parameters by appropriate calls to LINE and ALINE.
LINER	LINE	Prints one line of output identifying an estimable parameter and the utility of the assignment of the estimator to the parameter.
	ALINE	If two or more members of an alias set are tied for maximum utility, ALINE identifies the remainder not printed by LINE.
ERECT	RECT	Prints summary output table of the Bayes matching of physical to design variables, the optimal matchings for each stopping point, and the utilities.

Main Program NAMER

NAMER is the main program and is divided into 16 major sections, as indicated by the comments cards in the listing.

Sections 1 to 10. - Read and write input information.

Section 11. - This section is the heart of the program, where each d.p.g. is evaluated and its contribution to the total expected utility computed for a given permutation of the letters. The section is divided into two subsections, 11A and 11B. Section 11A chooses the parameter-estimator matching for the alias sets which are confounded with blocks. Section 11B does the same for the remainder of the alias sets. The computations for IUTILF = 1, 2 are less complicated than those for IUTILF = 3, 4, 5. Thus these cases are separated in the program. See section 7 of the input description for further explanation of IUTILF.

Section 12. - The expected utility of this ordering (PBAYEX) and the expected utilities at the stopping points (PSUMX(-)) are compared to the best utilities to this point (PBAYES AND PSUM(-)). If any one or more is larger than the best so far, the current appropriate utility is placed in PBAYES or PSUM(-) and the ordering (indicated by the contents of IALPHA(-)) is saved in ISAVEP(-, I). The convention is that ISAVEP(-, 1) saves the ordering which gave the best weighted utility and ISAVEP(-, I + 1) saves the ordering which gave the best utility for the I^{th} d.p.g.

Section 13. - This section permutes the current letter-variable relation for the appropriate class or classes. If all possible distinct permutations have been realized, control passes to section 16. If not all permutations have been realized, control passes to section 14.

Section 14. - If current execution time exceeds that allowed, go to section 15. Otherwise go to section 11 to evaluate the current ordering.

Section 15. - All essential information is punched on cards to permit a restart of this case on another computer run beginning with the current ordering.

Section 16. - Print current clock time and call REMACH.

Subroutine PERM

This subroutine uses a FORTRAN translation of Trotter's routine (ref. 5) to permute the elements in an array as a sequence of transpositions of adjacent elements. The first two blocks are the logic which determines which two adjacent elements are to be transposed. The third block (beginning with line 44) is where the arrays IALPHA, DPG, and BLOCK are actually permuted.

Subroutine REMAX

This subroutine uses the information saved in section 12 of NAMER to recompute the utilities of the optimal matchings.

Section 1. - At this point, the DPG and BLOCK arrays contain the same values they had as initial input data. The permutation required to achieve the first optimal ordering is computed and stored in KPERM(-).

Section 2. - Write the letter-variable matching.

Sections 3 and 4. - Translate permutation vector into transposition vector, and permute BLOCK and DPG arrays.

Section 5. - Performs same function as section 11 of NAMER except that calls to LINE and ALINE are made as appropriate.

Section 6. - Output overall expected utility and expected utilities for each d.p.g.

Section 7. - Compute next permutation and shift to section 2.

To illustrate sections 1, 3, and 7 of REMAX consider the following example:

$$O_1 = [4\ 1\ 3\ 2\ 5] = P_1[A]$$

$$O_2 = [3\ 4\ 1\ 5\ 2] = P_2[A]$$

$$O_3 = [4\ 1\ 3\ 2\ 5] = P_3[A]$$

$$O_4 = [4\ 3\ 2\ 1\ 5] = P_4[A]$$

$$O_5 = [1\ 4\ 5\ 2\ 3] = P_5[A]$$

The sequence of values is as follows:

Sequence	KPERM	KKSAVE	K2CYCL	Ordering
1	P_1 = (24315)	P_1 = (24315)	{ 24345 }	[41325]
2	P_2 = (35124)		(23154) = $P_2P_1^{-1}$	
3	$P_2P_1^{-1}$ = (23154)	P_2 = (35124)	{ 23355 }	[34152]
4	P_3 = (24315)		(31254) = $P_3P_2^{-1}$	
5	$P_3P_2^{-1}$ = (31254)	P_3 = (24315)	{ 33355 }	[41325]
6	P_4 = (43215)		(14235) = $P_4P_3^{-1}$	
7	$P_4P_3^{-1}$ = (14235)	P_4 = (43215)	{ 14445 }	[43215]
8	P_5 = (14523)		(25413) = $P_5P_4^{-1}$	
9	$P_5P_4^{-1}$ = (25413)	P_5 = (14523)	{ 25455 }	[14523]

Subroutine LINER

This is a double-entry subroutine with entry points LINE and ALINE. LINE is called by output once for each alias set. The entries of the calling vector are the standard-order subscript number of the parameter in the alias set to which the estimator should be assigned and the expected utility of that assignment. The standard-order subscript is then used to identify the parameter in terms of the interaction of the independent variables it measures. This identification is then printed in numerical form and in Hollerith form using the first six characters of each factor identification card. The utility is also printed.

ALINE is called whenever there are two or more parameters aliased which give the same maximized expected utility. The call to LINE causes one of the aliased parameters to be identified and the expected utility of the estimator to be printed. The call to ALINE causes the remaining parameters to be identified. They are, however, identified only by the numerical form of their interaction.

INPUT DESCRIPTION

The following is a detailed description of the input data necessary to run a problem. There are nine basic sets of input data. Each is described in detail here. An example of the type of problem to which this program may be applied is given in reference 1 and a similar problem is discussed in appendix A. A sample set of data for this problem is given in table I. A pictorial illustration of the input deck setup is given in figure 1. Multiple cases may be run back-to-back. The last card of the last case should have ENDALL punched in the first six columns.

The nine basic sets of input data are as follows:

(1) IDENTIFICATION (13A6, A2) (IDENT). This is one card; all 80 columns are used for Hollerith identification of problem.

(2) MAXIMUM TIME (F6.0) (TMAXX). This is the maximum machine time in minutes permitted for this case. If this time is exceeded and the case is not fully evaluated, all pertinent information is punched on cards to permit a restart of the program.

(3) TYPE OF RUN (16) (ITYPRN).

(3A) SPECIFIED MATCHING (9A1) (XT3). A "1" for ITYPRN indicates this is a regular first-time run and data sets 4 to 9 will be read. A "2" indicates this is a restarted case and only those cards punched by the previous run need be read. A "3" indicates that only one matching will be evaluated. This matching is specified on the 3A card with the first n letters of the alphabet (excluding I) in the first n columns of the card in the order appropriate to the matching desired.

For example, if to indicate an evaluation of the following matching is desired

$$X_1 = X_B$$

$$X_2 = X_A$$

$$X_3 = X_D$$

$$X_4 = X_C$$

one card would be supplied with BADC in the first four columns.

(4) NUMBER OF FACTORS (I6) (NFAC). Up to nine factors can be considered.

(5) FACTOR IDENTIFICATIONS (13A6, A2) (FAC). One card for each factor. The first six characters of each card are used as output identification, so they should serve as useful abbreviations.

(6) NUMBER OF CLASSES (I6) (NCLASS).

(6A) NUMBER IN EACH CLASS (9I6) (NSUBI). NCLASS is the number of classes of factors. If this is 1, input set 6A is not read. If the number of classes is more than one, card 6A specifies the number of factors in each class. The factors within a class will be permuted among each other, but permutations between classes will not be permitted. The first NSUBI(1) factors will be assumed to belong to the first class, the next NSUBI(2) to the second class, and so forth. Holms and Sidik (ref. 6) present an experiment in which there are two classes of variables which could not be mixed. Most experiments have only one class.

(7) NUMBER OF NONZERO PROBABILITIES, UTILITY FUNCTION, AND CONSTANT (NPIN, IUTILF, UCOEF) (2I6, F10.9). The number of parameters with nonzero prior probabilities is specified in the first six columns. The choice of utility function is indicated in the second six columns. UCOEF is used in defining utility function 5 and is given in the next 10 columns. Each parameter with a nonzero prior probability or utility is identified in terms of the integer subscripts of the independent variables in the interaction with which it is associated. See the input set (8) description for further information. The possible choices of utility function here are

(a) IUTILF = 1

$$u_i = \begin{cases} 1.0 \text{ for unbiased estimators} \\ 0.0 \text{ for biased estimators} \end{cases}$$

(b) IUTILF = 2

$$u_i = \begin{cases} p_i \text{ for unbiased estimators} \\ 0.0 \text{ for biased estimators} \end{cases}$$

(c) IUTILF = 3

$$u_i = \begin{cases} x_i \text{ for unbiased estimators} \\ 0.0 \text{ for biased estimators} \end{cases}$$

where x_i is given with the p_i in input set (8).

(d) IUTILF = 4

$$u_i = \begin{cases} p_i x_i \text{ for unbiased estimators} \\ 0.0 \text{ for biased estimators} \end{cases}$$

where x_i and p_i are given in input set (8).

(e) IUTILF = 5

$$u_i = \begin{cases} \text{UCOEF} \cdot x_i + (1 - \text{UCOEF})p_i \text{ for unbiased estimators} \\ 0.0 \text{ for biased estimators} \end{cases}$$

where x_i and p_i are given in input set (8) and it is assumed $0.0 \leq \text{UCOEF} \leq 1.0$.

It should be noticed that these utility functions do not depend upon the stopping point as implied by condition (3) on page 7. To provide a capability of making the utility function depend upon the stopping point, the user can weight the stopping points by use of the weighting values WT(I) read in input set (9B). Thus the $u_i(h)$ used in the program are computed as $u_i(h) = u_i * \text{WT}(h)$.

(8) PRIOR PROBABILITIES AND UTILITIES (9I1, 2F10.0) (IT1, P, UT). Each parameter with nonzero prior probability is identified in terms of the integer subscripts of the independent variables in the interaction with which it is associated. These subscripts may be supplied in any order anywhere in the first nine columns of the card. The prior probability and the utility follow with 10 columns each, in F10.0 format. The utility need not be specified if IUTILF = 1 or 2 as previously described, for the program then supplies the utility. If IUTILF = 3, 4, or 5, the utility must be specified explicitly. For example, suppose the $X_2X_3X_5$ interaction parameter $\beta_{10110} = \beta_{22}$ is assumed to satisfy $P(\beta_{22} \neq 0) = 0.850$ with utility of 0.95. Then the card input could be bbb5b2b3b.850bbbbbb.95. If the prior probability of a parameter being nonzero is zero, no data need be supplied for that parameter.

(9) NUMBER OF DEFINING GROUPS (I6) (NDPG). For each d.p.g. (as many as 32 permitted) there must be one set of inputs 9A to 9E.

(9A) IDENTIFICATION OF STOPPING POINT (4A6) (IDDCG).

(9B) NUMBER OF GENERATORS, PRIOR PROBABILITY OF STOPPING, WEIGHTING VALUE (I6, F6.6, F6.0) (NGEN, PSTOP, WT). If the d.p.g. corresponds to a $(1/2)^r$ fractional replicate, r independent generators must be supplied. Program limitations restrict NGEN to values less than or equal to seven.

(9C) THE GENERATORS OF THE d.p.g. (9A1). The generators are supplied in terms of the first NFAC letters of the alphabet on the first nine columns of the card. There is one card per generator. For example, if the d.p.g. at a particular stopping point is $\{\beta_I, \beta_{EDCBA}, \beta_{CBA}, \beta_{ED}, \beta_{DA}, \beta_{ECB}, \beta_{DCB}, \beta_{EA}\}$, three generators are sufficient; and one such choice might be β_{CBA}, β_{ED}, and β_{DA}. Three cards which will define the above d.p.g. might be

AbBbC

ED

bbAbbD

The order and position of the letters is unimportant as long as they are on the first nine columns of the card. If the number of generators is zero, no type-9C cards are read.

(9D) NUMBER OF BLOCK PARAMETERS (I6) (NBLOCK).

(9E) IDENTIFICATION OF ALIAS SETS CONFOUNDED WITH BLOCK EFFECTS AND THE PRIOR PROBABILITY ASSOCIATED WITH THE BLOCK EFFECTS (9A1, F5.5) (XT3, PBLOCK). Any single parameter from an alias set which is confounded with a block effect may be input in terms of the first NFAC letters of the alphabet in the first nine columns of one card. This is followed by the prior probability of the block effect on the next five columns. There is one card for each block effect that has a nonzero prior probability. If the alias set is the d.p.g., the first nine columns may be left blank.

OUTPUT DESCRIPTION

The first part of NAMER output is the printout of the input data. This is followed by NDPG + 1 printouts. The first set is for the Bayes DESIGN which optimizes the overall expected utility. The subsequent sets are for the DESIGNS that optimize the expected utilities for the individual stopping points. Each of these sets of output consists of the following:

(1) The optimal matching

(2) Tables of the parameters chosen to be estimated and the expected utilities of these choices

(3) The overall expected utility and the expected utilities at the stopping points

For example, the first two pages of the sample output in appendix A indicate the input data. The next two pages provide the information about the Bayes DESIGN, as the label indicates. The Bayes matching is seen to be

$$X_1(\text{TEMP}) = X_C$$

$$X_2(\text{PRESS}) = X_D$$

$$X_3(\text{TIME}) = X_B$$

$$X_4(\text{VEL}) = X_E$$

$$X_5(\text{ANGLE}) = X_A$$

Then for d.p.g. number one (which is the 1/4 replicate of the full factorial) the choices of parameters to be estimated are indicated. Each parameter is identified in terms of the integer subscripts of the independent variables in the interaction with which it is associated. They are also identified by the Hollerith identifications input in section (5) of the input, and the utility of the choice is printed at the far right of each line. Thus the first line of output for d.p.g. number one indicates that the coefficient of $X_1X_2X_3 = X_CX_DX_B$ has been chosen from its alias set as the parameter to be estimated. This interaction is the TEMP×PRESS×TIME interaction, and the expected utility of this choice is 0.20. This utility value does not include the weighting factor at this point.

Below the detailed output for the three d.p.g.'s are printed the overall expected utility and the utilities for each of the d.p.g.'s for this matching.

Similar output provides the detailed output for the designs which maximize the expected utilities for each of the stopping points. The format and arrangement are the same as that described for the Bayes matching. This is followed on the last page by a summary table providing the various matchings, their expected overall utilities, and expected utilities at the stopping points.

SPECIAL LEWIS RESEARCH CENTER ROUTINES

Some of the following functions and subroutines available in the FORTRAN IV - Version 13 language at the Lewis Research Center may not be available (or not available in FORTRAN) at other computer installations. Thus, their usage is explained, and the user can write functions or subroutines providing the same capabilities in a language compatible with the available computer.

The functions and subroutines available at Lewis are the following:

(1) AND(A, B). A real function of the Real or Integer variables A and B. Like bit positions of A and B are compared. A 1 is placed in those positions of the result where there are 1's in both A and B, and a zero is placed in the result otherwise.

(2) IEXOR(A, B). An integer function of the Real or Integer arguments A and B. Like bit positions of A and B are compared. A 1 is placed in those positions of the result where exactly one of A or B is a 1, and a zero is placed in the result otherwise.

(3) IALS(N, X). An integer function of Integer N and Real or Integer X. The contents of X are shifted to the left N places and zeros put into the vacated rightmost positions.

(4) IARS(N, X). An integer function of Integer N and Real or Integer X. The contents of X are shifted to the right N places and zeros put into the vacated leftmost positions.

(5) BCREAD(X1, X2) and BCDUMP(X1, X2, K). These subprograms provide for input and output in absolute binary. A call to BCREAD(X1, X2) causes cards to be read in binary format at the rate of 22 words per card. The data are stored sequentially in the core, beginning with the address of the variable X1 and ending with the address of the variable X2. A call to BCDUMP(X1, X2, K) causes cards to be punched in binary format at the rate of 22 words per card. The data are taken sequentially from the core, beginning with the address of variable X1 and ending with the address of variable X2. K provides card numbering control and is always set to zero by NAMER.

As an example of the usage of these routines, consider the first call to BCREAD in section 10 of NAMER. DUMP1(1) is equivalenced to NFAC. NFAC is the first variable of nine variables in the labeled common block B1. LD1 is set to 9 at the start of NAMER. Thus, the call BCREAD (DUMP1(1), DUMP1(LD1)) causes the variables NFAC, NCLASS, NN, NDPG, PBAYES, and so forth, to be read from unit 5 in binary format.

(6) TIME1(X). This subroutine enables the programmer to read the storage cell clock. The following illustrates the procedure for using TIME1 to calculate elapsed time.

```
CALL TIME1(X1)
      .
      .
      .
CALL TIME1(X2)
```

Then

$$X2 - X1 = \text{Clock pulses}$$

$$\frac{X2 - X1}{60} = \text{Elapsed time in seconds}$$

$$\frac{X2 - X1}{3600} = \text{Elapsed time in minutes}$$

(7) OR(A, B). A real function of the real or integer arguments A and B. Like bit positions of A and B are compared. A 1 is placed in those positions of the result where either or both of A and B are a 1. A zero is placed in those positions of the result wherever both A and B are zero.

TIMING INFORMATION

Several sample problems using single telescoping were run on NAMER to estimate the amount of time required by the program. For these problems, the first stage was assumed to be the smallest experiment large enough to estimate all main effects, and the last stage was the full factorial. Each problem was run once using utility function 2 and once using utility function 3. The results are summarized in table II and figure 2.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, November 8, 1971,
132-80.

APPENDIX A

SAMPLE PROBLEM AND PROGRAM OUTPUT

Consider a five-factor experiment involving

X_1 = Temperature

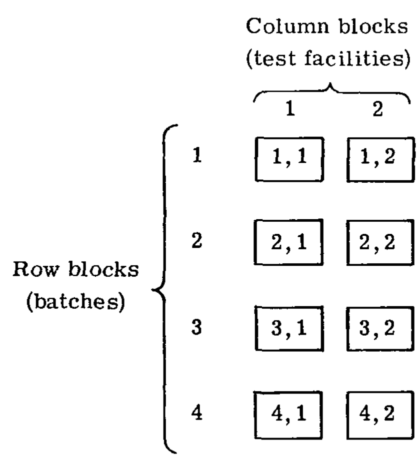
X_2 = Pressure

X_3 = Time

X_4 = Velocity

X_5 = Angle

Suppose that the experimenter's facilities are such that he can only perform four treatment combinations at one time and be reasonably sure that experimental conditions are homogeneous. Thus his experiment should be designed as a blocked factorial design with blocks of size four. Assume also that he has enough materials at one time to perform eight treatment combinations, but no more, and that batches of uniform material are not available in quantities that will supply more than eight treatment combinations. Then the blocks of the experiment might be as shown in the illustration, where the two columns represent two different test facilities and the four rows represent four different batches of raw material.



The difference between the first block and the second block in a row is due to performing the experiment in two different test facilities. The differences between rows are due to possible effects of new batches of materials. Suppose the experimenter feels that there is a probability of 0.50 of there actually being a test facility block effect. Let the probability of there being an effect due to differing batches of raw materials be 1.0. Assume further that the probability of an interaction between these block effects is specified as zero. The stopping points of the experiment at each stage are

(1) Stage one, after completion of blocks (1, 1), (1, 2)

(2) Stage two, after completion of blocks (1, 1), (1, 2), (2, 1), (2, 2)

(3) Stage three, after completion of the full factorial

Based upon his available resources and upon past histories of some similar projects he has worked on, the experimenter feels probabilities in the following table are appropriate:

Coefficient of-	Standard-order subscript	Prior probability of being nonzero
X_0	0	1.00
X_1	1	.80
X_2	2	
X_2X_1	3	
X_3	4	
X_3X_1	5	
X_3X_2	6	
$X_3X_2X_1$	7	
X_4	8	1.0
X_4X_1	9	.50
X_4X_3	12	.50
$X_4X_3X_1$	13	.40
X_5	16	1.0
X_5X_1	17	.40
X_5X_3	20	.30
Stopping probabilities: $p_{1s} = 0.30$, $p_{2s} = 0.40$, $p_{3s} = 0.30$		

All the other coefficients have zero prior probability.

Assume the purpose of the experiment is to maximize the response. Also assume that the cost of the experiment is about proportional to the number of treatment combinations run. Then a reasonable choice for utility function might be

$$u_i(h) = \begin{cases} p_i/n_h & \text{if unbiased} \\ 0 & \text{if biased} \end{cases}$$

where n_h is the number of treatment combinations. To achieve this using NAMER, use utility function 2 and for the weighting values at the stages use $1/n_h$.

Rather than investigating all the possible nonequivalent d.p.g.'s and their telescoping options, the best matchings of physical-design variables and parameters to estimators will be determined for the following choices of d.p.g.'s:

$$B(1) = \left\{\beta_I, \beta_{CBA}, \beta_{EDC}, \beta_{EDBA}\right\}$$

$$B(2) = \left\{\beta_I, \beta_{EDBA}\right\}$$

$$B(3) = \left\{\beta_I\right\}$$

Using the d.p.g. $\left\{\beta_I, \beta_{CBA}, \beta_{DCB}, \beta_{DA}, \beta_{EDC}, \beta_{EDBA}, \beta_{EB}, \beta_{ECA}\right\}$ for block (1, 1) and the rules presented in reference 3, it may be shown that the following assignment of treatment combinations will lead to the block confounding presented in the table:

Block	Treatment
(1, 1)	(1), dca, ecb, edba
(1, 2)	ba, dcb, eca, ed
(2, 1)	db, cba, edc, ea
(2, 2)	da, c, edcba, eb
(3, 1)	a, dc, ecba, edb
(3, 2)	b, dcba, ec, eda
(4, 1)	dba, cb, edca, e
(4, 2)	d, ca, edcb, eba

Stage	Alias sets confounded with -	
	Test facility effect	Raw material effect
1	$\{\beta_{DA}, \beta_{DCB}, \beta_{ECA}, \beta_{EB}\}$	$\{\beta_I, \beta_{CBA}, \beta_{EDC}, \beta_{EDB}\}$
2	$\{\beta_{DA}, \beta_{EB}\}$	$\{\beta_I, \beta_{EDBA}\}$ $\{\beta_{CBA}, \beta_{EDC}\}$
3	$\{\beta_{DA}\}$	$\{\beta_I\}$ $\{\beta_{CBA}\}$ $\{\beta_{EDBA}\}$ $\{\beta_{EDC}\}$

The sample FORTRAN data sheets given in table II supply the data necessary to run this problem as described. The sample output for this problem follows.

```
NAMER OUTPUT  NAMER SAMPLE PROBLEM

PROGRAM WILL DUMP FOR RESTART IF NOT FINISHED IN        2. MINUTES.

 CURRENT EXECUTION TIME          0.00
THERE ARE     5 FACTORS.  THEY ARE...

         1    TEMP     SOURCE TEMPERATURE
         2    PRESS    SOURCE PRESSURE
         3    TIME     TIME DURATION
         4    VEL      SOURCE VELOCITY
         5    ANGLE    ANGLE OF INJECTION

    15 PARAMETERS WITH NON-ZERO PRIOR PROBABILITIES AND UTILITIES

 UTILITY FUNCTION  2
        0   1.000000        1.000000
        1   0.800000        0.800000
        2   0.800000        0.800000
       12   0.800000        0.800000
        3   0.800000        0.800000
       13   0.800000        0.800000
       23   0.800000        0.800000
      123   0.800000        0.800000
        4   1.000000        1.000000
       14   0.500000        0.500000
       34   0.500000        0.500000
      134   0.400000        0.400000
        5   1.000000        1.000000
       15   0.400000        0.400000
       35   0.300000        0.300000

    2 DEFINING PARAMETER GROUPS

-----------------------------------------------------------------------------------------------

1/4 REP--ROW 1

DPG  1
     2 GENERATORS
   PROB OF STOPPING     0.30000       WEIGHT  0.125000
     ABC
     CDE

THERE ARE    2 BLOCK PARAMETERS

       AD        0.50000
                 1.00000
-----------------------------------------------------------------------------------------------
```

```
1/2 REP-- ROWS 1,2

DPG  2
     1 GENERATORS
   PROB OF STOPPING     0.40000        WEIGHT  0.062500
     ABDE

THERE ARE   2 BLOCK PARAMETERS

       AD        0.50000
                 1.00000
      ABC        1.00000

---------------------------------------------------------------------------------------------------------------------------

FULL--ALL ROWS

DPG  3
     0 GENERATORS
   PROB OF STOPPING     0.30000        WEIGHT  0.031250

THERE ARE   5 BLOCK PARAMETERS

       AD        0.50000
                 1.00000
      ABC        1.00000
     ABDE        1.00000
      CDE        1.00000
 CURRENT EXECUTION TIME         0.01
 CURRENT EXECUTION TIME         0.03




 THIS MATCHING IS THE BAYES MATCHING

VARIABLE           SHOULD BE CALLED
     1 TEMP                   C
     2 PRESS                  D
     3 TIME                   B
     4 VEL                    E
     5 ANGLE                  A

DEFINING PARAMETER GROUP NO.    1
1/4 REP--ROW 1
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *1 *2 *3                           TEMP  PRESS TIME                                        0.200000
 0*                                 G MEAN                                                  0
 *1                                 TEMP                                                    0.560000
    *2                                    PRESS                                             0.400000
          *4                                          VEL                                   0.200000
       *3                                       TIME                                        0.480000
             *5                                             ANGLE                           0.200000
    *2 *3                                 PRESS TIME                                        0.480000

DEFINING PARAMETER GROUP NO.    2
1/2 REP-- ROWS 1,2
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       *3 *4                                    TIME  VEL                                   0.250000
 0*                                 G MEAN                                                  0
 *1    *3    *5                     TEMP        TIME        ANGLE                           0
 *1                                 TEMP                                                    0.800000
    *2                                    PRESS                                             0.800000
 *1 *2                              TEMP  PRESS                                             0.800000
       *3                                       TIME                                        0.800000
 *1    *3                           TEMP        TIME                                        0.800000
    *2 *3                                 PRESS TIME                                        0.800000
 *1 *2 *3                           TEMP  PRESS TIME                                        0.800000
          *4                                          VEL                                   1.000000
 *1       *4                        TEMP              VEL                                   0.500000
       *3    *5                                 TIME        ANGLE                           0.300000
 *1    *3 *4                        TEMP        TIME  VEL                                   0.400000
             *5                                             ANGLE                           1.000000
 *1          *5                     TEMP                    ANGLE                           0.400000

DEFINING PARAMETER GROUP NO.    3
FULL--ALL ROWS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    *2       *5                           PRESS             ANGLE                           0
 0*                                 G MEAN                                                  0
 *1    *3    *5                     TEMP        TIME        ANGLE                           0
    *2 *3 *4 *5                           PRESS TIME  VEL   ANGLE                           0
 *1 *2    *4                        TEMP  PRESS       VEL                                   0
 *1                                 TEMP                                                    0.800000
    *2                                    PRESS                                             0.800000
 *1 *2                              TEMP  PRESS                                             0.800000
       *3                                       TIME                                        0.800000
 *1    *3                           TEMP        TIME                                        0.800000
    *2 *3                                 PRESS TIME                                        0.800000
 *1 *2 *3                           TEMP  PRESS TIME                                        0.800000
```

```
          *4                              VEL                          1.000000
 *1       *4                  TEMP        VEL                          0.500000
    *2    *4                       PRESS  VEL                          0
       *3 *4                             TIME  VEL                     0.500000
 *1    *3 *4                  TEMP       TIME  VEL                     0.400000
    *2 *3 *4                       PRESS TIME  VEL                     0
 *1 *2 *3 *4                  TEMP PRESS TIME  VEL                     0
             *5                                      ANGLE             1.000000
 *1          *5               TEMP                   ANGLE             0.400000
 *1 *2       *5               TEMP PRESS             ANGLE             0
       *3    *5                          TIME        ANGLE             0.300000
    *2 *3    *5                    PRESS TIME        ANGLE             0
 *1 *2 *3    *5               TEMP PRESS TIME        ANGLE             0
          *4 *5                                VEL   ANGLE             0
 *1       *4 *5               TEMP             VEL   ANGLE             0
    *2    *4 *5                    PRESS       VEL   ANGLE             0
 *1 *2    *4 *5               TEMP PRESS       VEL   ANGLE             0
       *3 *4 *5                          TIME  VEL   ANGLE             0
 *1    *3 *4 *5               TEMP       TIME  VEL   ANGLE             0
 *1 *2 *3 *4 *5               TEMP PRESS TIME  VEL   ANGLE             0

 FOR THE ABOVE PERMUTATION THE EXPECTED UTILITY IS    0.42169

 THE EXPECTED UTILITIES AT THE STOPPING POINTS ARE..
 DEFINING PARAMETER GROUP    1      0.31500
 DEFINING PARAMETER GROUP    2      0.59062
 DEFINING PARAMETER GROUP    3      0.30312



 THIS MATCHING MAXIMIZES THE EXPECTED VALUE AT THE  1 STOPPING POINT 1/4 REP--ROW 1

VARIABLE           SHOULD BE CALLED
     1 TEMP                D
     2 PRESS               A
     3 TIME                B
     4 VEL                 C
     5 ANGLE               E

DEFINING PARAMETER GROUP NO.   1
1/4 REP--ROW 1
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *1 *2                        TEMP PRESS                               0.168000
 0*                           G MEAN                                   0
 *1                           TEMP                                     0.800000
    *2                             PRESS                               0.400000
       *3                                TIME                          0.800000
 *1    *3                     TEMP       TIME                          0.800000
          *4                                   VEL                     0.120000
             *5                                      ANGLE             0.100000

DEFINING PARAMETER GROUP NO.   2
1/2 REP-- ROWS 1,2
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *1 *2                        TEMP PRESS                               0.280000
 0*                           G MEAN                                   0
    *2 *3 *4                       PRESS TIME  VEL                     0
 *1                           TEMP                                     0.800000
    *2                             PRESS                               0.800000
       *3                                TIME                          0.800000
 *1    *3                     TEMP       TIME                          0.800000
    *2 *3                          PRESS TIME                          0.480000
             *5                                      ANGLE             0.200000
          *4                                   VEL                     1.000000
 *1       *4                  TEMP             VEL                     0.500000
    *2    *4                       PRESS       VEL                     0
                                 *1    *3 *4 *5
 *1 *2    *4                  TEMP PRESS       VEL                     0
                                       *3 *4 *5
       *3 *4                             TIME  VEL                     0.500000
 *1    *3 *4                  TEMP       TIME  VEL                     0.400000
 *1 *2 *3 *4                  TEMP PRESS TIME  VEL                     0
                                          *4 *5

DEFINING PARAMETER GROUP NO.   3
FULL--ALL ROWS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *1 *2                        TEMP PRESS                               0.400000
 0*                           G MEAN                                   0
    *2 *3 *4                       PRESS TIME  VEL                     0
 *1 *2 *3    *5               TEMP PRESS TIME        ANGLE             0
 *1       *4 *5               TEMP             VEL   ANGLE             0
 *1                           TEMP                                     0.800000
    *2                             PRESS                               0.800000
       *3                                TIME                          0.800000
 *1    *3                     TEMP       TIME                          0.800000
    *2 *3                          PRESS TIME                          0.800000
 *1 *2 *3                     TEMP PRESS TIME                          0.800000
          *4                                   VEL                     1.000000
 *1       *4                  TEMP             VEL                     0.500000
```

```
    *2    *4                          PRESS       VEL                                  0
 *1 *2    *4                    TEMP  PRESS       VEL                                  0
       *3 *4                                TIME  VEL                                  C.500000
 *1    *3 *4                    TEMP        TIME  VEL                                  C.400000
 *1 *2 *3 *4                    TEMP  PRESS TIME  VEL                                  C
             *5                                         ANGLE                          1.000000
 *1          *5                 TEMP                    ANGLE                          C.400000
    *2       *5                       PRESS             ANGLE                          0
 *1 *2       *5                 TEMP  PRESS             ANGLE                          0
       *3    *5                             TIME        ANGLE                          C.300000
 *1    *3    *5                 TEMP        TIME        ANGLE                          C
    *2 *3    *5                       PRESS TIME        ANGLE                          C
          *4 *5                                   VEL   ANGLE                          0
    *2    *4 *5                       PRESS       VEL   ANGLE                          C
 *1 *2    *4 *5                 TEMP  PRESS       VEL   ANGLE                          0
       *3 *4 *5                             TIME  VEL   ANGLE                          0
 *1    *3 *4 *5                 TEMP        TIME  VEL   ANGLE                          C
    *2 *3 *4 *5                       PRESS TIME  VEL   ANGLE                          0
 *1 *2 *3 *4 *5                 TEMP  PRESS TIME  VEL   ANGLE                          C

 FOR THE ABOVE PERMUTATION THE EXPECTED UTILITY IS    0.37074

 THE EXPECTEC UTILITIES AT THE STOPPING POINTS ARE..
 DEFINING PARAMETER GROUP    1       0.39850
 DEFINING PARAMETER GROUP    2       0.41000
 DEFINING PARAMETER GROUP    3       0.29062




 THIS MATCHING MAXIMIZES THE EXPECTED VALUE AT THE  2 STOPPING POINT 1/2 REP-- ROWS 1,2

VARIABLE          SHOULD BE CALLED
     1 TEMP                C
     2 PRESS               B
     3 TIME                D
     4 VEL                 A
     5 ANGLE               E

DEFINING PARAMETER GROUP NO.   1
1/4 REP--ROW 1
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *1 *2 *3                       TEMP  PRESS TIME                                       0.200000
 0*                             G MEAN                                                 C
 *1                             TEMP                                                   C.560000
    *2                                PRESS                                            C.400000
          *4                                      VEL                                  C.200000
       *3                                   TIME                                       C.480000
             *5                                         ANGLE                          C.200000
    *2 *3                             PRESS TIME                                       C.480000

DEFINING PARAMETER GROUP NO.   2
1/2 REP-- ROWS 1,2
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       *3 *4                                TIME  VEL                                  C.250000
 0*                             G MEAN                                                 C
 *1 *2    *4                    TEMP  PRESS       VEL                                  0
 *1                             TEMP                                                   C.800000
    *2                                PRESS                                            C.800000
 *1 *2                          TEMP  PRESS                                            C.800000
       *3                                   TIME                                       C.800000
 *1    *3                       TEMP        TIME                                       C.800000
    *2 *3                             PRESS TIME                                       C.800000
 *1 *2 *3                       TEMP  PRESS TIME                                       C.800000
          *4                                      VEL                                  1.000000
 *1       *4                    TEMP              VEL                                  0.500000
       *3    *5                             TIME        ANGLE                          C.300000
 *1    *3 *4                    TEMP        TIME  VEL                                  C.400000
             *5                                         ANGLE                          1.000000
 *1          *5                 TEMP                    ANGLE                          C.400000

DEFINING PARAMETER GROUP NO.   3
FULL--ALL ROWS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       *3 *4                                TIME  VEL                                  C.250000
 0*                             G MEAN                                                 0
 *1 *2    *4                    TEMP  PRESS       VEL                                  C
    *2 *3 *4 *5                       PRESS TIME  VEL   ANGLE                          C
 *1    *3    *5                 TEMP        TIME        ANGLE                          C
 *1                             TEMP                                                   C.800000
    *2                                PRESS                                            C.800000
 *1 *2                          TEMP  PRESS                                            C.800000
       *3                                   TIME                                       C.800000
 *1    *3                       TEMP        TIME                                       C.800000
    *2 *3                             PRESS TIME                                       C.800000
 *1 *2 *3                       TEMP  PRESS TIME                                       C.800000
          *4                                      VEL                                  1.000000
 *1       *4                    TEMP              VEL                                  C.500000
    *2    *4                          PRESS       VEL                                  C
 *1    *3 *4                    TEMP        TIME  VEL                                  C.400000
```

```
    *2 *3 *4                                  PRESS TIME  VEL                                        0
 *1 *2 *3 *4                             TEMP PRESS TIME  VEL                                        C
             *5                                                ANGLE                                 1.000000
 *1          *5                          TEMP                  ANGLE                                 C.400000
    *2       *5                               PRESS            ANGLE                                 C
 *1 *2       *5                          TEMP PRESS            ANGLE                                 0
       *3    *5                                     TIME       ANGLE                                 C.300000
    *2 *3    *5                               PRESS TIME   .   ANGLE                                 C
 *1 *2 *3    *5                          TEMP PRESS TIME       ANGLE                                 C
          *4 *5                                           VEL  ANGLE                                 C
 *1       *4 *5                          TEMP             VEL  ANGLE                                 C
    *2    *4 *5                               PRESS       VEL  ANGLE                                 0
 *1 *2    *4 *5                          TEMP PRESS       VEL  ANGLE                                 C
       *3 *4 *5                                     TIME  VEL  ANGLE                                 C
 *1    *3 *4 *5                          TEMP       TIME  VEL  ANGLE                                 C
 *1 *2 *3 *4 *5                          TEMP PRESS TIME  VEL  ANGLE                                 0

 FOR THE ABOVE PERMUTATION THE EXPECTED UTILITY IS    0.41934

 THE EXPECTED UTILITIES AT THE STOPPING POINTS ARE..
 DEFINING PARAMETER GROUP    1       0.31500
 DEFINING PARAMETER GROUP    2       0.59062
 DEFINING PARAMETER GROUP    3       0.29531



 THIS MATCHING MAXIMIZES THE EXPECTED VALUE AT THE  3 STOPPING POINT FULL--ALL ROWS

VARIABLE             SHOULD BE CALLED
     1 TEMP                     C
     2 PRESS                    D
     3 TIME                     B
     4 VEL                      E
     5 ANGLE                    A

DEFINING PARAMETER GROUP NO.    1
1/4 REP--ROW 1
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *1 *2 *3                                TEMP PRESS TIME                                             C.200000
 0*                                      G MEAN                                                      C
 *1                                      TEMP                                                        C.560000
    *2                                        PRESS                                                  C.400000
          *4                                              VEL                                        0.200000
       *3                                           TIME                                             C.480000
             *5                                                ANGLE                                 C.200000
    *2 *3                                     PRESS TIME                                             C.480000

DEFINING PARAMETER GROUP NO.    2
1/2 REP-- ROWS 1,2
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       *3 *4                                        TIME  VEL                                        0.250000
 0*                                      G MEAN                                                      0
 *1    *3    *5                          TEMP       TIME       ANGLE                                 C
 *1                                      TEMP                                                        C.800000
    *2                                        PRESS                                                  C.800000
 *1 *2                                   TEMP PRESS                                          .       C.800000
       *3                                           TIME                                             C.800000
 *1    *3                                TEMP       TIME                                             C.800000
    *2 *3                                     PRESS TIME                                             C.800000
 *1 *2 *3                                TEMP PRESS TIME                                             C.800000
          *4                                              VEL                                        1.000000
 *1       *4                             TEMP             VEL                                        C.500000
       *3    *5                                     TIME       ANGLE                                 C.300000
 *1    *3 *4                             TEMP       TIME  VEL                                        C.400000
             *5                                                ANGLE                                 1.000000
 *1          *5                          TEMP                  ANGLE                                 C.400000

DEFINING PARAMETER GROUP NO.    3
FULL--ALL ROWS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    *2       *5                               PRESS            ANGLE                                 0
 0*                                      G MEAN                                                      C
 *1    *3    *5                          TEMP       TIME       ANGLE                                 C
    *2 *3 *4 *5                               PRESS TIME  VEL  ANGLE                                 C
 *1 *2    *4                             TEMP PRESS       VEL                                        C
 *1                                      TEMP                                                        C.800000
    *2                                        PRESS                                                  C.800000
 *1 *2                                   TEMP PRESS                                                  0.800000
       *3                                           TIME                                             C.800000
 *1    *3                                TEMP       TIME                                             C.800000
    *2 *3                                     PRESS TIME                                             C.800000
 *1 *2 *3                                TEMP PRESS TIME                                             C.800000
          *4                                              VEL                                        1.000000
 *1       *4                             TEMP             VEL                                        0.500000
    *2    *4                                  PRESS       VEL                                        C
       *3 *4                                        TIME  VEL                                        C.500000
```

```
*1     *3 *4                                TEMP        TIME  VEL                                  C.400000
   *2 *3 *4                                       PRESS TIME  VEL                                  C
*1 *2 *3 *4                                 TEMP  PRESS TIME  VEL                                  C
             *5                                                     ANGLE                          1.000000
*1           *5                             TEMP                    ANGLE                          C.400000
*1 *2        *5                             TEMP  PRESS             ANGLE                          0
      *3     *5                                         TIME        ANGLE                          C.300000
   *2 *3     *5                                   PRESS TIME        ANGLE                          C
*1 *2 *3     *5                             TEMP  PRESS TIME        ANGLE                          C
         *4 *5                                                VEL   ANGLE                          C
*1       *4 *5                              TEMP                VEL   ANGLE                        C
   *2    *4 *5                                    PRESS         VEL   ANGLE                        0
*1 *2    *4 *5                              TEMP  PRESS         VEL   ANGLE                        C
      *3 *4 *5                                          TIME    VEL   ANGLE                        C
*1    *3 *4 *5                              TEMP        TIME    VEL   ANGLE                        C
*1 *2 *3 *4 *5                              TEMP  PRESS TIME    VEL   ANGLE                        C

FOR THE ABOVE PERMUTATION THE EXPECTED UTILITY IS     0.42169

THE EXPECTEC UTILITIES AT THE STOPPING POINTS ARE..
DEFINING PARAMETER GROUP    1        0.31500
DEFINING PARAMETER GROUP    2        0.59062
DEFINING PARAMETER GROUP    3        0.30312



SUMMARY OUTPUT TABLE

      ** BAYES ***   **** 1 *****   **** 2 *****   **** 3 *****   ****

           C              D              C              C
           C              A              B              D
           B              B              D              B
           E              C              A              E
           A              E              E              A


EXPECTED UTILITY
OVER STOPPING PTS


       0.42169        0.37074        0.41934        0.42169

EXPECTED UTILITY
AT EACH STOPPING PT
  1    0.31500        0.39850        0.31500        0.31500
  2    0.59062        0.41000        0.59062        0.59062
  3    0.30312        0.29062        0.29531        0.30312
 CURRENT EXECUTION TIME          0.07
```

APPENDIX B

FORTRAN LISTING

```
$IBFTC BLOK

      BLOCK DATA
      COMMON/BDATA/      POWERS(11),        IALPHA(9),         ALPHA(10),
     X IUNIN,            IUNOUT,            MASK,              NEG
C* * * * * * * * * * * * * * * * * * * * * * * * * * *
      INTEGER POWERS
      DATA (POWERS(I),I=1,11)/0,1,2,4,8,16,32,64,128,256,0/
      DATA ( ALPHA(I),I=1,10)/  1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HJ,1H /
      DATA IUNIN/5/,IUNOUT/6/,MASK/01/,NEG/0400000000000/
      END
```

```
$IBFTC NAMER   DEBUG

      COMMON/BDATA/      POWERS(11),        IALPHA(9),         ALPHA(10),
     X IUNIN,            IUNOUT,            MASK,              NEG
      COMMON/B1/  NFAC,NCLASS,NN,NDPG,PBAYES,IRUN,PBAYEX,IUTILF,UTSWCH
      COMMON/B2/ DPG(128,32),BLOCK(128,32),IDPG(32),NBLOCK(32),IP(9),
     XID(9),NSUBI(9),LPERM(9),II(10),PSTOP(32),WT(32),PSUMX(32),PSUM(32)
      COMMON/REST/PROB(512), T1(512), PBLOCK(128,32), ISAVEP(9,33),
     X UTIL(512), ULIST(128), IULIST(128), IDDCG(4,32)
      INTEGER POWERS, T1, DPG, BLOCK
      LOGICAL UTSWCH, LPERM
      REAL IDENT
      EQUIVALENCE (X,IX)
      DIMENSION DUMP1(1),DUMP2(1)
      EQUIVALENCE (DUMP1(1),NFAC),(DUMP2(1),IDPG(1))
      DIMENSION IT1(9),XT3(9) ,IDENT(14)
      DATA BLANK/6H      /,ENDCRD/6HENDALL/
      COMMON/LINX/ XNOUT(5),HOLOUT(9),FAC(14,9)
C
C*****************************************************************
      LD1= 9
      CALL TIME1(TSTART)                                                   3
      TMAX= 0.0
      LD2 = 238
C
C*****************************************************************
C
C     NAMER SECTION  1
C
   10 READ(IUNIN,5000) IDENT                                               5
      IF(IDENT(1).EQ.ENDCRD) STOP
      WRITE(IUNOUT,5005) IDENT
C
C*****************************************************************
C
C     NAMER SECTION  2
C                                                                          8
      DO 12 J=1,5
   12 XNOUT(J)=BLANK
      DO 14 J=1,9
      IALPHA(J)= J
```

```
   14 HOLOUT(J)=BLANK
      READ(IUNIN,5010) TMAXX                                                    24
      TMAX= TMAX + TMAXX
      WRITE(IUNOUT,5020) TMAXX                                                  25
      CALL TIME1(TPRINT)                                                        26
      TPRINT=(TPRINT-TSTART)/3600.0
      WRITE(IUNOUT,5025) TPRINT
C
C************************************************************************
C
C     NAMER SECTION  3
C                                                                               28
      READ(IUNIN,5045) ITYPRN                                                   29
      GO TO (30,154,20),ITYPRN
   20 READ(IUNIN,5070) (XT3(K),K=1,9)
C
C************************************************************************
C
C     NAMER SECTION  4
C                                                                               32
   30 READ(IUNIN,5045) NFAC                                                     39
      IF((NFAC.LT.1) .OR. (NFAC.GT.9))  GO TO 8020
      WRITE(IUNOUT,5050) NFAC                                                   43
      IF(ITYPRN.NE.3) GO TO 38
      DO 35 K=1,NFAC
      DO 33 L=1,NFAC
      LL=L
      IF(XT3(K).EQ.ALPHA(L)) GO TO 34
   33 CONTINUE
      GO TO 8010
   34 ISAVEP(K,1)=LL
   35 CONTINUE
   38 CONTINUE
C
C************************************************************************
C
C     NAMER SECTION  5
C
      DO 40 J=1,NFAC
      READ(IUNIN,5000) (FAC(I,J),I=1,14)                                        69
      WRITE(IUNOUT,5055) J,(FAC(I,J),I=1,14)                                    74
   40 CONTINUE
C
C************************************************************************
C
C     NAMER SECTION  6
C
      II(1)=1
      LPERM(1)=.TRUE.
      NSUBI(1)=NFAC
      READ(IUNIN,5045) NCLASS                                                   82
      IF((NCLASS.LT.1).OR.(NCLASS.GT.9))  GO TO 8030
      IF(NCLASS-1) 60,60,50
   50 READ(IUNIN,5045) (NSUBI(I),I=1,NCLASS)                                    88
      WRITE(IUNOUT,7000) NCLASS,(NSUBI(I),I=1,NCLASS)                           95
      DO 55 J=1,NCLASS
      LPERM(J)=.TRUE.
      II(J+1)= II(J)+NSUBI(J)
   55 CONTINUE
   60 NN= 2**NFAC                                                               113
      DO 65 J=1,NN
      TT(J)=0
      PROB(J)=1.0
      UTIL(J)=0.0
   65 CONTINUE
C
C************************************************************************
```

```
C
C     NAMER SECTION  7
C
      READ(IUNIN,5043) NPIN,IUTILF,UCOEF                                123
      WRITE(IUNOUT,6045) NPIN,IUTILF                                    126
      UTSWCH=.FALSE.
      IF((IUTILF.LT.1) .OR. (IUTILF.GT.5))  GO TO 8035
      IF((NPIN.LE.0).OR.(NPIN.GT.NN)) GO TO 8032
      IF((UCOEF.LT.0.0).OR.(UCOEF.GT.1.0)) GO TO 8033
      IF(IUTILF.GE.3)  UTSWCH= .TRUE.
      IF(IUTILF.EQ.5)  WRITE(IUNOUT,7010) UCOEF                         138
      DO 90 J=1,NPIN
      READ(IUNIN,5060) (IT1(I),I=1,9),P,UT                              142
      IF((P.GT.1.0).OR.(P.LT.0.0)) GO TO 8038
      I=0
      IT=0
      KK=0
      DO 67 K=1,9
      KKX= 10-K
      KI= IT1(KKX) + 1
      I=I + POWERS(KI)
      IF(KI-1) 67,67,66
   66 IT= IT + IT1(KKX)*10**KK                                          162
      KK=KK+1
   67 CONTINUE
      I= I+1
   81 GO TO (82,84,86,88,87),IUTILF
   82 UTIL(I)=1.0
      GO TO 89
   84 UTIL(I)=P
      GO TO 89
   86 UTIL(I)=UT
      GO TO 89
   87 UTIL(I)= UCOEF*UT + (1.0-UCOEF)*P
      GO TO 89
   88 UTIL(I)=UT*P
   89 CONTINUE
      WRITE(IUNOUT,6060) IT,P,UTIL(I)                                   182
      PROB(I)=1.0 - P
   90 CONTINUE
C
C*******************************************************************
C
C     NAMER SECTION  8
C
      READ(IUNIN,5045) NDPG                                             187
      IF((NDPG.LT.1).OR.(NDPG.GT.32)) GO TO 8140
      WRITE(IUNOUT,6065) NDPG                                           191
      DO 150  I=1,NDPG
      READ(IUNIN,5000) (IDDCG(K,I),K=1,4)                               194
      WRITE(IUNOUT,6064) (IDDCG(K,I),K=1,4)                             199
      READ(IUNIN,5065) NGEN,PSTOP(I),WT(I)                              204
      IF((NGEN.LT.0).OR.(NGEN.GT.MINO(NFAC,7))) GO TO 8150
      IF((PSTOP(I).LT.0.0).OR.(PSTOP(I).GT.1.0)) GO TO 8160
      WRITE(IUNOUT,6066) I,NGEN,PSTOP (I),WT(I)                         215
      IDPG(I)= 2**NGEN-1
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     NAMER SECTION  8A
C                                                                       218
      IF(NGEN) 106,106,91
   91 DO 105 J=1,NGEN
      READ(IUNIN,5070) (XT3(K),K=1,9)                                   223
      WRITE(IUNOUT,6070) (XT3(K),K=1,9)                                 228
      KI=0
      DO 100 K=1,9
```

```
      DO 94  L=1,10
      LL=L
      IF(XT3(K).EQ ALPHA(L)) GO TO 98
   94 CONTINUE
      GO TO 8050
   98 KKI=LL+1
      KI = KI + POWERS(KKI)
  100 CONTINUE
      DPG(J,I)= KI
  105 CONTINUE
  106 IDI= IDPG(I)
      IF(IDI-1) 121,121,108
  108 KP=4
      N1= NGEN
      LPTR=NGEN+1
      LLEN=0
      DO 120 J=2,NGEN
      KK=J-1
      IF(DPG(J,I).EQ.DPG(1,I)) GO TO 8060
      DO 110 K=1,KK
      DPG(LPTR,I)=IEXOR(DPG(J,I),DPG(K,I))                                          275
      IF(DPG(LPTR,I).EQ DPG(1,I)) GO TO 8060
  110 LPTR=LPTR+1
      IF(LLEN.EQ.0) GO TO 118
      DO 115 K=1,LLEN
      N=NGEN+K
      DPG(LPTR,I)=IEXOR(DPG(J,I),DPG(N,I))                                          292
      IF(DPG(LPTR,I).EQ.DPG(1,I)) GO TO 8060
  115 LPTR=LPTR+1
  118 LLEN=2*LLEN+KK
  120 CONTINUE
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      NAMER SECTION   8B
C
  121 READ(IUNIN,5045) NBLOCK(I)                                                    3C7
      NB = NBLOCK(I)
      IF((NB.LT.0).OR.(NB.GT.NN)) GO TO 8170
      WRITE(IUNOUT,6080) NB                                                         313
      IF(NB) 150,150,122
  122 DO 140 J=1,NB
      READ(IUNIN,5070) (XT3(K),K=1,9),PBLOCK(J,I)                                   317
      IF((PBLOCK(J,I).LT.0.0).OR.(PBLOCK(J,I).GT.1.0)) GO TO 8180
      WRITE(IUNOUT,6070)(XT3(K),K=1,9),PBLOCK(J,I)                                  327
      KI=0
      DO 130 K=1,9
      DO 124 L=1,10
      LL=L
      IF(XT3(K).EQ.ALPHA(L)) GO TO 128
  124 CONTINUE
      GO TO 8070
  128 KKI=LL+1
      KI = KI + POWERS(KKI)
  130 CONTINUE
      BLOCK(J,I) = KI
  140 CONTINUE
  150 CONTINUE
      IF(ITYPRN.EQ.3)  CALL ONCE($10)
C
C*****************************************************************
C
C      NAMER SECTION    9
C                                                                                   359
      PBAYES=0.0
      DO 152 I=1,NDPG
      PSUM(I)=0.0
```

```
  152 CONTINUE
      IRUN = 20
      CALL TIME1(TPRINT)                                                370
      TPRINT=(TPRINT-TSTART)/3600.0
      WRITE(IUNOUT,5025) TPRINT                                         372
      GO TO 158
C
C***********************************************************************
C
C     NAMER SECTION  10
C
  154 CONTINUE
      WRITE(IUNOUT,5040)                                                374
      CALL BCREAD(DUMP1(1),DUMP1(LD1))                                  376
      CALL BCREAD(DUMP2(1),DUMP2(LD2))                                  379
      CALL BCREAD(IALPHA(1),IALPHA(NFAC))                               382
      CALL BCREAD(FAC(1,1),FAC(14,NFAC))                                385
      CALL BCREAD (PROB(1),PROB(NN))                                    388
      CALL BCREAD(UTIL(1),UTIL(NN))                                     391
      CALL BCREAD (ISAVEP(1,1),ISAVEP(NFAC,1))                          394
      CALL BCREAD(IDDCG(1,1),IDDCG(4,NDPG))                             397
      DO 156 I = 1,NDPG
      IDI = IDPG(I)
      NB=NBLOCK(I)
      IF(IDI.NE.0)CALL BCREAD(DPG(1,I),DPG(IDI,I))                      409
      CALL BCREAD (ISAVEP(1,I+1),ISAVEP(NFAC,I+1))                      413
      IF(NB.EQ.0) GO TO 156
      CALL BCREAD(PBLOCK(1,I),PBLOCK(NB ,I))                            420
      CALL BCREAD(BLOCK(1,I),BLOCK(NB ,I))                              424
  156 CONTINUE
C
C***********************************************************************
  158 CONTINUE
C
C***********************************************************************
C
C     NAMER SECTION  11
C     EVALUATE THE CURRENT ORDERING
C
      DO 1000 I=1,NDPG
C
C* * * * * * * * * * * *     * * * * *
C
C     INITIALIZATIONS
C
      IRUN=IRUN+1
      PSUMX(I)=0.0
      NB=NBLOCK(I)
      IDI = IDPG(I)
      IDI1=IDI+1
C
C*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
C
C     NAMER SECTION 11A
C
C     CHECK FOR BLOCK CONFOUNDING. IF THERE ARE NO BLOCK PARAMETERS
C     GO TO NAMER SECTION 11B
C
  165 IF(NB) 228,228,166
  166 DO 226  K=1,NB
      KI= BLOCK(K,I)
      KI1 = KI+1
      T1(KI1)= IRUN
C
C-----------------------------------------------------------------------
C
C     IS THIS A FULL OR FRACTIONAL FACTORIAL
```

```
C
      IF(IDI) 2035,167,171
  167 PSUMX(I)= PSUMX(I)+UTIL(KI1)*(1.0-PBLOCK(K,I))
      GO TO 226
C
C     IF BLOCK PARAMETER HAS PRIOR PROB=1.0 THERE CAN BE NO UTILITY
C     FOR THIS ESTIMATOR.  JUST TAG ALIASES.
C
  171 IF(PBLOCK(K,I)-1.0) 176,173,173
  173 DO 174  IK= 1,IDI
      JJ = IEXOR(KI,DPG(IK,I))+1                                    461
      T1(JJ)= IRUN
  174 CONTINUE
      GO TO 226
C
C-----------------------------------------------------------------
C
C     INITIALIZE BEFORE FINDING OPTIMAL MATCHING OF PARAMETER TO ESTIMAT
C
  176 PS= PROB(KI1)
      PR= PS
      IULIST(1)= KI1
      ULIST(1)= UTIL(KI1)
      ISTAR=0
      IF(PS) 178,178,179
  178 ISTAR=1
      ULIST(1)= -UTIL(KI1)
      PR=1.0
  179 IMAX= KI1
C
C-----------------------------------------------------------------
C
C     FOR UTILITY FUNCTIONS 1 AND 2, PARAMETER WITH MAXIMUM PROBABILITY
C     HAS MAXIMUM UTILITY.
C
      IF(UTSWCH) GO TO 200
      DO 187  KK=1,IDI
      JJ = IEXOR(KI,DPG(KK,I))+1                                    484
      T1(JJ)= IRUN
      IF(PROB(JJ)) 2035,184,182
  182 PR= PR*PROB(JJ)
      IF(PROB(JJ)-PS) 186,186,187
  184 ISTAR= ISTAR+1
  186 PS= PROB(JJ)
      IMAX= JJ
  187 CONTINUE
C
C     COMPUTE UTILITY
C
      IF(ISTAR-1) 190,192,226
  190 PR= PR/PROB(IMAX)
  192 PSUMX(I)=PSUMX(I)+PR*UTIL(IMAX)*(1.0-PBLOCK(K,I))
      GO TO 226
C
C-----------------------------------------------------------------
C
C     FOR ARBITRARY UTILITY FUNCTIONS COMPUTE UTILITY FOR EACH MATCHING
C
  200 DO 206 KK=2,IDI1
      JJ=IEXOR(KI,DPG(KK-1,I))+1                                    511
      T1(JJ)= IRUN
      IULIST(KK)= JJ
      IF(PROB(JJ)) 204,204,202
  202 PR= PR * PROB(JJ)
      ULIST(KK)= UTIL(JJ)
      GO TO 206
```

```
  204 ISTAR= ISTAR+1
      ULIST(KK)= -UTIL(JJ)
  206 CONTINUE
C
C     FIND MATCHING THAT MAXIMIZES UTILITY
C
      IF(ISTAR-1) 208,214,226
  208 DO 210 KK=1,IDI1
      JJ= IULIST(KK)
      ULIST(KK)= ULIST(KK)*PR/PROB(JJ)
  210 CONTINUE
      UMAX=0.0
      DO 212 KK=1,IDI1
      IF(ULIST(KK) - UMAX) 212,212,211
  211 UMAX= ULIST(KK)
  212 CONTINUE
      PSUMX(I)= PSUMX(I)+UMAX*(1.0 - PBLOCK(K,I))
      GO TO 226
C
C
  214 DO 216 KK=1,IDI1
      KS= KK
      TST= AND(ULIST(KK),NEG)
      TST = OR(TST,MASK)
      IF(TST) 218,2035,216
  216 CONTINUE
  218 PSUMX(I)= PSUMX(I) - ULIST(KS)*PR*(1.0-PBLOCK(K,I))
  226 CONTINUE
C
C* * * * * * * * * * * * * * * * * *
C
  228 IF(IDI) 229,229,245
C
C*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
C
C     NAMER SECTION  11B
C     BLOCK PARAMETERS HAVE NOW BEEN ACCOUNTED FOR.
C     CONTINUE COMPUTING UTILITY FOR REMAINING PARAMETERS.
C     COMPUTE UTILITY FOR FULL FACTORIAL
C
  229 IF(IUTILF.NE.1)  GO TO 230
      PSUMX(I)= PSUMX(I)+FLOAT(NN-NBLOCK(I))
      GO TO 1000
  230 DO 234  K=1,NN
      IF(T1(K)-IRUN) 231,234,231
  231 PSUMX(I)=PSUMX(I)+UTIL(K)
  234 CONTINUE
      GO TO 1000
C
C-----------------------------------------------------------------------
C     COMPUTE UTILITY FOR FRACTIONS
C
C     FIND NEXT UNTAGGED PARAMETER
C
  245 DO 700  K=1,NN
      IF(T1(K)-IRUN) 246,700,246
  246 PS= PROB(K)
      KM1= K-1
      PR=PS
      ISTAR=0
      IULIST(1)= K
      ULIST(1)= UTIL(K)
      IF(PS) 248,248,250
  248 ISTAR=1
      ULIST(1)= -UTIL(K)
      PR=1.0
```

```
  250 IMAX=K
      IF(UTSWCH) GO TO 430
C
C-----------------------------------------------------------------------
C
C     FOR UTILITY FUNCTIONS 1 AND 2, PARAMETER WITH MAXIMUM PROBABILITY
C     HAS MAXIMUM UTILITY.
C
      DO 410  KK=1,IDI
      JJ = IEXOR(KM1,DPG(KK,I))+1                                          609
      T1(JJ)=IRUN
      IF(PROB(JJ)) 380,390,380
  380 PR= PR*PROB(JJ)
      IF(PROB(JJ)-PS) 400,410,410
  390 ISTAR=ISTAR+1
  400 PS=PROB(JJ)
      IMAX=JJ
  410 CONTINUE
      GO TO 505
C
C-----------------------------------------------------------------------
C
C     FOR ARBITRARY UTILITY FUNCTIONS COMPUTE UTILITY FOR EACH MATCHING
C
  430 DO 460 KK=2,IDI1
      JJ= IEXOR(KM1,DPG(KK-1,I))+1                                         628
      T1(JJ)= IRUN
      IULIST(KK)= JJ
      IF(PROB(JJ)) 450,450,440
  440 PR= PR*PROB(JJ)
      ULIST(KK)= UTIL(JJ)
      GO TO 460
  450 ISTAR= ISTAR+1
      ULIST(KK)= -UTIL(JJ)
  460 CONTINUE
      GO TO 600
C
C-----------------------------------------------------------------------
C
C     INCREMENT UTILITY OF THIS DCG BY UTILITY OF ESTIMATOR
C
  505 IF(ISTAR-1)  520,530,700
  520 PR= PR/PROB(IMAX)
  530 PSUMX(I)=PSUMX(I)+PR*UTIL(IMAX)
      GO TO 700
C
C
C
  600 IF(ISTAR-1)  620,660,700
  620 DO 630 KK=1,IDI1
      JJ= IULIST(KK)
      ULIST(KK)= ULIST(KK) * PR/PROB(JJ)
  630 CONTINUE
      UMAX=0.0
      DO 640 KK=1,IDI1
      IF(ULIST(KK)-UMAX) 640,640,635
  635 UMAX= ULIST(KK)
  640 CONTINUE
      PSUMX(I)= PSUMX(I)+UMAX
      GO TO 700
  660 DO 670 KK=1,IDI1
      KS= KK
      TST=AND(ULIST(KK),NEG)
      TST=OR(TST,MASK)
      IF(TST) 680,2035,670
  670 CONTINUE
  680 PSUMX(I)= PSUMX(I) - ULIST(KS)*PR
```

```
  700 CONTINUE
 1000 CONTINUE
C
C*********************************************************************
C
C     NAMER  SECTION  12
C
      PBAYEX= 0.0
      DO 1020 I=1,NDPG
      PSUMX(I)= PSUMX(I)*WT(I)
      PBAYEX= PBAYEX + PSUMX(I)*PSTOP(I)
 1020 CONTINUE
      IF(PBAYEX-PBAYES) 1060,1060,1030
 1030 PBAYES=PBAYEX
      DO 1050 I=1,NFAC
      ISAVEP(I,1)=IALPHA(I)
 1050 CONTINUE
 1060 DO 1100 I=1,NDPG
      IF(PSUMX(I)-PSUM(I)) 1100,1100,1070
 1070 PSUM(I)=PSUMX(I)
      DO 1080 J=1,NFAC
      ISAVEP(J,I+1)= IALPHA(J)
 1080 CONTINUE
 1100 CONTINUE
C
C*********************************************************************
C
C     NAMER  SECTION  13
C
      DO 1150 K=1,NCLASS
      ISEND=II(K)
      CALL PERMUT(IALPHA(ISEND),NSUBI(K  ),LPERM(K  ),ISEND)        742
      IF(LPERM(K)) GO TO 1150
      GO TO 2000
 1150 CONTINUE
      GO TO 3000
C
C*********************************************************************
C
C     NAMER  SECTION  14
C
 2000 CALL TIME1(TNOW)                                               755
      T=(TNOW-TSTART)/3600.
      IF(T-TMAX) 158,2010,2010
C*********************************************************************
C
C     NAMER  SECTION  15
C
 2010 CONTINUE
      WRITE(IUNOUT,6090) T,TRUN                                      758
      CALL  BCDUMP(DUMP1(1),DUMP1(LD1),0)                            760
      CALL  BCDUMP(DUMP2(1),DUMP2(LD2),0)                            763
      CALL  BCDUMP(IALPHA(1),IALPHA(NFAC),0)                         766
      CALL  BCDUMP(FAC(1,1),FAC(14,NFAC),0)                          769
      CALL  BCDUMP(PROB(1), PROB(NN),0)                              772
      CALL  BCDUMP(UTIL(1),UTIL(NN),0)                               775
      CALL  BCDUMP (ISAVEP(1,1),ISAVEP(NFAC,1),0)                    778
      CALL  BCDUMP(IDDCG(1,1),IDDCG(4,NDPG),0)                       781
      DO 2030 I =1,NDPG
      IDI=IDPG(I)
      NB=NBLOCK(I)
      IF(IDI.NE.0) CALL BCDUMP(DPG(1,I),DPG(IDI,I),0)                793
      CALL BCDUMP (ISAVEP(1,I+1),ISAVEP(NFAC,I+1),0)                 797
      IF(NB.EQ.0) GO TO 2030
      CALL BCDUMP(PBLOCK(1,I),PBLOCK(NB ,I),0)                       804
      CALL BCDUMP(BLOCK(1,I),BLOCK(NB ,I),0)                         808
 2030 CONTINUE
 2035 STOP
```

```
C
C*********************************************************************
C
C     NAMER SECTION 16
C
 3000 CONTINUE
      CALL TIME1(TPRINT)                                              814
      TPRINT=(TPRINT-TSTART)/3600.0
      WRITE(IUNOUT,5025) TPRINT                                       816
      CALL REMACH                                                     817
      CALL TIME1(TPRINT)                                              819
      TPRINT=(TPRINT-TSTART)/3600.0
      WRITE(IUNOUT,5025) TPRINT                                       821
      GO TO 10
C
C*********************************************************************
 5000 FORMAT(13A6,1A2)
 5005 FORMAT(14H1NAMER OUTPUT 13A6,1A2/1H )
 5010 FORMAT(F6.0)
 5020 FORMAT(50H PROGRAM WILL DUMP FOR RESTART IF NOT FINISHED IN F10.0,
     X  9H MINUTES. /1H )
 5025 FORMAT(25H  CURRENT EXECUTION TIME   F12.2)
 5030 FORMAT(L1)
 5040 FORMAT(38H THIS IS A RESTART OF A PREVIOUS CASE /1H )
 5043 FORMAT(2I6,F10.9)
 5045 FORMAT(9I6)
 5050 FORMAT(11H THERE ARE  I4,22H FACTORS.  THEY ARE... /1H )
 5055 FORMAT(1X,I10,3X,13A6,A2)
 5060 FORMAT(9I1,2F10.0)
 5065 FORMAT(I6,F6.6,F6.0)
 5070 FORMAT(9A1,F5.5)
 6045 FORMAT(1HK I6,61H PARAMETERS WITH NON-ZERO PRIOR PROBABILITIES AND
     X UTILITIES   /1X/19H  UTILITY FUNCTION I2)
 6060 FORMAT(1X,I9,2G14.6)
 6065 FORMAT(1HK I6,26H DEFINING PARAMETER GROUPS   /1X)
 6064 FORMAT (1HK 120(1H-)/1HK 4A6)
 6066 FORMAT(4HKDPG I3/4H      I3,11H GENERATORS /
     X 21H    PROB OF STOPPING F10.5,13H        WEIGHT  F10.6)
 6070 FORMAT(5X, 9A1, F10.5)
 6080 FORMAT(10HKTHERE ARE  I4, 17H BLOCK PARAMETERS  /1X)
 6090 FORMAT(35H TIME EXCEEDED. DUMPING FOR RESTART  /13H EXEC. TIME
     X  F7.2,5H MIN. /8H IRUN = I15)
 7000 FORMAT(11HKTHERE ARE I3,21H CLASSES OF VARIABLES  /5X,9I6)
 7010 FORMAT(9HK UCOEF=  F12.9)
C*********************************************************************
C     ERROR MESSAGES
C
 8010 WRITE(IUNOUT,9010)                                              823
 9010 FORMAT(40H ILLEGAL CHARACTER IN SPECIFIED MATCHING )
      GO TO 2035
 8020 WRITE(IUNOUT,9020) NFAC                                         825
 9020 FORMAT(33H NUMBER OF FACTORS OUT OF RANGE  I6)
      GO TO 2035
 8030 WRITE(IUNOUT,9030) NCLASS                                       827
 9030 FORMAT(29H NUMBER CLASSES OUT OF RANGE I6)
      GO TO 2035
 8032 WRITE(IUNOUT,9032) NPIN                                         829
 9032 FORMAT(48H IMPROPER NUMBER OF NONZERO PRIOR PROBABILITIES I6)
      GO TO 2035
 8033 WRITE(IUNOUT,9033) UCOEF                                        831
 9033 FORMAT(20H UCOEF OUT OF RANGE G14.6)
      GO TO 2035
 8035 WRITE(IUNOUT,9035) IUTILF                                       833
 9035 FORMAT(33H UTILITY FUNCTION CHOICE ILLEGAL I6)
      GO TO 2035
 8038 WRITE(IUNOUT,9038) P                                            835
```

```
 9038 FORMAT(33H ILLEGAL INPUT PRIOR PROBABILITY G14.6)
      GO TO 2035
 8050 WRITE( IUNOUT,9050)                                                  837
 9050 FORMAT(52H ILLEGAL CHARACTER USED TO INPUT A GENERATOR FOR DPG)
      GO TO 2035
 8060 WRITE( IUNOUT,9060)                                                  839
 9060 FORMAT(28H GENERATORS NOT INDEPENDENT )
      GO TO 2035
 8070 WRITE( IUNOUT,9070)                                                  841
 9070 FORMAT(45H ILLEGAL CHARACTER USED TO INPUT BLOCK EFFECT )
      GO TO 2035
 8140 WRITE( IUNOUT,9140)                                                  843
 9140 FORMAT(20H INVALID NO. OF DPGS )
      GO TO 2035
 8150 WRITE( IUNOUT,9150)                                                  845
 9150 FORMAT(26H INVALID NO. OF GENERATORS )
      GO TO 2035
 8160 WRITE( IUNOUT,9160)                                                  847
 9160 FORMAT(27H STOPPING PROB OUT OF RANGE )
      GO TO 2035
 8170 WRITE( IUNOUT,9170)                                                  849
 9170 FORMAT(29H INVALID NO. OF BLOCK EFFECTS )
      GO TO 2035
 8180 WRITE( IUNOUT,9180)                                                  851
 9180 FORMAT(31H BLOCK EFFECT PROB OUT OF RANGE )
      GO TO 2035
      END
```

```
$IBFTC LINER

      SUBROUTINE LINE(I,U)
      COMMON /BDATA/POWERS(11),IALPHA(9),ALPHA(10),IUNIN,IUNOUT,MASKK,NEG
      COMMON /B1/  NFAC,NCLASS,NN,NDPG,PBAYES,IRUN,PBAYEX,IUTILF,UTSWCH
      COMMON /LINX/ XNOUT(5),HOLOUT(9),FAC(14,9)
      DIMENSION XNUMER(9),BLANK(3),BLOCKS(4)
      DATA(BLOCKS(I),I=1,4)/6HCONFOU,6HNDED W,6HITH BL,6HOCKS   /
      DATA(BLANK(I),I=1,3)/0777777606060,0606060777777,6H       /
      DATA(XNUMER(I),I=1,9)/  0605401777777, 0777777605402,
     X   0605403777777, 0777777605404, 0605405777777, 0777777605406 ,
     X   0605407777777, 0777777605410, 0605411777777  /
       DATA MASK/03/,ZERO/6H 0*    /,GMEAN/6HG MEAN/
      EQUIVALENCE (X,IX) ,(Y,IY)
C
C*******************************************************************
      IX=I
      IF(IX.EQ.0) WRITE(IUNOUT,5005) ZERO,GMEAN,U                          3
      IF(IX.EQ.0) RETURN
      NF= NFAC
      JJ=1
      NF1=NF-1
      DO 100 J=1,NF1,2
      NF= NF-2
      Y=AND(MASK,X)
      IY=IY+1
C
      GO TO (20,40,60,80),IY
   20 XNOUT(JJ)= BLANK(3)
      HOLOUT(J)= BLANK(3)
      HOLOUT(J+1)= BLANK(3)
      GO TO 95
```

```
C
   40 XNOUT(JJ)= AND (BLANK(1),XNUMER(J))
      HOLOUT(J)= FAC(1,J)
      HOLOUT(J+1)= BLANK(3)
      GO TO 95
C
   60 XNOUT(JJ)= AND(BLANK(2),XNUMER(J+1))
      HOLOUT(J)= BLANK(3)
      HOLOUT(J+1)= FAC(1,J+1)
      GO TO 95
C
   80 XNOUT(JJ)= AND(XNUMER(J),XNUMER(J+1))
      HOLOUT(J)= FAC(1,J)
      HOLOUT(J+1)= FAC(1,J+1)
C
   95 IX=IARS(2,X)                                                       43
      JJ= JJ+1
  100 CONTINUE
      IF(NF) 500,130,105
  105 X= AND(MASK,X)
      IX= IX+1
      GO TO (110,120), IX
  110 XNOUT(JJ)= BLANK(3)
      HOLOUT(NFAC)= BLANK(3)
      GO TO 130
  120 XNOUT(JJ)= AND(BLANK(1),XNUMER(NFAC))
      HOLOUT(NFAC)= FAC(1,NFAC)
C
C
  130 CONTINUE
      WRITE(IUNOUT,5000)(XNOUT(I),I=1,5),(HOLOUT(I),I=1,9),U             60
  499 RETURN
 5000 FORMAT(1H 5A6,6X,9A6,G14.6)
 5005 FORMAT(1H A6,30X,A6,48X,G14.6)
 5010 FORMAT(1H 40X,A6)
      ENTRY  ALINE(IA,IALIAS)
      DIMENSION IALIAS(1)
      DO 800 K=1,IA
      IX=IALIAS(K)
      IF(IX.EQ.0) WRITE(IUNOUT,5010) ZERO                                80
      IF(IX.EQ.0) RETURN
      NF = NFAC
      JJ = 1
      NF1= NF-1
      DO 700 J=1,NF1,2
      NF = NF-2
      Y= AND(MASK,X)
      IY= IY+1
      GO TO (620,640,660,680),IY
  620 XNOUT(JJ)=BLANK(3)
      GO TO 695
  640 XNOUT(JJ)= AND(BLANK(1),XNUMER(J))
      GO TO 695
  660 XNOUT(JJ)=AND(BLANK(2),XNUMER(J+1))
      GO TO 695
  680 XNOUT(JJ) = AND(XNUMER(J),XNUMER(J+1))
  695 IX=IARS(2,X)                                                       107
      JJ=JJ+1
  700 CONTINUE
      IF(NF) 500,730,705
  705 X=AND(MASK,X)
      IX=IX+1
      GO TO (710,720),IX
  710 XNOUT(JJ)= BLANK(3)
      GO TO 730
  720 XNOUT(JJ)= AND(BLANK(1),XNUMER(NFAC))
  730 WRITE(IUNOUT,850) (XNOUT(I),I=1,5)                                 120
```

```
  800 CONTINUE
  850 FORMAT (40X,5A6)
      RETURN
  500 STOP
      END
```

```
$IBFTC ERECT
      SUBROUTINE RECT(N,X,Y,NFAC)
      DIMENSION X(9,33),Y(33,33),XOUT(9)
      COMMON/PDATA/POWERS(11),IALPHA(9),ALPHA(10),IUNIN,IUNOUT,MASK,NEG
      DATA J9/9/
      LOGICAL OUT
      OUT= .FALSE.
      JTIMES=0
      JCOL= N
      IW=1
    5 IWW=1
      JNXT=JCOL-9
      IF(JNXT) 10,10,30
   10 JP= JCOL
      OUT=.TRUE.
      GO TO 50
   30 JCOL= JNXT
      JP=J9
   50 GO TO (52,54), IW
   52 IW=JP-1
      WRITE(IUNOUT,1001) (J,J=1,IW)                                       15
      WRITE(IUNOUT,1002)                                                  21
      IW=2
      GO TO 56
   54 J1=JTIMES
      JL=J1+JP-1
      WRITE(IUNOUT,1003) (J,J=J1,JL)                                      25
      WRITE(IUNOUT,1002)                                                  31
   56 DO 100  I=1,NFAC
      DO 70  J=1,JP
      JJ=JTIMES+J
   70 XOUT(J)= X(I,JJ)
      WRITE(IUNOUT,1005)(XOUT(K),K=1,JP)                                  41
  100 CONTINUE
      WRITE(IUNOUT,1002)                                                  48
      DO 200  I=1,N
      DO 170  J=1,JP
      JJ= JTIMES+J
  170 XOUT(J)= Y(I,JJ)
      GO TO (172,174), IWW
  172 WRITE(IUNOUT,1007)(XOUT(K),K=1,JP)                                  59
      IWW=2
      WRITE(IUNOUT,1008)                                                  65
      GO TO 200
  174 J1 = I-1
      WRITE(IUNOUT,1009)J1,(XOUT(K),K=1,JP)                               68
  200 CONTINUE
      IF(OUT) RETURN
      JTIMES= JTIMES+JP
      GO TO 5
 1001 FORMAT(21H1SUMMARY OUTPUT TABLE / 1HK/8X,12H** BAYES ***,8(6H  ***
     X* I2,6H *****))
 1002 FORMAT(1H )
 1003 FORMAT(21H1SUMMARY OUTPUT TABLE /1HK/9(6H  ****I2,6H *****))
```

```
1005 FORMAT(6X,9(7X,A1,6X))
1007 FORMAT(17HKEXPECTED UTILITY /18H OVER STOPPING PTS /1HK/
    X  6X,9G14.5)
1008 FORMAT(17HKEXPECTED UTILITY /20H AT EACH STOPPING PT )
1009 FORMAT (1H I3,2X,9G14.5)
     END
```

```
$IBFTC PERM     DEBUG
      SUBROUTINE PERMUT(IA,N,LOG,ISEND)
      COMMON/BDATA/     POWERS(11),        IALPHA(9),        ALPHA(10),
     X IUNIN,           IUNOUT,            MASK,             NEG
      COMMON/B1/  NFAC,NCLASS,NN,NDPG,PBAYES,IRUN,PBAYEX,IUTILF,UTSWCH
      COMMON/B2/ DPG(128,32),BLOCK(128,32),IDPG(32),NBLOCK(32),IP(9),
     XID(9),NSUBI(9),LPERM(9),II(10),PSTOP(32),WT(32),PSUMX(32),PSUM(32)
      INTEGER  POWERS,DPG,BLOCK
      LOGICAL LOG
      EQUIVALENCE (X,IX),(SI,IS),(SJ,JS)
      DIMENSION IA(1),MASK1(15),MASK2(15)
      EQUIVALENCE (MASK1,POWERS(2))
      DATA(MASK2(I),I=1,9)/07777777774,07777777771,07777777763,
     X  07777777747,07777777717,07777777637,07777777477,
     X  07777777177,07777776377 /
C
C*******************************************************************
      NT=N
      IF(NT-1) 500,500,5
    5 IF(.NOT.LOG) GO TO 20
      DO 10  K=2,NT
      IP(K)= 0
      ID(K)=1
   10 CONTINUE
      LOG= .FALSE.
C
C*******************************************************************
   20 K=0
   30 IQ= IP(NT) + ID(NT)
      IP(NT)= IQ
      IF(IQ-NT) 80,40,80
   40 ID(NT)= -1
   45 IF(NT-2)60,60,50
   50 NT=NT-1
      GO TO 30
   60 IQ=1
      LOG= .TRUE.
      GO TO 150
   80 IF(IQ) 150,85,150
   85 ID(NT)=1
      K= K+1
      GO TO 45
C
C*******************************************************************
  150 IQ= IQ+K
      I= IA(IQ)
      IA(IQ)= IA(IQ+1)
      IA(IQ+1)= I
      I2= ISEND+IQ
      M1= MASK1(I2-1)
      M2= MASK1(I2)
      M3= MASK2(I2-1)
      DO 400 I=1,NDPG
      IDI=IDPG(I)
      IB= NBLOCK(I)
```

```
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF(IDI) 240,240,190
  190 DO 200  M=1,IDI
      IX = DPG(M,I)
      SI= AND(M1,IX)
      SJ= AND(M2,IX)
      X= AND(M3,IX)
      IS= IS*2
      JS=JS/2
      DPG(M,I)=IX+IS+JS
  200 CONTINUE
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * *
  240 IF(IB) 400,400,250
  250 DO 300  M=1,IB
      IX= BLOCK(M,I)
      SI= AND(M1,IX)
      SJ= AND(M2,IX)
      X= AND(M3,IX)
      IS= IS*2
      JS=JS/2
      BLOCK(M,I) = IX+IS+JS
  300 CONTINUE
  400 CONTINUE
  500 RETURN
      END
```

```
$IBFTC REMAX    DEBUG

      SUBROUTINE REMACH
      COMMON/BDATA/      POWERS(11),          IALPHA(9),           ALPHA(10),
     X IUNIN,            IUNOUT,              MASK,                NEG
      COMMON/B1/  NFAC,NCLASS,NN,NDPG,PBAYES,IRUN,PBAYEX,IUTILF,UTSWCH
      COMMON/B2/ DPG(128,32),BLOCK(128,32),IDPG(32),NBLOCK(32),IP(9),
     XID(9),NSUBI(9),LPERM(9),II(10),PSTOP(32),WT(32),PSUMX(32),PSUM(32)
      COMMON/REST/PROB(512), T1(512), PBLOCK(128,32), ISAVEP(9,33),
     X UTIL(512), ULIST(128), IULIST(128), IDDCG(4,32)
      INTEGER POWERS, T1, DPG, BLOCK
      COMMON/LINX/ XNOUT(5),HOLOUT(9),FAC(14,9)
      LOGICAL UTSWCH,ONLY1
      DIMENSION K2CYCL(9), KPERM(9)
      DIMENSION   MASK0(9)    ,MASK1(9)     ,KKSAVE(9)
      EQUIVALENCE(MASK0,POWERS(2))
      DATA (MASK1(I),I=1,9)/ 07777777776,07777777775,07777777773,
     X  07777777767,07777777757,07777777737,07777777677,07777777577,
     X  07777777377 /
      EQUIVALENCE (X,IX),(IS,SI),(JS,SJ)
      DIMENSION IALIAS(128),SUMALF(9,33),SUMVAL(33,33)
C
C*********************************************************************
C
C       REMACH SECTION 1
C
      ONLY1= .FALSE.
      NDPG1= NDPG+1
      GO TO 4
      ENTRY ONCE(*)
      NDPG1=1
      ONLY1=.TRUE.
```

```
    4 DO 10  LL=1,NFAC
      KKS= ISAVEP(LL,1)
      KPERM(KKS)=LL
   10 KKSAVE(KKS)=LL
C
C*************************************************************************
C
      DO 2500 L=1,NDPG1
C
C*************************************************************************
C
C       REMACH SECTION 2
C
      IF(ONLY1)  GO TO 25
      NOUT= L-1
      IF(NOUT.EQ.0)  WRITE(IUNOUT,5030)                                      23
      IF(NOUT.NE.0)  WRITE(IUNOUT,5035)NOUT,(IDDCG(K,NOUT),K=1,4)            25
   25 WRITE(IUNOUT,4090)                                                     31
      DO 100  LL=1,NFAC
      KKS        = ISAVEP(LL,L)
      WRITE(IUNOUT,4095)LL, FAC(1,LL),ALPHA(KKS)                             35
      SUMALF(LL,L)= ALPHA(KKS)
  100 CONTINUE
C
C*************************************************************************
C
C       REMACH SECTION 3
C
      DO 120  LLL=1,NFAC
      K2CYCL(LLL)= KPERM(LLL)
      DO 115  K= LLL,NFAC
      IF(KPERM(K)-LLL) 115,112,115
  112 KPERM(K)= KPERM(LLL)
      GO TO 120
  115 CONTINUE
  120 CONTINUE
C
C*************************************************************************
C
C       REMACH SECTION 4
C
      NF1=NFAC-1
      DO 158 LLL=1,NF1
      I= NFAC-LLL
      J= K2CYCL(I)
      KK= J-I
      IF(KK) 158,158,130
  130 MOI= MASKO(I)
      MOJ= MASKO(J)
      M1I= MASK1(I)
      M1J= MASK1(J)
C
C* * * * * * * * * * * * * * * * * *
C
      DO 157 L4= 1,NDPG
      I4=IDPG(L4)
      IF(I4) 152,152,148
  148 DO 150  L5=1,I4
      IX= DPG(L5,L4)
      SI= AND(MOI,X)
      SJ= AND(MOJ,X)
      X= AND(AND(M1I,X),M1J)
      IS=IALS(KK,SI)                                                         85
      JS=IARS(KK,SJ)                                                         87
      DPG(L5,L4)=IX+IS+JS
  150 CONTINUE
C
```

```
C* * * * * * * * * * * * * * * * * * * * * * * * * * *
  152 I4 = NBLOCK(L4)
      IF(I4) 157,157,155
  155 DO 156 L5=1,I4
      IX=BLOCK(L5,L4)
      SI= AND(MOI,X)
      SJ= AND(MOJ,X)
      X= AND(AND(M1I,X),M1J)
      IS=IALS(KK,SI)                                                    101
      JS=IARS(KK,SJ)                                                    103
      BLOCK(L5,L4)=IX+IS+JS
  156 CONTINUE
  157 CONTINUE
  158 CONTINUE
C
C*********************************************************************
C
C      REMACH SECTION 5
C
      DO 1000 I=1,NDPG
      WRITE(IUNOUT,5020) I                                              113
      WRITE(IUNOUT,5022) (IDDCG(K,I),K=1,4)                             114
      IRUN=IRUN+1
      PSUMX(I)=0.0
      NB=NBLOCK(I)
      IDI = IDPG(I)
      IDI1=IDI+1
C
C*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
C
C
      IF(NB) 7010,228,166
  166 DO 226  K=1,NB
      KI= BLOCK(K,I)
      KI1= KI+1
      T1(KI1)= IRUN
      IA=0
      IF(IDI) 7010,167,171
C
C--------------------------------------------------------------------
C
  167 IOUT= KI1
      U= UTIL(KI1) * (1.0-PBLOCK(K,I))
      GO TO 220
C
C--------------------------------------------------------------------
C
  171 IF(PBLOCK(K,I)-1.0)176,173,7010
  173 DO 174  IK= 1,IDI
      JJ= IEXOR(KI,DPG(IK,I))+1                                         148
      T1(JJ)= IRUN
  174 CONTINUE
      IOUT= KI1
      U=0.0
      GO TO 220
C
C--------------------------------------------------------------------
C
  176 PS= PROB(KI1)
      PR= PS
      IULIST(1)= KI1
      ULIST(1)= UTIL(KI1)
      ISTAR=0
      IA=0
      IF(PS) 178,178,179
```

```
  178 ISTAR=1
      ULIST= -UTIL(KI1)
      IALIAS(1)= KI1
      PR=1.0
  179 IMAX= KI1
C
C-----------------------------------------------------------------------
C
      IF(UTSWCH) GO TO 200
      DO 187  KK=1,IDI
      JJ=IEXOR(KI,DPG(KK,I))+1                                    172
      T1(JJ)= IRUN
      IF(PROB(JJ)) 7010,184,182
  182 PR= PR*PROB(JJ)
      IF(PROB(JJ)-PS) 186,183,187
  183 IA=IA+1
      IALIAS(IA)=JJ
      GO TO 187
  184 ISTAR= ISTAR+1
      IALIAS(ISTAR)=JJ
  186 PS= PROB(JJ)
      IMAX= JJ
      IA=0
  187 CONTINUE
C
C-----------------------------------------------------------------------
C
      IF(ISTAR-1) 190,192,189
  189 U=0.0
      IA = ISTAR -1
      IOUT= IMAX
       GO TO 220
  190 PR= PR/PROB(IMAX)
  192 U=PR*UTIL(IMAX)*(1.0-PBLOCK(K,I))
      IOUT=IMAX
      GO TO 220
C
C-----------------------------------------------------------------------
C
  200 DO 206 KK=2,IDI1
      JJ=IEXOR(KI,DPG(KK-1,I))+1                                  207
      T1(JJ)= IRUN
      IULIST(KK)= JJ
      IF(PROB(JJ)) 7010,204,202
  202 PR= PR * PROB(JJ)
      ULIST(KK)= UTIL(JJ)
      GO TO 206
  204 ISTAR= ISTAR+1
      IALIAS(ISTAR)=JJ
      IMAX = JJ
      ULIST(KK)= -UTIL(JJ)
  206 CONTINUE
C
C-----------------------------------------------------------------------
C
      IF(ISTAR-1) 208,214,189
  208 DO 210 KK=1,IDI1
      JJ= IULIST(KK)
      ULIST(KK)= ULIST(KK)*PR/PROB(JJ)
  210 CONTINUE
      UMAX=0.0
      IOUT=IULIST(1)
      DO 213 KK=1,IDI1
      IF(ULIST(KK) - UMAX) 213,212,211
  211 UMAX= ULIST(KK)
      IA=0
      IOUT=IULIST(KK)
      GO TO 213
```

```
  212 IA=IA+1
      IALIAS(IA)=IULIST(KK)
  213 CONTINUE
      U=UMAX*(1.0-PBLOCK(K,I))
      GO TO 220
C
C-----------------------------------------------------------------
C
  214 DO 216 KK=1,IDI1
      KS= KK
      TST= AND(ULIST(KK),NEG)
      TST= OR(TST,MASK)
      IF(TST) 218,218,216
  216 CONTINUE
  218 IOUT=IULIST(KS)
      U=-ULIST(KS)*PR*(1.0-PBLOCK(K,I))
C
C-----------------------------------------------------------------
C
  220 PSUMX(I)=PSUMX(I)+U
      IOUT = IOUT-1
      CALL LINE(IOUT,U)                                                 272
      IF(IA) 226,226,222
  222 DO 223 IIA= 1,IA
  223 IALIAS(IIA)=IALIAS(IIA)-1
      CALL ALINE(IA,IALIAS)                                             282
  226 CONTINUE
C
C*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
C
  228 IF(IDI) 7010,229,245
C* * * * * * * * * * * * * * * * * *
  229 DO 240 K=1,NN
      IF(TI(K)-IRUN) 230,240,230
  230 U=UTIL(K)
  239 IOUT= K-1
      PSUMX(I)=PSUMX(I)+U
      CALL LINE(IOUT,U)                                                  98
  240 CONTINUE
      GO TO 1000
C
C-----------------------------------------------------------------
C
  245 DO 700  K=1,NN
      IF(TI(K)-IRUN) 246,700,246
  246 PS= PROB(K)
      KM1= K-1
      PR=PS
      ISTAR=0
      IA=0
      IULIST(1)= K
      ULIST(1)= UTIL(K)
      IF(PS) 248,248,250
  248 ISTAR=1
      IALIAS(1)= K
      ULIST(1)= -UTIL(K)
      PR=1.0
  250 IMAX=K
      IF(UTSWCH) GO TO 430
C
      DO 410  KK=1,IDI
      JJ=IEXOR(KM1,DPG(KK,I))+1                                         325
      TI(JJ)=IRUN
      IF(PROB(JJ)) 380,390,380
  380 PR= PR*PROB(JJ)
      IF(PROB(JJ)-PS) 400,385,410
```

```
  385 IA=IA+1
      IALIAS(IA)=JJ
      GO TO 410
  390 ISTAR=ISTAR+1
      IALIAS(ISTAR)=JJ
  400 PS=PROB(JJ)
      IMAX=JJ
      IA=0
  410 CONTINUE
      GO TO 505
C
C-----------------------------------------------------------------
C
  430 DO 460 KK=2,IDI1
      JJ=IEXOR(KM1,DPG(KK-1,I))+1                                    351
      T1(JJ)= IRUN
      IULIST(KK)= JJ
      IF(PROB(JJ)) 450,450,440
  440 PR= PR*PROB(JJ)
      ULIST(KK)= UTIL(JJ)
      GO TO 460
  450 ISTAR= ISTAR+1
      IALIAS(ISTAR)=JJ
      IMAX = JJ
      ULIST(KK)= -UTIL(JJ)
  460 CONTINUE
      GO TO 600
C
C-----------------------------------------------------------------
C
  505 IF(ISTAR-1)  520,530,510
  510 U=0.0
      IA=ISTAR-1
      IOUT=IMAX
      GO TO 690
  520 PR=PR/PROB(IMAX)
  530 U=PR*UTIL(IMAX)
      IOUT=IMAX
      GO TO 690
C
C-----------------------------------------------------------------
C
  600 IF(ISTAR-1)  620,660,510
  620 DO 630 KK=1,IDI1
      JJ= IULIST(KK)
      ULIST(KK)= ULIST(KK) * PR/PROB(JJ)
  630 CONTINUE
      UMAX=0.0
      IOUT=IULIST(1)
      DO 650 KK=1,IDI1
      IF(ULIST(KK)-UMAX) 650,640,635
  635 UMAX= ULIST(KK)
      IA=0
      IOUT=IULIST(KK)
      GO TO 650
  640 IA=IA+1
      IALIAS(IA)=IULIST(KK)
  650 CONTINUE
      U=UMAX
      GO TO 690
C
C-----------------------------------------------------------------
C
  660 DO 670 KK=1,IDI1
      KS= KK
      TST=AND(ULIST(KK),NEG)
      TST=OR(TST,MASK)
      IF(TST) 680,680,670
```

```
  670 CONTINUE
  680 IOUT=IULIST(KS)
      U=-ULIST(KS)*PR
C
C-----------------------------------------------------------------
C
  690 PSUMX(I)=PSUMX(I)+U
      IOUT=IOUT-1
      CALL LINE(IOUT,U)                                                426
      IF(IA) 700,700,695
  695 DO 696 IIA=1,IA
  696 IALIAS(IIA)=IALIAS(IIA)-1
      CALL ALINE(IA,IALIAS)                                            436
  700 CONTINUE
 1000 CONTINUE
C
C*****************************************************************
C
C       REMACH SECTION 6
C
      PBAYEX= 0.0
      DO 1020 I=1,NDPG
      PSUMX(I)= PSUMX(I)*WT(I)
      PBAYEX= PBAYEX + PSUMX(I)*PSTOP(I)
      SUMVAL(I+1,L)= PSUMX(I)
 1020 CONTINUE
      SUMVAL(1,L)= PBAYEX
      WRITE(IUNOUT,5000) PBAYEX                                        453
      WRITE(IUNOUT,5005) (I,PSUMX(I),I=1,NDPG)
C
C*****************************************************************
C
C       REMACH SECTION 7A
C                                                                      454
      IF(L-NDPG1) 1050,2700,2700
 1050 DO 1070  LL=1,NFAC
      J=ISAVEP(LL,L+1)
      KPERM(J)=LL
 1070 CONTINUE
C
C*****************************************************************
C
C       REMACH SECTION 7B
      DO 2000  LL=1,NFAC
      DO 1080  L4=1,NFAC
      IF(KKSAVE(L4)-LL) 1080,1075,1080
 1075 K2CYCL(LL)= KPERM(L4)
      GO TO 2000
 1080 CONTINUE
 2000 CONTINUE
C
C*****************************************************************
C
C       REMACH SECTION 7C
      DO 2010  LL=1,NFAC
      KKSAVE(LL)=KPERM(LL)
      KPERM(LL)= K2CYCL(LL)
 2010 CONTINUE
C
C*****************************************************************
C
 2500 CONTINUE
 2700 IF(ONLY1) RETURN1
      CALL RECT(NDPG1,SUMALF,SUMVAL,NFAC)                              496
      RETURN
C
C*****************************************************************
```

```
 5000 FORMAT(52HL FOR THE ABOVE PERMUTATION THE EXPECTED UTILITY IS
     X  G14.5)
 5005 FORMAT(53HK THE EXPECTED UTILITIES AT THE STOPPING POINTS ARE.. /
     X  (28H  DEFINING PARAMETER GROUP  I3,3X,G14.5))
 5010 FORMAT(16H ERROR IN OUTPUT)
 5020 FORMAT(30HKDEFINING PARAMETER GROUP NO.  I3)
 5022 FORMAT (1H  4A6/60(2H *))
 5025 FORMAT(65(2H -))
 5030 FORMAT(37H1 THIS MATCHING IS THE BAYES MATCHING  )
 5035 FORMAT(52H1 THIS MATCHING MAXIMIZES THE EXPECTED VALUE AT THE I2,
     X 16H STOPPING POINT  4A6 )
 4095 FORMAT(1H I6,1X,A6,19X,A1)
 4090 FORMAT(38HKVARIABLE           SHOULD BE CALLED    )
C*****************************************************************************
C
C     ERROR MESSAGE
 7010 WRITE(IUNOUT,8010)                                                    498
 8010 FORMAT(52H PROGRAM ERROR--PREVIOUSLY GOOD DATA HAS BECOME BAD    )
      STOP
      END
```

APPENDIX C

PROGRAM SYMBOLS

This appendix presents a listing of the major program variables used in NAMER. Dimensioned variables have their dimensions specified.

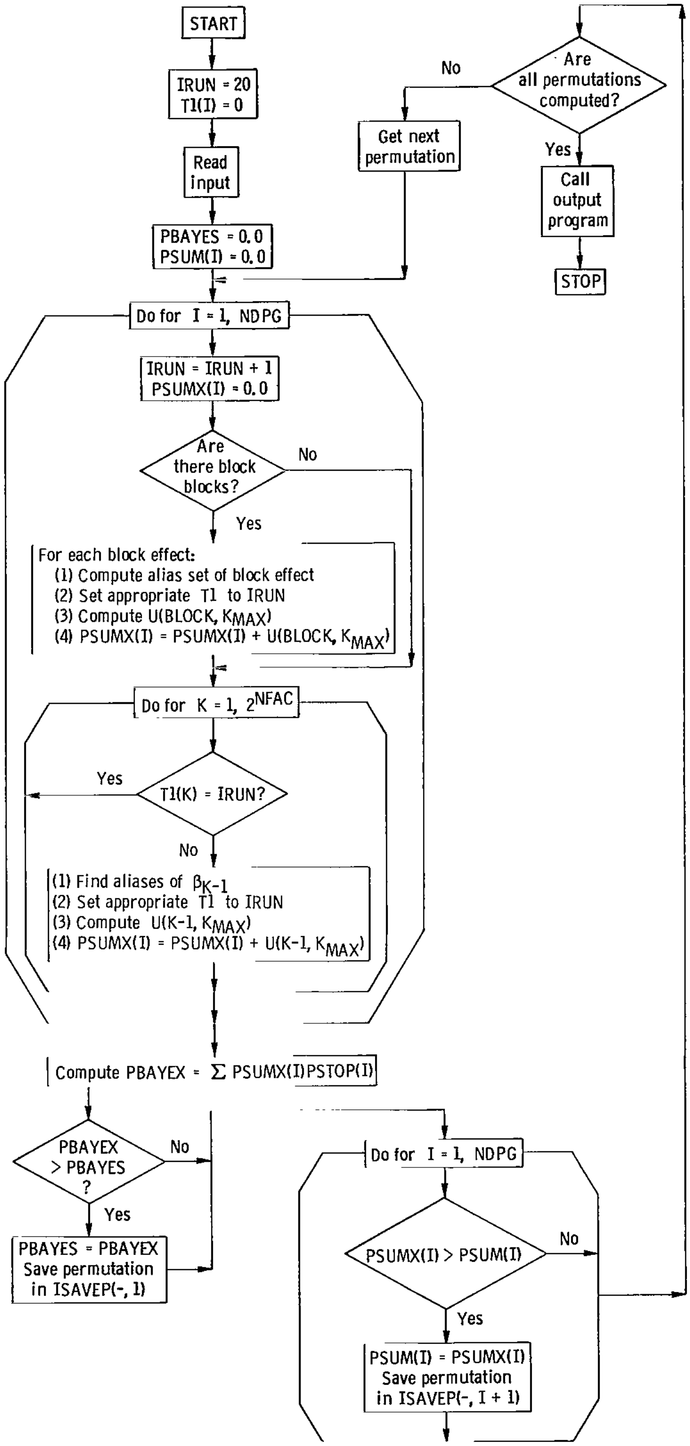
ALPHA(10)	First nine letters of the alphabet (excluding I) and a blank; used for output of optimal matchings.
BLOCK(128, 32)	Standard-order subscripts of representative members of alias sets confounded with blocks.
DPG(128, 32)	Standard-order subscripts of parameters in d.p.g.'s.
DUMP1, DUMP2	See section 5 of SPECIAL LEWIS RESEARCH CENTER ROUTINES.
FAC(14, 9)	Hollerith identification of factors.
HOLOUT(9)	Temporary storage for Hollerith output; used in LINER and initialized in NAMER.
IALIAS(128)	Standard-order number subscripts of parameters in an alias set, besides IOUT, which also maximize expected utility of eq. (6) or eq. (7).
IALPHA(9)	The integers 1 to 9; these are permuted by PERMUT and used to indicate optimal matchings for output.
ID(9)	Indicator vector used in PERMUT.
IDDCG(4, 32)	Hollerith identification of stopping points.
IDENT(14)	Hollerith identification of current problem.
IDPG(32)	Number of elements in each d.p.g. (not including identity).
II(10)	Subscripts of beginning locations of classes of factors.
IMAX, IOUT	Subscript of parameter which maximizes eq. (6) or eq. (7).
IP(9)	Indicator vector used in PERMUT.
IRUN	Running counter used as indicator of parameters evaluated.
ISAVEP(9, 33)	Optimal matchings.
ISTAR	Counter for number of parameters in an alias set with prior probability 1.0 of being nonzero.
ITYPRN	Type of run; see section 3 of INPUT DESCRIPTION.

IULIST(128)	Vector of subscripts of parameters in an alias set.
IUNIN	Variable input unit designation.
IUNOUT	Variable output unit designation.
IUTILF	Choice of function; see section 7 of INPUT DESCRIPTION.
KKSAVE(9)	Saves permutation vector required to achieve optimal matching.
KPERM(9)	Permutation vector.
K2CYCL(9)	Transposition vector for permutation in KPERM.
LD1, LD2	Delimiters for DUMP1 and DUMP2 vectors; see section (5) of SPECIAL LEWIS RESEARCH CENTER ROUTINES.
LPERM(9)	Logical variables controlling calls to PERMUT.
NBLOCK(32)	Number of alias sets confounded with blocks for each d. p. g.
NCLASS	Number of classes of factors; see section 3 of INPUT DESCRIPTION.
NDPG	Number of d. p. g. 's.
NFAC	Number of factors.
NN	2^{NFAC}
NSUBI(9)	Number of factors per class.
ONLY1	Logical variable set to .TRUE. if only a single specified matching is to be evaluated.
P	Temporary storage of input prior probability.
PBAYES	Overall expected utility of best matching evaluated so far.
PBAYEX	Overall expected utility of current matching.
PBLOCK(128, 32)	Prior probabilities of block effects being nonzero.
PR	Temporary storage used for calculation of $\prod(1 - p_i)$.
PROB(512)	Vector of $(1 - p_i)$.
PSTOP(32)	Probabilities of stopping exactly at each stopping point.
PSUM(32)	Expected utility at each stopping point of the best matching for that stopping point found so far.
PSUMX(32)	Expected utility at each stopping point of the current matching.
SUMALF(9, 33)	Saves optimal orderings for output of summary table.

SUMVAL(33, 33)	Saves expected utilities of various optimal orderings for output of summary table.
TMAX	Maximum total running time permitted.
TMAXX	Maximum running time for current case.
T1(512)	Indicator array used in finding all distinct alias sets.
UCOEF	Constant used to define utility function 5; see section 7 of INPUT DESCRIPTION.
ULIST(128)	Vector of utilities corresponding to choices of parameters indicated in vector IULIST.
UMAX	Maximum of ULIST.
UT	Temporary storage used in input of utilities.
UTIL(512)	Vector of u_i.
UTSWCH	Logical variable used to indicate whether utility function 1 or 2 or utility function 3, 4, or 5 is being used.
WT(32)	Weighting values for the stopping points; see section 7 of INPUT DESCRIPTION.
XNOUT(5)	Temporary storage used in LINER for output of numerical identification of parameters chosen to be estimated.

APPENDIX D

PROGRAM GENERAL FLOW DIAGRAM



REFERENCES

1. Sidik, S. M.; and Holms, A. G.: Optimal Design Procedures for Two-Level Fractional Factorial Experiments Given Prior Information about Parameters. NASA TN D-6527, 1971.
2. Holms, Arthur G.: Designs of Experiments as Telescoping Sequences of Blocks for Optimum Seeking (As Intended for Alloy Development). NASA TN D-4100, 1967.
3. Holms, Arthur G.; and Sidik, Steven M.: Design of Experiments as "Doubly Telescoping" Sequences of Blocks with Application to a Nuclear Reactor Experiment. Technometrics, vol. 13, no. 3, Aug. 1971, pp. 559-574.
4. Ord-Smith, R. J.: Generation of Permutation Sequences: Part 1. Computer J., vol. 13, no. 2, May 1970, pp. 152-155.
5. Trotter, H. F.: Algorithm 115 - PERM. Comm. ACM, vol. 5, no. 8, Aug. 1962, pp. 434-435.
6. Holms, Arthur G.; and Sidik, Steven M.: Design of Experiments as "Doubly Telescoping" Sequences of Blocks with Application to a Nuclear Reactor Experiment. NASA TN D-5369, 1969.

TABLE I. - SAMPLE INPUT FOR PROBLEM

DESCRIBED IN APPENDIX A

1 2 3 4 5	6	7 8 9 10 11 12	13 14 15 16 17 18	19 20 21 22 23 24
NAM	E	R SAMP	LE PRO	BLEM
	2			
	1			
	5			
TEMP		SOURCE	TEMPE	RATURE
PRESS		SOURCE	PRESS	URE
TIME		TIME D	URATIO	N
VEL		SOURCE	VELOC	ITY
ANGLE		SOURCE	ANGLE	
	1			
1	5	2		
		0	1.0	
		1	.80	
		2		
		12		
		3		
		13		
		23		
		123		
		4	1.0	
		14	.50	
		34	.50	
		134	.40	
		5	1.0	
		15	.40	
1 2 3 4 5	6	7 8 9 10 11 12	13 14 15 16 17 18	19 20 21 22 23 24

1 2 3 4 5	6	7 8 9 10 11 12	13 14 15 16 17 18	19 20 21 22 23 24
		35	.30	
	3			
1/4 R	E	P--ROW	1	
	2	.30	.125	
ABC				
CDE				
	2			
AD		.50		
		1.0		
1/2 R	E	P--ROW	S 1, 2	
	1	.40	.0625	
ABDE				
	3			
AD		.50		
		1.0		
ABC		1.0		
FULL	-	- ALL	ROWS	
	0	.30	.03125	
	5			
AD		.50		
		1.0		
ABC		1.0		
ABDE		1.0		
CDE		1.0		
ENDAL	L			
1 2 3 4 5	6	7 8 9 10 11 12	13 14 15 16 17 18	19 20 21 22 23 24

TABLE II. - ESTIMATE OF TIME REQUIRED BY PROGRAM - BASED ON RUNNING OF SAMPLE PROBLEMS

Utility function		Number of independent variables				
		5	6	7	8	9
2	Total time to evaluate all permutations, min	0.02	0.23	3.90	60.32	-----
	Time required to print out results, min	0.04	0.07	0.19	0.44	-----
	Number of d.p.g.'s	4	4	5	5	-----
	Time to evaluate all permutations divided by number of d.p.g.'s, min	0.005	0.058	0.780	12.06	[a]151.0
3	Total time to evaluate all permutations, min	0.03	0.27	4.64	75.67	-----
	Time required to print out results, min	0.04	0.08	0.22	0.47	-----
	Number of d.p.g.'s	4	4	5	5	-----
	Time to evaluate all permutations divided by number of d.p.g.'s	0.008	0.068	0.928	15.13	[a]224.0

[a]Estimated from fig. 2.

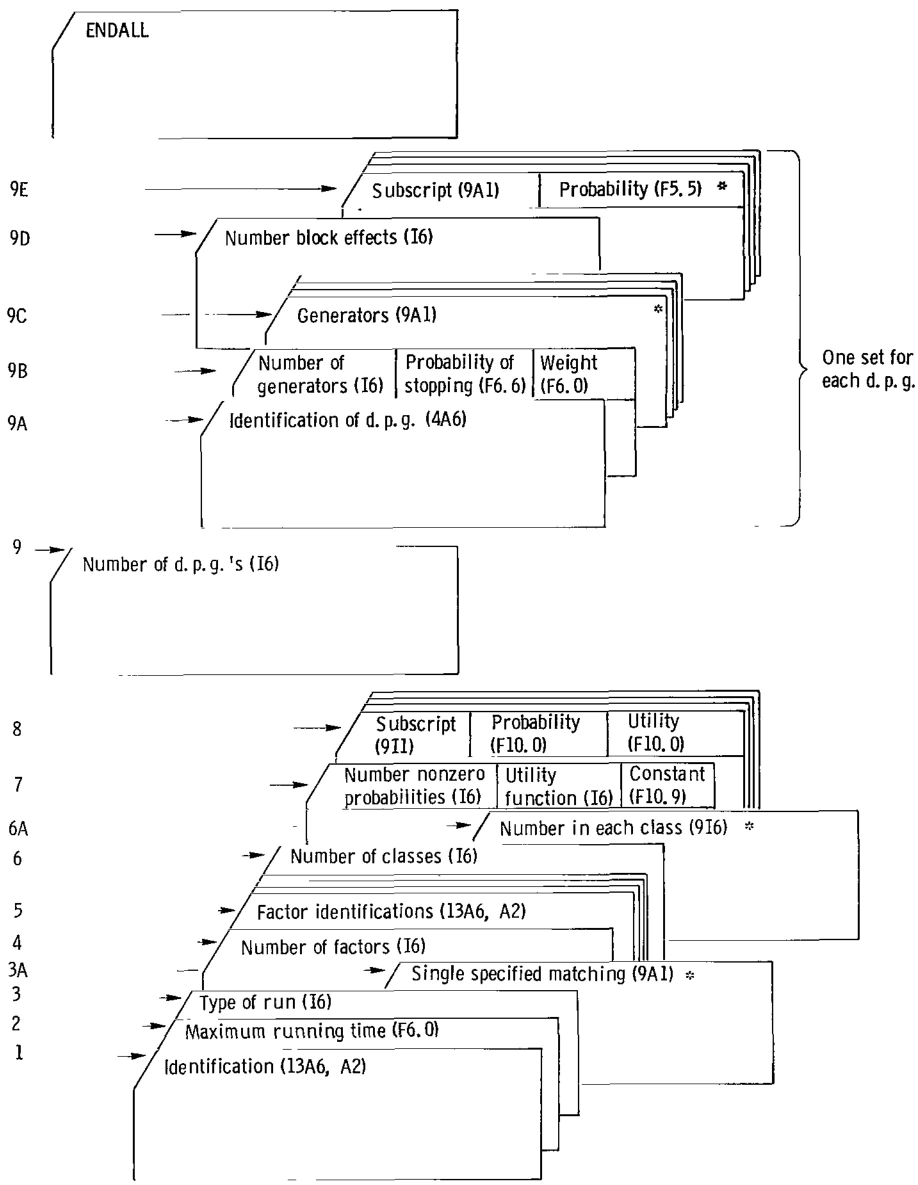


Figure 1. - Pictorial representation of data card arrangement. (Asterisk denotes cards which are optional. Presence depends on input information contained on earlier cards.)

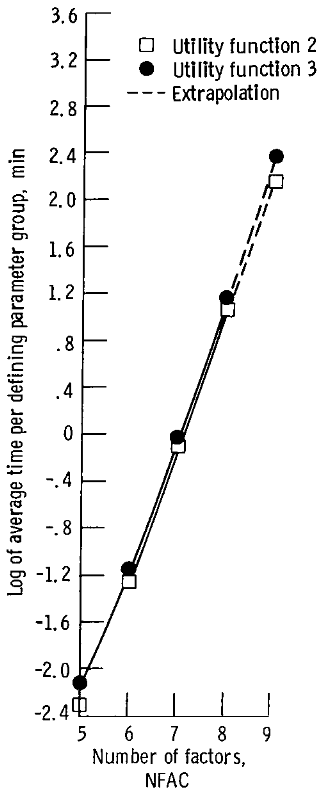


Figure 2. - Logarithm of average time required per defining parameter group as function of number of factors and choice of utility function.

NASA-Langley, 1972 —— 8 E-6329

NATIONAL AERONAUTICS AND SPACE ADMISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

U.S.MAIL

012 001 C1 U 08 720128 S00903DS
DEPT OF THE AIR FORCE
AF WEAPONS LAB (AFSC)
TECH LIBRARY/WLOL/
ATTN: E LOU BOWMAN, CHIEF
KIRTLAND AFB NM 87117

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546