A RAND NOTE

THE FINITE SELECTION MODEL: DESCRIPTION AND USER'S GUIDE

Martin Seda, Joan Keesey

June 1980

N-1183-HEW

Prepared For

The U.S. Department of Health, Education, and Welfare

Rand
SANTA MONICA, CA. 90406

The research reported herein was performed pursuant to Grant No. 016B-7901-P2021 from the U.S. Department of Health, Education, and Welfare, Washington, D. C. The opinions and conclusions expressed herein are solely those of the author, and should not be construed as representing the opinions or policy of any agency of the United States Government.

Published by The Rand Corporation

A RAND NOTE

THE FINITE SELECTION MODEL: DESCRIPTION AND USER'S GUIDE

Martin Seda, Joan Keesey

June 1980

N-1183-HEW

Prepared For

The U.S. Department of Health, Education, and Welfare

Rand
SANTA MONICA, CA. 90406

PREFACE

This Note describes the design, application, and operational use of the Finite Selection Model (FSM) computer program. The program was developed specifically for the Rand Health Insurance Study (HIS), which is evaluating the effects of alternative insurance plans upon demand for medical services and general health status. A significant portion of the study involves collecting three to five years of longitudinal data on more than two thousand families after assigning them to one of several predesignated health insurance plans. The FSM was used to determine which families should be assigned to which plans. Numerous other statistical design applications of the FSM are also possible.

This Note was written primarily for analysts and programmers intending to use the FSM who need documentation of program capabilities as well as a user's guide. Theoretical proofs and discussions of the statistical validity of the underlying model are left to other reports.

The Health Insurance Study is supported by a grant from the U.S. Department of Health, Education, and Welfare.

SUMMARY

This Note describes the design, application, and operational use of the Finite Selection Model (FSM). The FSM is a computer program designed to assign subjects from a finite pool of eligible sets to user specified treatment groups while generating statistical measures, such as the cost effectiveness of the resulting assignments to each group.

This model was developed specifically for the Health Insurance Study (HIS) - a study designed to evaluate the effects of alternative insurance plans upon demand for medical services and health status. The FSM was used to select and assign eligible families to predesignated insurance plans. Random sampling of the free pool of eligible subjects could have been used; however, the availability of pre-experimental data made possible the development of a statistical model which would improve on random selection and provide measures that show the amount of improvement.

The derivation and proof of the FSM are not within the scope of this Note. However, the statistical design concepts and procedural algorithms are described briefly in Section IV. This Note was designed primarily as a user's guide for programmers and analysts who intend to use the FSM for other applications and as documentation of the model's capabilities.

ACKNOWLEDGMENTS

The concepts underlying the present computer program are the product of research by Dr. Carl Morris. His knowledge of the experimental design problems faced by the Health Insurance Study and his familiarity with the inadequacies of the existing tools used in sample selection prompted him to develop the Finite Selection Model. He guided the design and development of the FSM program from its initial design to the form documented in this Note.

CONTENTS

FIGURES

TABLES

I. INTRODUCTION

In the Health Insurance Study, the selection of subjects for enrollment on health insurance plans is a complex process. Specifically, the process includes:

1. Administration of thousands of screening interviews to determine whether subjects are eligible for selection based on predetermined criteria (income, age, military status, etc.) and collection of other pre-experimental data.
2. Compilation of a "free pool" of eligible subjects suitable for selection.
3. Selection of subjects from the free pool for specific health insurance plans.
4. Enrollment of selected subjects who agree to be assigned to the specified plan.
5. Deletion of selected subjects who refuse to be enrolled in specified plans from the free pool. If the desired sample size has not been attained, the procedure is repeated from step 2.

Random sampling of the free pool could have been used for selection of subjects in step 3. However, the availability of pre-experimental data (step 1) made it possible to develop a statistical model that would improve on random selection and provide measures of precision and balance that show the amount of improvement.

The Finite Selection Model was built on foundations of earlier work by Conlisk and Watts[1]. Their allocation model had been used to determine optimal sample sizes for other public policy experiments before the HIS. The Conlisk-Watts allocation model assumes an infinite population stratified by a limited number of variables. It generates optimal stratified samples once the experimental constraints are defined and associated costs are specified. It was used in the HIS to help determine good sample sizes for insurance plans.

The Conlisk-Watts model, however, is not adequate for assigning families to health insurance plans in the presence of a substantial collection of pre-experimental data on every family. Stratified samples have the disadvantages that

1. More families than those available may be selected from a given stratum.
2. The number of pre-experimental variables used must be severely limited.
3. Families with dissimilar values of a continuous variable, such as income, must be grouped in the same stratum.

The Finite Selection Model circumvents these difficulties by making selections for health insurance plans from the finite population of subjects known to be available for assignment. Pre-experimental variables are permitted to be continuous, and the number allowed are essentially a function of the user's budget and computer memory limitations (as many as 24 in the case of HIS).

II. CAPABILITIES OF THE FSM

The Finite Selection Model is a computer program designed to assign subjects from predefined free pools to user-specified treatment groups while generating statistics indicating the cost effectiveness of the assignments to each group. The choice of any particular assignment depends primarily on user-defined objective functions and input weights for the independent variables of the objective functions.

Several objective functions can be used simultaneously for sample selection, each consisting of a different combination of independent variables. The models are then weighted by their relative importance to the problem. These weights govern the variables' contribution to the selection marginal-cost-effectiveness calculations.

"Treatment group" is a term used widely in experimental design and statistics to refer to the set of experimental subjects that receive a certain treatment (program, plan, etc.). Since the HIS enrolled families in insurance plans, the term "plan" is used occasionally to denote the same concept. Treatment group, group, and plan have synonymous meanings in this document.

In the terminology of the FSM, each subject or individual for analysis is a "point." While this defines the unit of analysis, the unit of selection may be a group or "set" of points. In the HIS, for example, persons were points, but all persons in a household (the "set") had to be assigned to the same treatment. Points are the smallest units of concern to the model, and values of the defined independent variables used by the objective functions are stored for each point. The counting unit for each set is the "set increment." This can be any positive number, usually the cost of the set, or the number of points in the set. In the HIS, a basic constraint was the number of families that could be assigned to each plan, so the number of families in the household (set) was taken to be the set increment. The FSM ceases to make assignments to a treatment group after the total number of assigned set increments reaches the user-defined limit.

Selection of a set for a group involves computing the marginal cost effectiveness (CE) of sets yet unselected and choosing that one with the best CE. The CE is dependent on the characteristics of the set's component points. The CE of the new set is combined with the CE's of the sets already assigned to the group to provide a measure of the relative value of the assignment. This process of adding sets onto groups is defined as a "build-up." If this process is reversed, it is called "build-down."

FSM provides the user with a capability for grouping sets into any number of strata. Build-up and build-down can then be based on strata rather than on the complete population with constraints on the number of sets assigned from each stratum to each group.

Finally, the user can indicate the "selection strategy" to be used in the selection run. This tells the model in which order groups make choices from the free pool of sets. Selection strategies vary from random selection to the use of a predetermined user-specified selection order matrix. These strategies will be discussed in detail in Sections IV and V.

III. USE OF THE FSM ON THE HIS

The HIS selection application provided a very complex test of the Finite Selection Model. Some of the design features implemented were especially helpful in the context of the HIS enrollment process. Since the HIS problem exercises so many of the FSM abilities that may be helpful to users, we shall review the problem more closely.

The FSM used in this case was set up to run on person data (points), accumulate families (set increments), and assign households (sets), which may consist of more than a single family. Additionally, in an effort to preserve balance across plans, households were stratified by multiple households and family size. The final strata used were as shown in Table 1.

Table 1

HIS SAMPLE STRATIFICATION

Stratum Number	Definition
1	Single Family Household - 1 person
2	Single Family Household - 2 persons
3	Single Family Household - 3 persons
4	Single Family Household - 4 persons
5	Single Family Household - 5 or more persons
6	Double family households
7	Triple family households

Typically, fourteen primary treatment groups were to be constructed in each experimental site (city). Each group would receive a different kind of insurance plan. A linear regression equation was specified for the individuals in each group, and the FSM was used to choose households for each plan so that the characteristics of the individuals selected permitted sufficiently precise estimation of the regression coefficients.

The plans were built up to the final family sample sizes desired by allowing them to take turns in selecting households from strata in a random order determined by a selection order matrix (S.O.M.). The S.O.M. specifies at each selection step which plan is to choose from which stratum. The construction of a S.O.M. will be discussed later.

The HIS enrollment was complicated further by the pre-experimental screening data, which became available in stages for each site. Because of time pressures, selections had to be made at each stage in the absence of future data. These partial sample selections had to be processed before enrollment at a site was completed. After the initial partial selection, the number of households assigned to specific plans by previous partial selections had to be determined. The selection process was then restarted at that point to continue the sequential build-up process.

Since a relatively small number of households were being allocated to a relatively large number of plans during the early partial selections at each site, the plan sizes at those stages were not large enough to permit all the coefficients of the independent variables to be estimated; the regression matrix was not invertible. This happened because some objective functions contained as many as 24 independent pre-experimental variables.

The solution to this problem was to divide each partial selection into several "splits." The first split was made by using the complete objective function with its full complement of independent variables and dividing the available sets into about three groups. Then, by using objective functions dependent on fewer variables, these groups were subdivided further. Figure 1 illustrates this process. It required three such splits.

As a standard check of results achieved at every split of each partial selection, FSM was instructed to select a stratified random sample for purposes of comparison. The difference between optimal and random selection must necessarily diminish as the selections progress from the initial partial selection/first split to completion, because random sampling is a nearly optimal procedure in large samples.

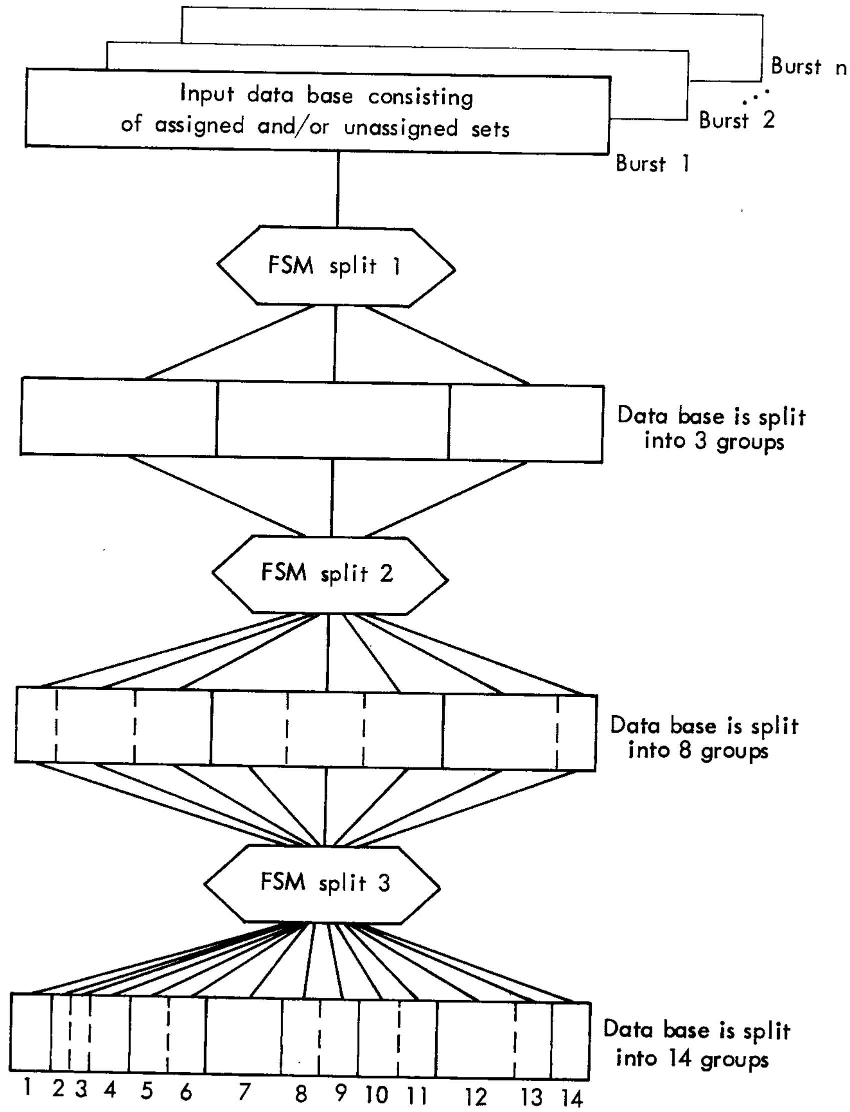

Fig. 1 — HIS burst driven selection process

IV. MATHEMATICAL DESCRIPTION OF THE FSM

The major objective of the Finite Selection Model is to choose samples from a finite pool of eligible sets for a specified number of groups. At each selection, the FSM attempts to add the marginally most cost effective set to a specific group. One set at a time is assigned from the free pool to a group until all groups have been built up to their desired sample sizes. This section describes the functions that are performed by the model to achieve such a solution.

The complete derivation and proof of the FSM concepts is not within the scope of this Note. The reader who is interested in the statistical foundations of the FSM should see [2], for example. A brief description of the FSM calculations is given below.

FSM CALCULATIONS

The FSM calculations depend upon the availability of an [X] matrix that has the following structure:

$$[X] = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,k} \\ X_{2,1} & & & \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ X_{n,1} & & \dots & X_{n,k} \end{bmatrix}$$

This matrix is interpreted by the model as containing n points of measurements or independent variables for each point.

It is assumed that the regression equation is of the form

$$[Y] = [X]\ [B] + \text{error},$$

where [B] is a columnar matrix of k coefficients (betas) corresponding to the independent variables represented in [X].

Normally, the researcher wishes to combine these betas linearly to formulate a suitable objective function. Thus, FSM requires a k by p "policy" matrix [p] of the following form to compute the linear combination:

$$[P] = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,k} \\ P_{2,1} & & & \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ P_{p,1} & & & P_{p,k} \end{bmatrix}$$

where k is the number of betas and p is the number of policies. This policy matrix provides a means of defining a new matrix of "alpha" coefficients, [A], representing the p linear combinations of the original beta's, so that

$$[A] = [P]\ [B].$$

In FSM terminology, the alphas are called "policies." The program requires that the first k policies yield the betas themselves. Therefore, the first k by k portion of the [P] matrix must be the identity matrix [I].

Assuming that each point in the [X] matrix is a vector of the form

$$(1,\ X_1\ ,\ X_2\ ,\ X_3\ ),$$

then, the following are admissible [P]'s:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ & & & . \\ & & & \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & 6 & 0 & 0 \\ 5 & 5 & 5 & 5 \end{bmatrix}$$

The alphas become the component terms of user-defined arbitrary

model and objective functions. The user is allowed to define multiple models and objective functions. Each model corresponds to a different regression relation between the dependent variables and certain columns of [X]. The user then must provide an alpha weight matrix $[W]_m$ for each model defined. The totality of all such weight matrices is represented by a matrix [W] such that

$$[W] = \begin{bmatrix} W_1 \\ W_2 \\ \cdot \\ \cdot \\ \cdot \\ W_p \end{bmatrix}$$

where p is the number of policies, and a $[W]_m$ matrix exists for each model m.

These matrices then represent the basic inputs involved in the FSM selection process. The initial calculations required to produce the selection environment are discussed next.

FSM first calculates a [Q0] matrix which may be described as the k by k matrix specifiying an average point in the [X] matrix. Since points represent persons in the HIS, the [Q0] matrix describes the characteristics of an average person. The required computation is given by

$$[Q0]_m = ([X]'_m \; [X]_m \;)/(\text{NPTS-NBCF-1})$$

where m is the model used in the calculations, and

$[X]_m$ = those variates within the [X] matrix required by model m,

NPTS = the number of points (rows) in the matrix [X],

NBCF = the number of regression coefficients in the model.

The FSM program uses this matrix to insure invertibility of various matrices composed of submatrices of [X], as discussed below. The construction of the [Q0] matrix should be considered a consequence of the program design rather than a theoretical requirement.

For each model, several k by k [S] matrices are constructed. The first is the [S0] matrix and is the negative inverse of the previously computed [Q0]. Thus,

$$[S0]_m = -[Q0]_m^{-1} ,$$

where m indicates the appropriate model. This calculation is repeated separately for each group. A multiple, e, of the [Q0] matrix is added to each group's [S] matrix to insure that the inverse of Q0 exists. The factor e is the multiple of a user-specified S multiplier and the desired plan size. A detailed explanation of the S multiplier appears in Section V. Thus,

$$[S]_{gm} = -(([X]'_{gm} [X]_{gm}) + e_g [Q0]_m)^{-1}$$

where g is the indicated group, and

e = (S multiplier) (final size of group g),

$[X]_{gm}$ = submatrix of variates within the [X] matrix required by model m for those points included in group g.

The creation of [S] matrices for each group and model is necessary only when sets have been preassigned prior to build-up.

Finally, it is possible to derive the objective functions applying to each model used in the run. At each step, the FSM minimizes the weighted sum of the variances of the alpha coefficients involved in each model for every group. The estimated covariance matrix of the alpha coefficients is given by

$$\text{cov}\ ([A]_m) = -[P]'_m\ [SO]_m\ [P]_m$$

where $[P]_m$ is the policy matrix for model m. The diagonal terms in the covariance matrix are the variance terms. So, the estimated variance for each alpha a_{jm} is

$$\text{var}\ (a_{jm}) = -[p]'_{jm}\ [SO]_m\ [p]_{jm}$$

where a_{jm} = the j^{th} element of $[A]_m$, and

$[p]_{jm}$ = the j^{th} row of $[P]_m$.

Actually, var (a_{jm}) is the expected variance of the alpha coefficient and is denoted more correctly as E var(g_{jm}). This expectation is exact if [X] is randomly sampled from a multivariate normal population.

Finally, the proportional weights from the [W] matrix for each alpha are normalized by the variance of the alpha coefficients themselves, and a matrix [D] is calculated. Thus, each policy weight becomes

$$\bar{w}_{jm} = \frac{w_{jm}}{E\ \text{var}\ (\bar{a}_{jm})}$$

where w_{jm} = j^{th} policy weight assigned in matrix $[W]_m$,

$\bar{w}_{jm}$ = j^{th} normalized policy weight in matrix $[\bar{W}]_m$.

Now, using the newly computed weights, the matrix [D] is calculated for each model according to

$$[D]_m = [P]_m [\overline{W}]_m [P]'_m$$

This matrix is used in the build-up calculations described below and completes the initial selection calculations. It now is possible for the FSM to proceed with the actual sample selection.

THE ORDER OF SELECTION

The FSM makes its assignments one step at a time in sequence. At each step, one particular treatment group is chosen to be assigned an experimental subject from a designated stratum. The user must specify which treatment group is to be selected from which stratum for every step in the entire selection sequence by specifying a "selection order matrix" (S.O.M.). The choice of the S.O.M. can significantly affect the quality of the final sample assignments, so we urge the user to discuss this choice with someone familiar with the options and the potential consequences of a choice of options. The form of the externally specified S.O.M. is described in Section V and an example is given in Appendix B.

Since an externally generated S.O.M. can take a long time to construct, two algorithmic options are provided by the FSM software. They are

1. Global selection strategy.
2. Local selection strategy.

Both algorithms permit the group that has been assigned the smallest number of sets relative to its final number of choices to have the first new assignment. The strategies are distinguished from each other by the consideration they give to preassigned sets.

The global strategy selects the group that has the minimum ratio defined by

$$\frac{\text{(current group size)} + 1/2}{\text{(desired group size in set increments)}}$$

This ratio measures the relative completeness of a group's sample selection.

To illustrate the global selection order, assume that there are two groups, where group 1 chooses 6 sets and group 2 chooses 9. The selection order generated is the following:

2 1 2 1 2 2 1 2 1 2 2 1 2 1 2.

The local strategy selects the group that has the smallest value of the ratio

$$\frac{[(\text{current group size})-(\text{number preassigned})] + 1/2}{[(\text{desired group size})-(\text{number preassigned})]}$$

Again, this gives a measure of selection completeness for each group independent of the initial conditions existing at the start of the selection process.

The generated selection orders for the local selection strategy will be the same as the global strategy when no sets have been assigned to the groups involved. Assuming the two-group example illustrated above, the selection order under the local algorithm also will be

2 1 2 1 2 2 1 2 1 2 2 1 2 1 2

when the starting group sizes are zero.

Assuming that 6 sets are preassigned so that initially group 1 contains 3 sets and group 2 has 3 sets. Then the following patterns will be generated by the two strategies for the remaining 9 sets:

Global 2 2 1 2 2 1 2 1 2
Local 2 1 2 2 1 2 2 1 2

This example illustrates the basic difference between global and local algorithms. The global selection order restores balanced choice as early as possible if the sets are not preassigned to the treatment

groups in proportion to their ultimate size. The local selection order requires group 2 to catch up evenly throughout the selection order. In neither case is the selection order random; randomization can only be achieved by using an externally produced S.O.M.

The global and local options are also available when selections are to be made from designated strata. The strata are indicated by "Strata Commands," which will be defined and discussed in Section V.

At each step, after a treatment group has been designated as the next group to which a set will be assigned, the FSM will find the marginally optimal set for the group. The FSM may be instructed to select this set from all of those available, or a "search length" may be specified to indicate that only the specified number of sets are to be considered. The FSM then will choose the set that provides the most favorable ratio of reduction in weighted marginal variances of the alphas to the marginal cost of the new set, as follows.

Assume that group g has been designated as the next group that will be assigned a set and has a number of sets already assigned to it. Then, the most cost effective set to be added to group g will be the set that maximizes the function

$$CE([X_i]) = \frac{1}{C_i} \times \frac{([x]_i\ [S]_{gm}\ [D]_m\ [S]_{gm}\ [x]_i)}{(1 - [x]'_i\ [S]_{gm}\ [x]_i)}$$

where X_i is permitted to range over those points available for selection at the particular step, $[x]_i$ is the row of [X] associated with the i^{th} point, and C_i is the cost of the i^{th} point. This function is evaluted for each point of every set considered up to the specified search length. (Actually, the formulas described above and just below are only correct when each set has only one point. When several points are contained in the same set, both formulas take more complicated forms to reflect the combined contribution of all the points.)

Once a set has been selected for a group, the S matrices for each model for the group are recalculated by the formula

$$[S]_{gm} = [S]_{gm} + \frac{([S]_{gm}\ [x]_i\ [x]'_i\ [S]_{gm})}{(1 - [x]'_i\ [S]_{gm}\ [x]_i)}$$

Here, the left-hand value is the updated version of $[S]_{gm}$ on the right-hand side, to take account of the selection of $[X]_i$.

This procedure of selecting sets to be added to an indicated group (using a defined strategy) is reiterated until the desired sample size or number of set increments has been attained. Before terminating the procedure, the user may require FSM to calculate certain statistical measures to evaluate the selected samples, as follows.

Calculations are performed for each group showing the relative departures of each policy mean for the group from that of the associated policy mean for the total population (the "Z-test statistics"). Standard deviations of the policy variates for the plan are divided by the policy standard deviation for all sets to provide a relative measure of variability of policies within groups.

Characteristics of the design are indicated by the t by p matrix [H] of relative variances, where t indicates the total number of groups involved. The elements of the matrix represent the variances of the regression coefficients of the policies normalized by their expected values under random sampling from a multivariate normal population. The (g, j) element of the matrix for model m is given by

$$h_{gjm} = \frac{-(n_g - k - 1)([p]_{jm}\ [S]_{gm}\ [p]'_{jm})}{([p]_{jm}\ [L]_{gm}\ [p]'_{jm})}$$

where n_g = number of sets included in the sample for group g,

$[p]_{jm}$ = the j^{th} row of matrix [P] defining the j^{th} policy,

$[S]_{gm}$ = the S matrix calculated for group g,

$[L]_{gm}$ = expected value of the inverse covariance

matrix $-(n_g - k - 1) * [SO]_m$.

The expected value of any element of the [H] matrix under random sampling from a multivariate normal population is

$$E(h_{gj}) = 1$$

Therefore, for each plan and policy variate, a selection of associated sets is an improvement on the expected outcome of random sampling if the corresponding value in the [H] matrix is less than 1. If the [X] matrix is not sampled from a multivariate normal population, then, under random sampling it will usually happen that

$$E(h_{gj}) > 1$$

and the [H] matrix will provide only a rough guideline to the relative efficiency of a selection. In the case of an abnormal population, more exact comparisons have to be obtained by using the FSM to generate [H] matrices for random samples (see MAXIT command usage in Section V).

The elements of the [H] matrix are characterized by various means and standard deviation calculations for each row and column of the matrix. The derived matrices of statistics are shown below, and related spatially to the [H] matrix in Fig. 2 on page 19.

The elements of the $[HR]_m$ matrix represent the mean of the policy statistics across any group g. This matrix characterizes the objective function that is being minimized by FSM. The matrix [RR] represents the associated standard deviation of any group's policy statistics from the group mean. In both matrices, the policy weight matrix for the model $[W]_m$ is used to weight the corresponding elements in the $[H]_m$ matrix in calculating the mean and standard deviation. This matrix is normalized relative to the sum of the weights assigned to all the policies for the model m.

The [HC] matrix represents the mean of the policy statistics across all groups, and the [RC] matrix is the associated standard deviation of the $[H]_m$ elements making up those means. The statistics for each group are weighted equally in calculating these matrices.

The $[V]_m$ matrix describes the efficiency and balance of the complete selection across all groups for model m. The elements of this 2 by 2 matrix measure the following characteristics:

1. $v_{1,1}$ = mean of the average policy statistics across all groups weighted by the relative weight w of each policy j in the model m. Each group has the same weight. This statistic indicates the overall efficiency of the selection.
2. $v_{1,2}$ = the standard deviation of the $[HR]_m$ elements around the weighted mean $v_{1,1}$ using the w_{jm} weights.
3. $v_{2,1}$ = the standard deviation of the hr_g elements around $v_{1,1}$ This measures the balance of the objective functions of the groups.
4. $v_{2,2}$ = the weighted standard deviation of all the elements of the [H] matrix computed around the grand mean $v_{1,1}$.

This brief description is intended to clarify the next few sections, which cover the control inputs and outputs. An understanding of the concepts described here is necessary in adapting the FSM to a new application.

		Mean of Policies for Grps.	Std. Dev. Between Policies for Grps.
[H] Matrix with p Policies on t Groups	$\begin{bmatrix} h_{1,1} & \cdots & h_{1,p} \\ \vdots & & \\ h_{t,1} & \cdots & h_{t,p} \end{bmatrix}_m$	$\begin{bmatrix} hr_1 \\ \vdots \\ hr_t \end{bmatrix}_m$	$\begin{bmatrix} rr_1 \\ \vdots \\ rr_t \end{bmatrix}_m$
Policy Means Across Grps.	$\begin{bmatrix} hc_1 & \cdots & hc_p \end{bmatrix}_m$	$\begin{bmatrix} v_{1,1} & \cdots & v_{1,2} \\ & & \\ v_{2,1} & \cdots & v_{2,2} \end{bmatrix}_m$	
Std. Dev. of Policies Across Grps.	$\begin{bmatrix} rc_1 & \cdots & rc_p \end{bmatrix}_m$		

hr_g = element of the $[HR]_m$ matrix representing weighted mean of the corresponding $h_{g,h}$ elements for group g.

rr_g = element of the $[RR]_m$ matrix representing the weighted standard deviation of the elements $h_{g,j}$ around the corresponding group mean hr_g, using the weights w_{jm}.

hc_j = element of the $[HC]_m$ matrix representing the unweighted mean of the $h_{g,j}$ elements across all groups for policy j.

rc_j = element of the $[RR]_m$ matrix representing the unweighted standard deviation of each $h_{g,j}$ from the appropriate hc_j.

$v_{x,y}$ = element of the $[V]_m$ matrix summarizing efficiency and balance of total selection (see text).

Fig. 2 - The [H] matrix and associated derived statistical matrices

V. USER'S GUIDE TO THE FSM

The Finite Selection Model requires three basic kinds of inputs:

1. Problem definition inputs.
2. Program command inputs.
3. User defined data base.

The problem definition inputs and the program command inputs will be defined in this section; the user-defined data base will be discussed in Section VI.

PROBLEM DEFINITION INPUTS

Problem definition inputs establish the model's runtime environment. These inputs include the model control parameters, objective function specifications, and constants used in the model calculations. There are two types of problem definition inputs:

1. Runtime parameters defined by a FORTRAN NAMELIST input data specification.
2. Policy labels and objective function definitions specified through a free format input stream of tokens.

The form of input used by the first type is well defined by FORTRAN language specifications. In the second type, a character string or an integer separated by one or more blanks is defined as a token.

Runtime Parameters Using FORTRAN NAMELIST Input

Certain basic model parameters that are required to define correctly the arrays used by the model and thereby determine necessary core requirements, are set by using standard FORTRAN NAMELIST input conventions. These parameters normally remain static for the duration of any model run. They are listed in Table 2 with their definitions and defaults.

Table 2

NAMELIST INPUT PARAMETERS

Name	Definition	Default
NBCF	Maximum number of independent variables stored for each point in the [X] matrix.	None (required input)
NBCFMX	Maximum number of independent variables used within any single objective function.	NBCF
NPOL	Maximum number of policies (linear combinations of independent variables) to be estimated.	NBCF
NPLANS	Number of plans (groups).	None (required input)
MODELS	Number of objective functions.	1
MAXIT	Search length; maximum number of searches involved in selection of a single set.	50
NSETS	Maximum number of sets (maximum number selections by all plans).	None (required input)
NPTS	Maximum number of points (all plans).	None (required input)
FLGM	Print Z test and standard deviations for policies by plan.	.TRUE.
FLGW	Print policy information; names, weights, variances.	.TRUE.
FLGID	Print ID numbers of selected sets on each plan.	.TRUE.
FLGA	Not in present use.	.TRUE.
FLGD	Print counts of sets by strata	.FALSE.
FLGDB	Prints debug outputs; selected order of sets, determinant for [Q0] matrix, [Q0] matrix, [Q0] [S0] matrix product.	.FALSE.
FLGSF	Creates a data set containing the selected sets and associated plan assignments. The format of the data set is described in Section VII.	.TRUE.

Policy Labels and Objective Function Specification

Policy labels with weights assigned to policies by various objective functions are indicated sequentially, following each policy label, for each function. Policy weights are represented by integers that are normalized after input across all policies included within a single objective function. Negative policy weights indicate that the variate is not included in that objective function. The following table shows the specifications of two objective functions involving several policies each:

	Function 1	Function 2 ...	Function n
Policy Label	Policy Weight	Policy Weight	Policy Weight
AGE	-100	100	.
SEX	200	-100	.
RACE	100	50	.

Note that Function 1 includes SEX and RACE (with SEX weighted twice as much as RACE), and that Function 2 includes AGE and RACE (with AGE weighted twice as much as RACE). NPOL must be set to the correct number of policies, and MODELS must reflect the number of objective functions defined. Both parameters are part of the NAMELIST input section.

PROGRAM COMMAND INPUTS

Program command inputs cause specific actions to be performed by the model other than initializing a program environmental value. These usually cause the creation of a particular output, initiation of a sequence of calculations, or specification of relationships that exist among the sets. The program command inputs generally follow the problem definition inputs. Only those commands required for a specific application need be used in any given run.

The program command inputs have a form similar to the "stream of tokens" described previously. However, the first token defines the command, which must be a character string recognizable to the model. Other tokens representing command arguments follow the command token. A discussion of the input requirements of the individual commands follows. Figure 3 on page 33 is a flow diagram of the FSM program and

illustrates the effect of the program command inputs on the model. A summary of the FSM program commands can be found in Table 3 on page 36.

FSM PROGRAM COMMANDS

The MODEL Command

Function weights that determine the degree to which each objective function influences the selection process must be supplied for each of the functions. This is done by issuing the MODEL command followed by the corresponding function weights in sequence. The weights are input as integers and are normalized across all functions. The following schema exemplifies the MODEL command:

Command	Weight 1	Weight 2	Weight 3 ...	Weight n
MODEL	100	50	200	

In the illustration above, Function 1 is weighted twice as much as Function 2, but half as much as Function 3. The default of equal unit weights across all functions will result if this command is excluded.

The STRATA Command

Sets may be placed in different strata by the user. Since build-up is performed by strata, assignment of sets to groups can occur for sets belonging to the stratum specified for build-up only. Sets belonging to other strata are disregarded until the build-up stratum is changed to an appropriate matching stratum number.

Sets that are excluded from the selection process, but are included in the data base (just for purposes of participating in estimating the coefficients of the governing equation) should be placed in negative strata. This saves FSM computing costs because the sets in negative strata are grouped separately and are never considered during selection. The following procedure enables the user to stratify sets:

```
Command     Strata number      Set ID 1    Set ID 2    Set ID 3...$
            ...Set ID n-1       Set ID n

STRATA          99     5         34          53      71      $
               100    20
```

The above example places Sets 5, 34, 53, 71, 100 and 20 into Stratum 99. Strata numbers may be any integer that corresponds to a valid set identifier in the data base. If more than 72 columns are required to indicate the stratum members, a "$" must be included as the last symbol before column 72 to indicate that the list of stratum members continues on the next card.

The PLAN Command

Sets may be preassigned to plans (groups) prior to any build-up process. This allows selection to be broken up into numerous discrete steps. The output assigned to sets of the last selection may be preassigned to appropriate plans in the input to the next selection run. Thus, the final selection may, in fact, be the accumulation of results from several runs. The procedure for assigning sets to plans is similar to assigning sets to strata. The following schema is an example of the PLAN command.

```
Command     Plan number        Set ID 1     Set ID 2...               $
            ... Set ID n-1           Set ID n

  PLAN        12     5            9            17       30            $
               4    50
```

The above illustration shows the indicated sets to Plan 12. To continue a set list across several cards, the "$" continuation convention must be used with the PLAN card.

The RUN Command

The RUN command initiates the computation of several matrices that are either printed directly and/or saved for use in the build-up phase. Briefly, the calculations performed include:

1. The [Q0] matrix for each objective function, ([Q0] = ([X]' [X])/(NPTS-NBCF-1)).
2. The correlation matrix for each objective function.
3. The variance of the policy coefficients for each objective function.
4. The [D] matrix ([D] = [P]' [W] [P] and is called T in [2]).

Several arguments need to be supplied on the RUN command, as in

Command	S Multiplier		Plan 1 Build-up Size	Plan 2 Build-up Size...$
	...	Plan n-1 Build-up size	Plan n Build-up size	
RUN	300	10	20	15

The "S multiplier" above represents some number of average set increments to be preassigned to each plan to insure that the resulting S matrices are invertible before any set assignments are attempted during the build-up phase. The S multiplier as input is divided by 1000 and multiplied by a build-up plan size. The build-up plan sizes follow the S multiplier on the RUN card. A final plan size must be input for each plan defined for the FSM run. Thus, in the example shown above, an S multiplier of .3 (300/1000) is used on the three defined plans that are to have final sizes of 10, 20, and 15 set increments, respectively. The initial number of average set increments given in each plan's [S] matrix is as follows:

Plan	Final Plan Size	S Multiplier	Average Set Increment Assigned
1	10	.3	3
2	20	.3	6
3	15	.3	4.5

Effectively, this puts 3, 6, and 4.5 typical points in the designated plans.

An S multiplier of 10 is frequently recommended. This preassigns 1% of the sample. An S multiplier of 1 preassigns .1% and will give a very accurate characterization of the precision obtained from the

regression model. A large value would be assigned to the S multiplier if a singular or nearly singular covariance matrix is expected, but is not of significant concern.

The RANDOM Command

The RANDOM command causes an initial randomization of the order in which sets are considered for build-up or build-down. The command may be included in the input sequence to protect against possible bias in the consideration of sets for selection due to set order after model initialization. It only has an effect when MAXIT is less than the number of sets available for selection. The sets are randomized on the basis of a random number seed that is fed into a random number generator. The ensuing random sequence is dependent on the initial seed. Thus, if the user wants to regenerate a particular random sequence, the RANDOM command allows a seed to be specified. Otherwise, if a seed is not indicated, the model randomly generates one and it is printed so that it may be used later to regenerate the same random sequence. An example of the RANDOM command is shown below:

Command	Random Seed (optional)
RANDOM	5267611

The Build-Up/Build Down Commands

The FSM allows three types of build-up or build-down commands. These commands are characterized by the selection strategy they use in deciding which plan is to have the next set assigned to it or deleted from it. The commands implementing the defined strategies are shown below.

1. Local selection strategy
   - LOCAL
   - INCRLOC
2. Global selection strategy
   - GLOBAL

- INCRGLOB

3. User predefined Selection Order Matrix (S.O.M.)
   - MANUAL

The MANUAL command is the simplest to explain. It implies that the selection of the set gaining or losing will be done "manually" rather than algorithmically as in the other two categories. It is a manual means in that the user is forced to supply an S.O.M. specifying the plan losing a set, the plan gaining the set, and the stratum from which the chosen set is to be taken. This three value vector must be detailed for each set addition or deletion to every plan, until the desired plan sizes are reached. In this scheme, the unassigned set pool is plan zero and the set of assignment vectors follow the MANUAL command card, one vector per card. The series of indicated assignment vectors must be terminated with a blank card. An example of a MANUAL command is shown below:

MANUAL

From Plan	To Plan	From Stratum
MANUAL		
0	1	3
1	0	2
(...blank card ...)		

In this example, two assignments are specified. A set from stratum 3 is assigned to Plan 1 from the unassigned pool. Then, Plan 1 loses a set from stratum 2 to the unassigned pool.

The commands employing local or global selection strategies are conceptually more complex but easier to use, since the selection order matrix is not required. Section IV discusses the algorithms and explains the process of determining the plan for each strategy. The user must decide which strategy he prefers, and we recommend that he consult with a statistician in making his decision.

The GLOBAL, INCRGLOB, LOCAL, INCRLOC commands all require the same arguments to be input at the time that the commands are used. Since any particular build-up/build-down is performed by stratum, the appropriate stratum is indicated as the first argument. The desired plan sizes

after this build-up/build-down step are listed sequentially across the card following the stratum number. The continuation convention ($) should be used if more than one card is required. An example of the GLOBAL command is shown below:

Command	Stratum Number	Plan Size 1	Plan Size 2	Plan Size ... 3	$
	... Plan Size n-1	Plan Size n			
GLOBAL	1	2	3		
GLOBAL	2	4	6		

In the above illustration, assuming that initially all plans had no assignments, the first GLOBAL command builds up plan 1 to 2 set increments and plan 2 to 3 set increments from the group of sets included in Stratum 1. Then, the next GLOBAL command increases Plan 1 to 4 set increments and Plan 2 to 6 set increments from sets that are members of Stratum 2.

The plan sizes indicated on these command cards are not necessarily the final plan sizes desired but rather the desired plan sizes after the execution of the single build-up/build-down step. This points out the difference between the LOCAL/GLOBAL commands and the INCRLOC/INCRGLOB commands. The INCRLOC/INCRGLOB commands produce the same results, but rather than desired plan sizes, the number of set increments added to each plan are specified:

Command	Stratum Number		Plan 1 Increment	Plan 2 .... Increment	$
	... Plan n-1 Increment		Plan n Increment		
INCRGLOB	1	2	3		
INCRGLOB	2	2	3		

This example should produce the same results as the last example showing use of the GLOBAL command.

A few options have been implemented to make these commands easier to use. If the stratum number is set to zero, then the stratum is

disregarded during the execution of the command. Sets from any strata are available for reassignment. Also, if "*" is indicated as a plan size on a LOCAL or GLOBAL command card, the indicated plan size will remain unchanged from its previous value:

LOCAL	0	4	6
LOCAL	1	*	4

The above illustrates the last few points. Assuming that the initial plan sizes are zero, sets from any stratum can be used to build up Plans 1 and 2 to the sizes of 4 and 6 respectively. Then, Plan 2 is ordered to discard 2 set increments from stratum 1, while Plan 1 remains unchanged.

The MAXIT Command

The MAXIT command (MAXimum ITerations) sets the search length for the selection process. The search length indicates the maximum number of sets considered for the plan during the assignment of a single set. If fewer sets are available for selection than are indicated by the search length, then only these sets will be considered. The command has the same function as the MAXIT variable in the NAMELIST input section. If MAXIT remains static throughout the selection run, the MAXIT command may be omitted because the parameter has been set previously. However, if the search length is to change at some point in the run, it must be done through the MAXIT command.

The MAXIT command, as illustrated below, is followed by a single argument that indicates the search length:

Command	Search length
MAXIT	10

Random samples may be selected by using a search length of 0 or 1, provided the set order has been randomized either by design on input or through the use of the RANDOM command. A search length of 1 is wasteful because it will cause the FSM to perform the associated cost effectiveness calculation it normally needs to make during selection,

whereas the set must be selected regardless of the outcome of this computation. A search length of zero assigns the set without making these unnecessary calculations.

The PRINT Command

The PRINT command causes a standard set of outputs to be printed. Included in the outputs are the following statistics:

1. Plan-related totals of sets, set increments, points, and costs.
2. Z-test statistics, and standard deviation of policy variates belonging to the plan.
3. Listing of set ID's of the selected set.
4. [H] matrix statistics showing unnormalized and normalized variances of policy variates by plan and over all plans.

Although the PRINT command may be inserted at any point in the output stream, the derived information is usually of particular interest after the final, desired build-up or build-down plan sizes have been attained. The PRINT command, as shown below, is input separately without any additional arguments:

Command
PRINT

The DUMPS Command

The DUMPS command (DUMP the Sets) is used to print all pertinent set information at any point in time for all sets in the data base. Listed in the output are the following:

1. Set identifier.
2. Number of set increments included.
3. Number of points included.
4. Assigned plan (zero if unassigned).
5. Assigned stratum (default is 1).

6. Total cost associated with a set.

The DUMPS command is only used if the user wants to check set status at any point during the run; the command is input separately without any additional arguments. A simple structure of the DUMPS command is shown below:

Command

DUMPS

The REMOVE and INSERT Commands

The REMOVE and INSERT commands enable the user to edit the data base after it has already been read and the [X] matrix has been created. The REMOVE command logically excludes indicated sets from the data base so that the excluded sets will not have any effect on the ensuing calculations. The INSERT command includes previously excluded sets into the data.

The set identifiers excluded/included must follow the command on the card. Examples of the REMOVE/INSERT commands are shown below:

Command	Set ID 1	Set ID 2	...	Set ID n
REMOVE	10	20	30	4
INSERT	20	30		

In the example, Sets 10, 20, 30, and 4 are logically excluded, and then Sets 20 and 30 are reincluded in the data base.

The POLICY Command

The POLICY command allows the user to change policy weights across objective functions that had been specified previously as problem definition inputs. The command requires several arguments; these include the name of the policy to be altered and the policy weights for each objective function defined. An example of the POLICY command is shown below:

Command	Policy Name	Weight for Function 1	Weight for Function 2	...	Weight for Function n
POLICY	AGE	10	-10		5

The POLICY command shown will change the policy weight on AGE to 10 for Function 1 and -10 for Function 2.

The CLEAR Command

The CLEAR command permits the user to reassign all sets that have a positive stratum number to the free or unassigned pool. This command negates the effect of any previous build-up step. The command is input at any point in the command stream and requires no additional arguments:

Command CLEAR

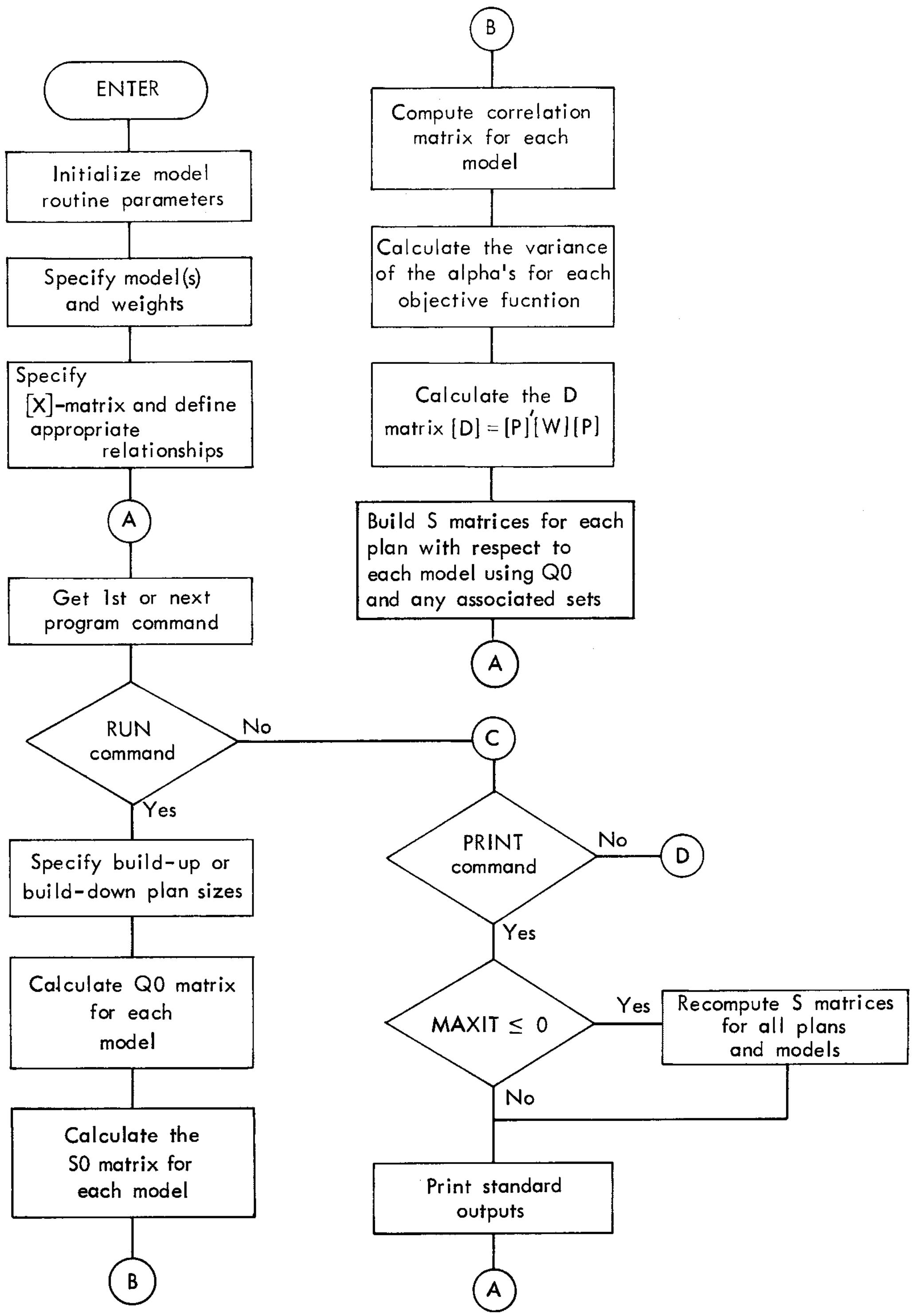

Fig. 3 —Top level FSM flow of control

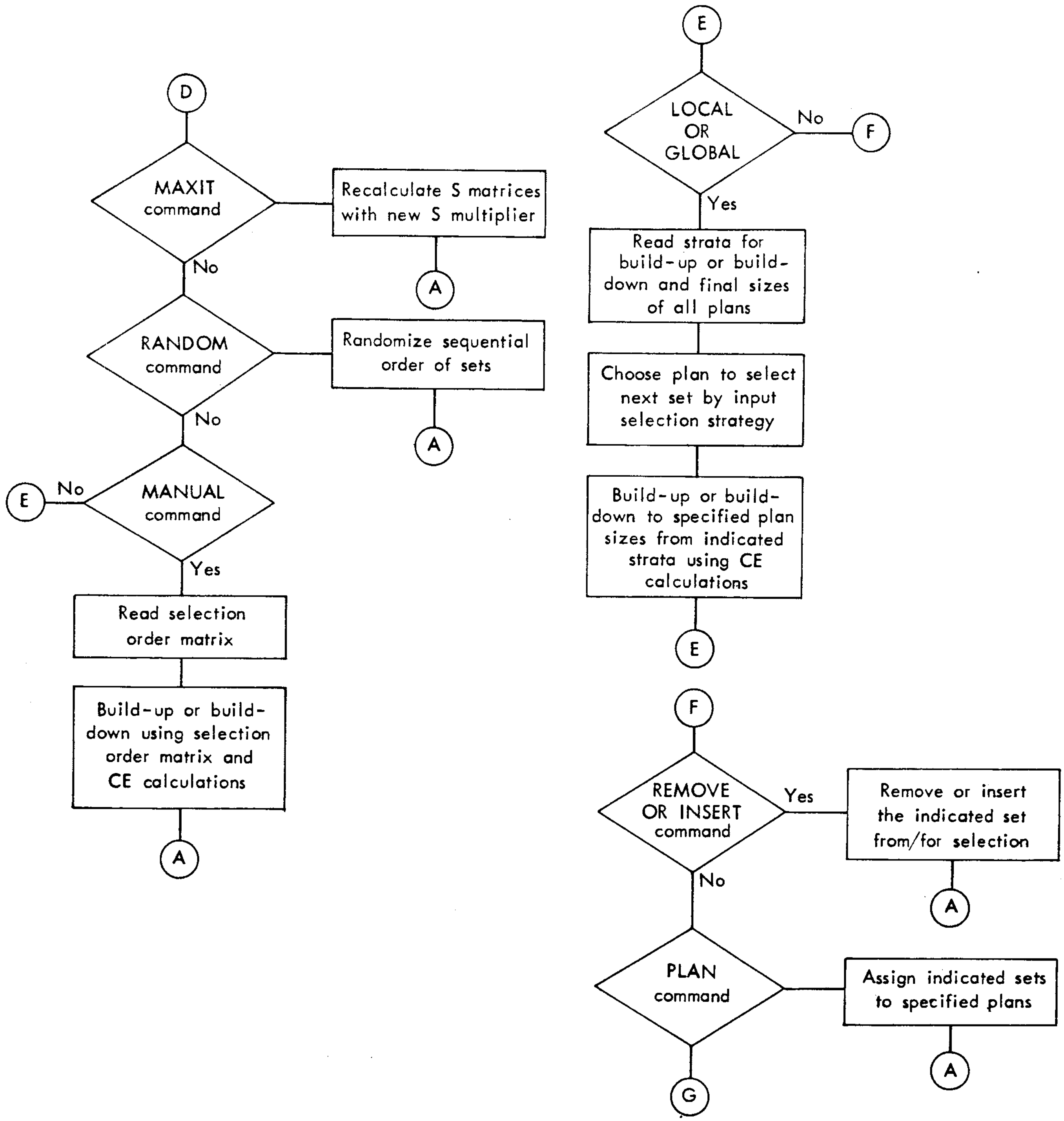

Fig.3 — Top level FSM flow of control— continued

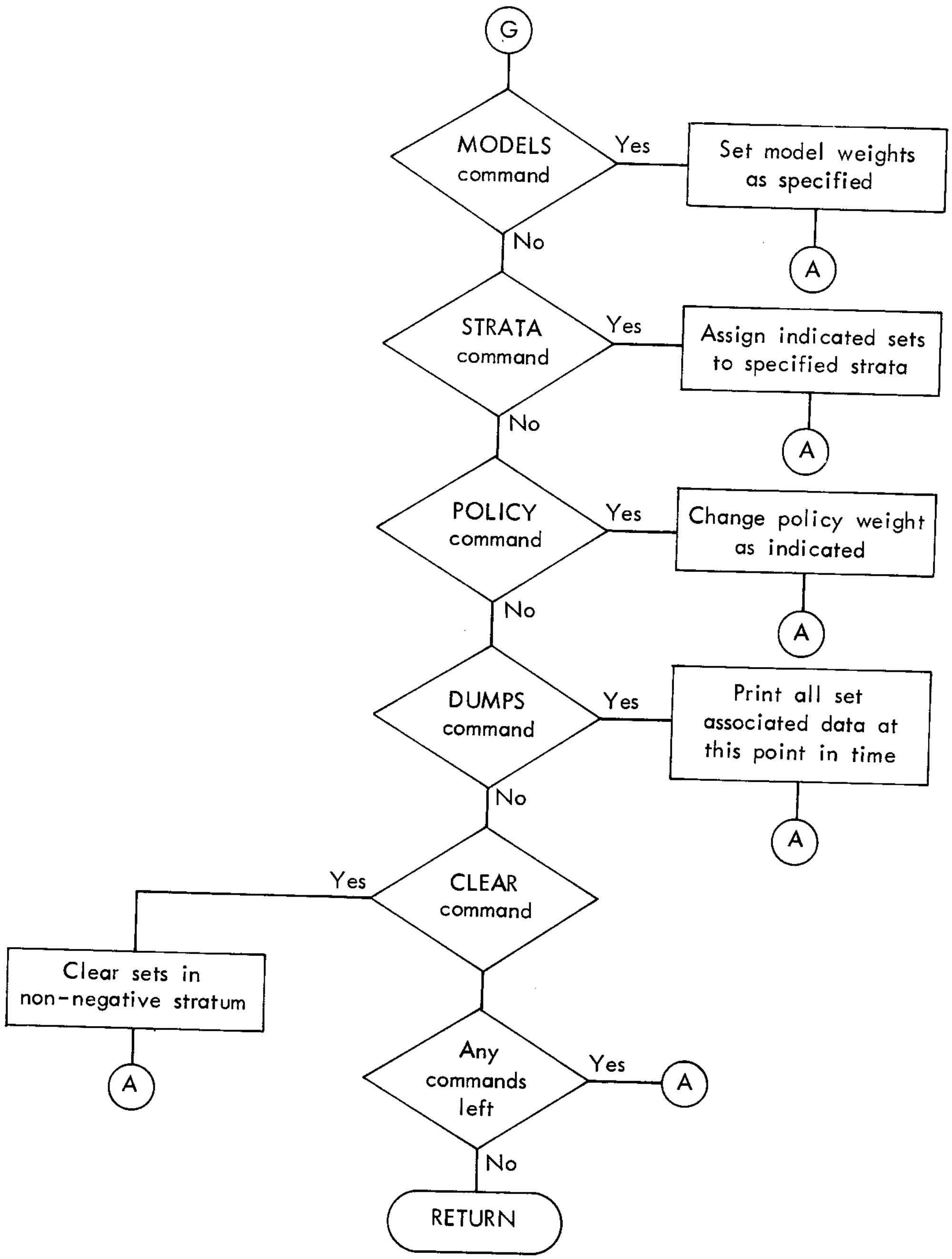

Fig.3 — Top level FSM flow of control — continued

Table 3

SUMMARY OF FSM COMMANDS LISTED IN ALPHABETICAL ORDER

Command	Function
CLEAR	Reassigns all sets in positive strata to the free or unassigned set pool.
DUMPS	Prints out internal information on all sets.
GLOBAL	Builds up or builds down by using the global selection strategy. Desired plan sizes are provided as arguments.
INCRGLOB	Builds up or builds down using the global selection strategy. Increments to last plan sizes are provided as arguments.
INCRLOC	Builds up or builds down by using the local selection strategy. Increments to last plan sizes are provided as arguments.
INSERT	Inserts previously logically excluded sets into the data base.
LOCAL	Builds up or builds down using the local selection strategy. Desired plan sizes are provided as arguments.
MANUAL	Builds up or builds down by using a user input S.O.M.
MAXIT	Alters search length to the value of the specified argument.
MODELS	Distributes weights among the defined objective functions.
PLAN	Assigns specified sets to a particular plan.
POLICY	Changes the policy weight of the specified policy for all objective functions.
PRINT	Generates standard outputs.
RANDOM	Randomizes the sequence in which sets are considered for assignment to plans.
REMOVE	Logically excludes specified sets in the data base.
RUN	Initiates preliminary calculations necessary to perform assignments, generates statistics on the data base. Final, desired plan sizes are provided as arguments.
STRATA	Assigns specified sets to a particular stratum.

VI. OPERATIONAL REQUIREMENTS OF THE FSM

CAUTIONARY NOTE

The Finite Selection Model includes many computational options. The user should realize that it will produce a good design in relation to the inputs he makes, but may not if he misspecifies his model and criteria. The FSM can be used in a way that reduces sensitivity to these criteria if the user (a) sets up a discard group as a treatment group if there are excess objects for selection, (b) transforms his data to remove outliers, (c) uses a randomized selection order matrix (manually determined) with randomized strata order, (d) is careful about the choice of strata and the cost function, (e) uses the model with plan sizes that are not too small and that are not too unequal. The user is urged to talk with a knowledgeable statistician about these matters before making such decisions.

PRE-OPERATIONAL STEPS

Several steps must be completed before the FSM becomes operational for a specific application. These include

1. Preparation of the data base.
2. Specification of the objective function(s).
3. Definition of an the input routine.
4. Generation of the interface between the input routine and the body of the FSM.
5. Creation of the load module.

All of them are important processes and deserve the user's due consideration. Each is described below.

Preparation of the Data Base

The user is responsible for providing the data base needed by the FSM. The basic logical building blocks of the data base are point records. These records include the ordered values of all defined variates for a single point. The meaning of "point" is left to the

user. If, for example, the point is defined to be a person, then the point record consists of an array of values of independent variables describing a person. The FSM then reads the data base, one point record at a time, and builds the desired point vector of data elements to be included in the [X] matrix.

Constructing the data base and running the FSM against it is a two step, disjoint process. The data base should be constructed in such a way as to anticipate changes in the variables used by the objective function. By including currently unused variables in the point record, the user may save himself the burden of having to recreate the data base at some future time to include currently excluded data. The unused variates may be excluded through specification of negative runtime policy weights as part of the problem definition inputs.

Similarly, the inclusion of complete unused point records may be beneficial. The undesired points may be kept from entering the [X] matrix by including appropriate logic within the body of the user's input routine or by logically removing the points with a specific REMOVE command.

Specificaton of the Objective Function

Specification of the objective function(s) to use as selection criteria creates several problems that are not apparent at first. In practice, the user is most concerned with defining an objective function that embodies relationships among the variates that are of concern to his understanding of the problem. Additionally, the user must make sure that the following conditions are met:

a. The final plan sizes must be large enough to estimate the specified equation(s).
b. Categorical variables should be omitted if they include values that occur infrequently relative to the number of plans being assigned.

The first condition is consistent with basic statistical practice. If the objective function involves estimating n coefficients (n-1 independent variables and the constant term), then each resulting

sample must include at least n members. The second condition concerns the FSM's design concept of balance among plans. For example, suppose race is to be included as a variable in the objective function and may have values of "white" or "nonwhite." Furthermore, suppose four samples are to be selected and only two sets are classified as "nonwhite." In this case, it will be impossible for the sample to balance on race because only two plans have the possibility of being assigned sets that include both "white" and "nonwhite" members. Thus, any set characteristic that is too scarce to be distributed among all groups cannot be balanced.

Definition of Input Routine

After an objective function has been determined, the user is ready to design an input routine that will access the data base and create the [X] matrix required for FSM calculations. The input routine must execute the following functions:

a. Reading the data base.
b. Initializing model arrays that describe data relationships and specification of a cost function.
c. Creating the point vector.
d. Initializing the [X] matrix.

Reading the Data Base. The data base may reside in any location and be physically file-structured in any way accessible to a FORTRAN subroutine. The subroutine must read the point records one at a time from the data base and decide whether or not they should contribute to the [X] matrix.

Initializing Model Arrays. The input routine must initialize certain arrays that are passed among the subroutines comprising the FSM. These arrays are initialized in Table 4 on page 42. The arrays that define characteristics of sets, or relationships between sets to points and sets to sets, are numbered 1 though 6 and are discussed in detail in the following paragraphs.

o SCOST

The cost of a set is a relative value that represents the marginal cost of assigning a particular set into a plan. If cost is not a consideration in assignment, the cost of any set may be set to the same constant. If the cost of the set is dependent on a function of variates in the point records, then this function should be specified in the input routine.

o SSIZE

The size of a set is defined as the number of points included in any particular set. If the set never spans more than a single point, then the set size for all sets will be unity (e.g., in this case NSETS=NPTS). If a set may span a variable number of points the input routine must have a way of determining which points belong to particular sets. Usually, this determination is based on an indentifier defined in the point record. Any point with the same identifier (i.e., household number) belongs to the same set.

o SID

The set identifier is a unique number that identifies a particular set. Set identifiers are used as arguments with model commands. If a set is to be assigned to a plan (i.e., PLAN command), included in a stratum (i.e., STRATA command), or omitted from consideration in the selection process (i.e., REMOVE command), the set identifier(s) must be provided with the command.

o SINCR

The set increment value array indicates the number of set increments that are included in the assignment of a particular set to a plan. For example, if households are being assigned to plans, but the final plan sizes are to be expressed in numbers of families, then the set represents the household relationship, and the set increments express the number of families in any household.

Set increments also may be interpreted as a measure of cost. If a marginal cost can be associated with each set, the set increment may be assigned this value to represent the cost of choosing the set. Plans may then have sets assigned to them as long as they do not exceed an indicated cost constraint. This implies that each plan must have a cost constraint associated with it instead of a final plan size.

o SPLAN

Finally, a set may be preassigned to a plan or stratum in the input routine if there is an automatic method for determining to which plan and/or stratum the set belongs. The effect of performing these assignments in the input routine is equivalent to using the PLAN and STRATUM commands. It is recommended that the commands be used whenever possible to keep the input routine simple.

Creation of the Point Vector. The user must create the point vector that is included in the [X] matrix. The point vector needs to include only those variates from the point record used by the objective function(s). Thus, once the point record has been read, appropriate transformation on specific variables must be performed and the necessary variates must be arranged in the order indicated by the Policy Labels and Objective Function Specification Inputs (see page 22).

Initialization of the [X] Matrix. Once the point vector has been constructed, it can be included in the [X] matrix (see item 7 in Table 4). A point counter is incremented every time a valid point vector is to be constructed. Then the counter is used as an index into the [X] matrix. An appropriate policy matrix also has to be specified. This matrix represents the linear combination of defined variates represented by each policy (see item 8 in Table 4).

The FSM arrays requiring initialization are summarized in Table 4. Figure 4 on page 43 is a general flow diagram of the FSM input routine. An example of an FSM input routine is illustrated in Fig. 16 in Appendix B.

Table 4

FSM ARRAYS REQUIRING INITIALIZATION

Number	Variable Symbol and Dimensions	Data Type	Definition
1	SCOST(NSETS)	Integer	Cost of any set included in the [X] matrix.
2	SSIZE(NSETS)	Integer	Number of points included in any set.
3	SID(NSETS)	Integer	User-defined set identifier; the name of the set.
4	SINCR(NSETS)	Integer	Number of set increments included in any set. The basic counting unit by which assignments are made to plans.
5	SPLAN(NSETS)	Integer	The plan number to which any set is currently assigned. (If unassigned, the plan number must be set to zero.)
6	SSTRAT(NSETS)	Integer	Stratum number assigned to any particular set.(If not stratified, all sets should be initialized to the same constant.)
7	X(NBCFMX,NPTS)	Double Precision	The array of point vectors; the [X] matrix.
8	P(NBCFMX,NPOL)	Double Precision	The array of vectors defining linear combinations of variates; the policy matrix.

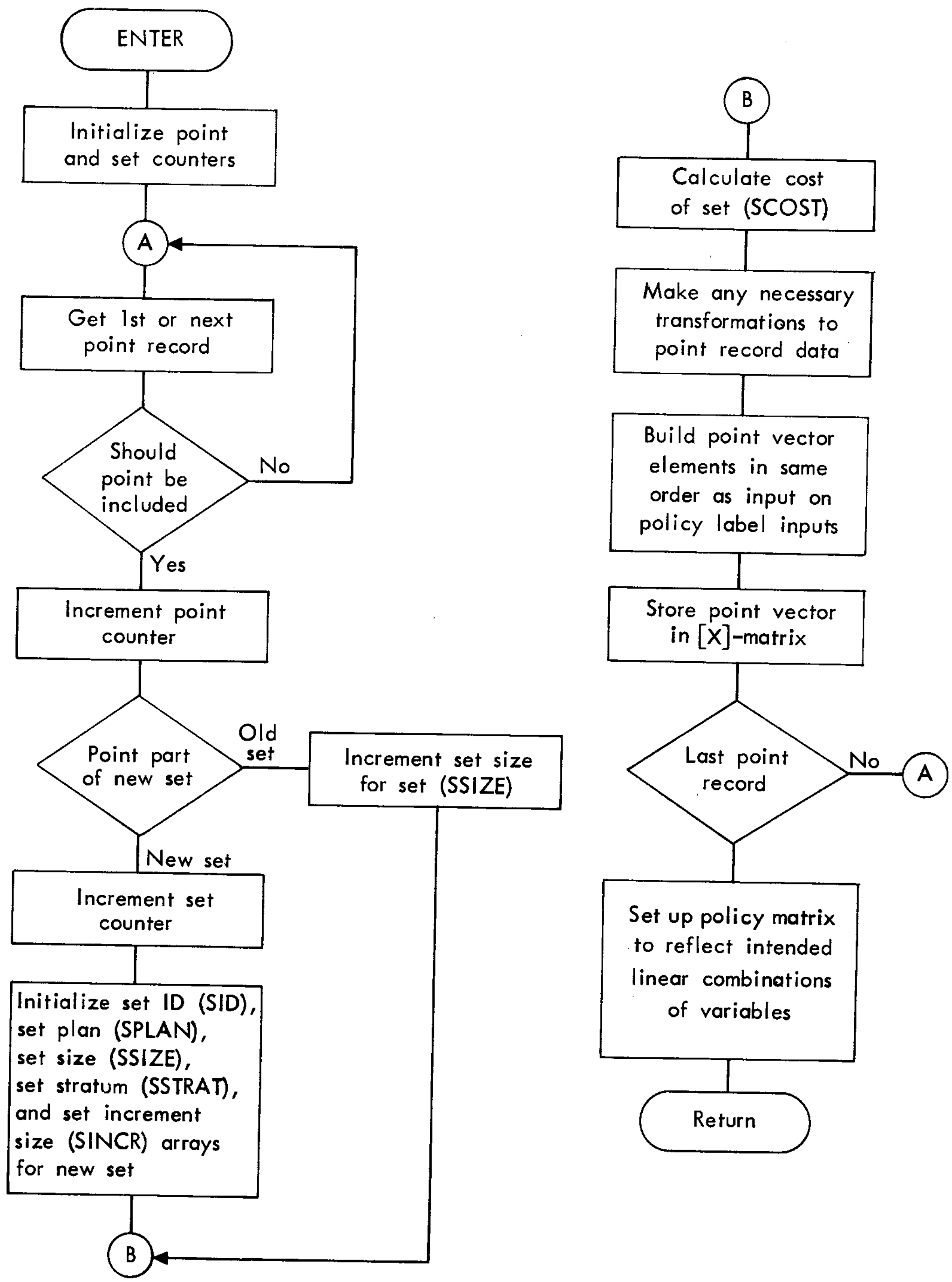

Fig.4 — Basic input routine flow chart

Interface between Input Routine and FSM Program

The user must use a predefined interface convention for the input routine in order to link it properly to the body of the FSM using the JCL illustrated below. Briefly, the interface requires that

1. The input routine be named DEF.
2. The input routine must communicate with the FSM in the form of a standardized argument list employed by all major subroutines of the FSM.
3. The elements of the argument list be of a predetermined type and structure, and that the elements be specified at the head of the input routine.

These requirements are expressed in standard FORTRAN IV in the following way:

```
 SUBROUTINE DEF(D,P,X,XT,V,VS,VSTD,S,Q0,ZS,LABS,PNSETS,PINCR,PPTR,
&PTEMP,PINCBE,PITER,SCOST,SSIZE,SFP,SINCR,SID,SPTR,STEMP,SPLAN,
&SSTRAT,WP,EVARA,WTEMP,WM,HMAT1,HMAT2)
 LOGICAL FLGW,FLGID,FLGD,FLGM,FLGA,FLGDB,FLGSF
 REAL*8 TOP,BOTTOM,T,TT,DET
 COMMON TOP,BOTTOM,EFF,NBCF,INSERT,NPOL,NBCFX,NPOLX,NLTX,NPLANS,
&NPTS,NSETS,MODELS,MAXIT,NPZERO,NIT,FLGW,FLGID,FLGD,FLGM,FLGA,
&FLGDB,FLGSF,SMULT,NPO,COSTO,WMSUM
 INTEGER PNSETS(NPZERO),PPTR(NPZERO),PTEMP(NPZERO)
 INTEGER PINCR(NPLANS),PINCBE(NPLANS),PITER(NPLANS)
 INTEGER SCOST(NSETS),SSIZE(NSETS),SFP(NSETS),SINCR(NSETS),
&SID(NSETS),SPTR(NSETS),STEMP(NSETS),SPLAN(NSETS),SSTRAT(NSETS)
 REAL*8 D(NLTX,MODELS),Q0(NLTX,MODELS),S(NLTX,MODELS,NPLANS),
&X(NBCFX,NPTS)
 REAL*8 ZS(NBCFX,NBCFX),LABS(NPOLX)
 REAL*8 XT(NBCFX),V(NBCFX),VS(NBCFX),VSTD(NBCFX),P(NBCFX,NPOLX)
 INTEGER WP(NPOLX,MODELS),WM(MODELS)
 REAL HMAT1(NPOLX,NPLANS,MODELS),HMAT2(NPOLX,NPLANS,MODELS)
 REAL WTEMP(NPOLX),EVARA(NPOLX,MODELS)
```

Creation of the Load Module

When the input routine is complete, the user is ready to produce a load module for his FSM production runs. This process is shown graphically in Fig. 5 on page 46. The input routine must be compiled to produce an object module. The object module is then linked to the body of the FSM code by using the linkage editor. The following Job Control

Language (JCL) is used to compile the input routine, create a load module, and execute the program on the IBM 370/158 computer:

```
//Job card
// EXEC FORTCL,REGC=140K,PARMC=SOURCE,PARML='XREF,LIST,MAP,OVLY',
//      LIBL='K.K1632.AR012.KDLIB'
//DDDEL DD DSN=name of load module,DISP=(MOD,DELETE),
// UNIT=USER,VOL=SER=PRIV01,SPACE=(TRK,(5,3,1))
//FORT.SYSIN DD *
      .
      .
      .
 FORTRAN Source
      .
      .
      .
//LKED.SYSLMOD DD DSN=name of load module,DISP=(,CATLG),
// UNIT=USER,VOL=SER=PRIV01,SPACE=(TRK,(5,3,1))
//LKEDOLD DD DSN=K.K1632.AR012.FSMA(PROG),DISP=SHR
  INCLUDE LKEDOLD(PROG)
  ENTRY MAIN
  OVERLAY ONE
  INSERT DEF
  OVERLAY ONE
  INSERT PRTHMT
  INSERT CPINIT
  INSERT MKS
  INSERT PRT
  INSERT XVECT
// EXEC PGM=PROG,REGION=300K,COND=(4,LT)
//STEPLIB DD DSN=name of load module,DISP=SHR
//FT01F001 DD DSN=name of the data base,DISP=SHR
//FT02F001 DD DSN=name of the output data set if FLGSF=T;
//                omit this 'DD' statement if FLGSF=F.
//            DCB=(RECFM=FB,LRECL=34,BLKSIZE=3400)...
//FT05F001 DD *
      .
      .
      .
FSM input stream
      .
      .
      .
//FT06F001 DD SYSOUT=A
//
```

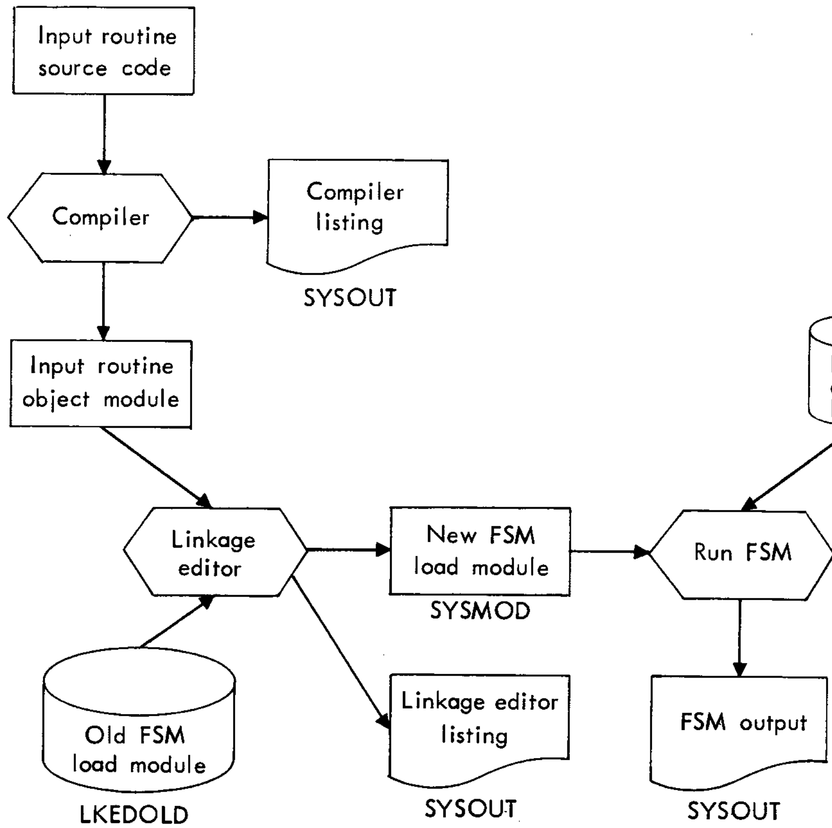

Fig. 5 — Creation and execution of FSM load module

VII. DESCRIPTION OF FSM OUTPUTS

The Finite Selection Model has the capability of producing an extensive number of output sections. Some of these sections can be suppressed by setting the print control flags appropriately (see Table 2, NAMELIST Input Parameters). The significance of each output is explained in the following paragraphs.

The FSM heading and all runtime parameters, including policy labels with the objective function specification, are printed at the top of the run. The heading includes the date and time so as to uniquely identify the run. The runtime parameters printed are those that have been input through the NAMELIST facility preset model defaults. The variable NCORE indicates the number of bytes of core needed by the model arrays assuming the input parameter values (see Appendix A).

The policy labels and objective function specification are printed exactly as they have been input by the user. Figure 6 illustrates a sample section.

The next output section gives a statistical description of the data assembled by the user's input routine in the [X] matrix. The total number of points and sets represented in the [X] matrix is output, along with the time consumed by the input routine (in seconds). This output is followed by two types of means and the standard deviation of each policy variate for all points represented (see Fig. 7).

The "DEF-mean" represents the raw mean of the data calculated by the DEF routine. The "mean" value indicates the normalized mean of the policy variate. If the "DEF-mean" has not been calculated, it will be printed as zeros.

Along with the output discussed above, any additional output sections included in an FSM run are a function of the print control flags set by the NAMELIST input section and the program commands found in the primary input stream.

```
FINITE SELECTION MODEL OF 22 NOV 75 ** WEDNESDAY, JULY 21, 1976
&IN
NBCF=           5,NBCFMX=          5,NPOL=          5,NPLANS=    4,
MODELS=           1,NSETS=          48,NPTS=          48,
FLGW=T,FLGID=T,FLGD=T,FLGM=T,FLGA=T,FLGDB=T,FLGSF=F,MAXIT=       10,
NCORE=       5312
&END

SEQ  NAME    WEIGHTS
  1 CONST      100
  2 POP70      100
  3 AREA       100
  4 LAT        100
  5 LONG       100
```

Fig. 6 - Policy labels and objective function specifications

THE NUMBER OF SETS INPUT: 48

THE NUMBER OF POINTS INPUT: 48

SEQ	LABEL	DEF-MEAN	MEAN	STD
1	CONST	0.9999999	1.000000	0.3725290D-08
2	POP70	7.868913	0.4690238D-06	1.014841
3	AREA	7.282001	0.4340412D-06	2.953730
4	LAT	39.34624	0.2345219D-05	4.605330
5	LONG	90.91457	0.5418931D-05	14.76103

Fig. 7 - Means and standard deviation of policy variates

Any program command is echoed in the printout in the following format:

INPUT program command text

Examples of this format are given in Fig. 8. Any output resulting from the actions invoked by the command follow this text. The outputs following the command echos are generally self-explanatory; however, the RUN, PRINT, and DUMPS command outputs will be discussed in more detail.

The RUN command produces output that indicates the progress of necessary preselection calculations. Initially the number of sets, set increments, and points are recalculated to detect if any sets have been logically deleted by the REMOVE command since a count was last performed. The results of this operation are produced for the user's inspection, along with the average cost of the included sets.

These results are followed by a determination of the actual numbers of data coefficients and policies used in the calculations. This determination involves decrementing the value of the NBCF and NPOL parameters once for each negative policy weight encountered in the objective function specification (see Fig. 9a).

A series of matrices are printed in the ensuing output. These illustrate the results of several calculations and include

1. A complete correlation matrix showing the correlation between two policies.
2. The [Q0] matrix for each model, $([Q0]_m = ([X]'[X])/(NPTS-NBCF=1))$.
3. The [S0] matrix for each model $([S0]_m = -[Q0]_m^{-1})$.
4. The product of $[Q0]_m$ and $[S0]_m$.

The last matrix is printed only for checking purposes. If the calculations are correct, the diagonal elements of the matrix should all be ones and the off diagonals zero (see Fig. 9b, c, d, and e on pages 51-52).

```
*INPUT* MODEL   1


*INPUT* MAXIT   0


*INPUT* PRINT


*INPUT* RANDOM
RANDOM SEED IS:       -1876884337



*INPUT* LOCAL   0    0    0    0    0
```

Fig. 8 - Program commands

INPUT MODEL 1

INPUT RUN 001 12 12 12 12

NUMBER OF SETS 48

NUMBER OF INCR 48

NUMBER OF POINTS 48 48

AVE COST PER INCR 1.000000

INITIAL COMPUTATIONS FOR MODEL 1

THE NUMBER OF BETA COEFFICIENTS: 5

THE NUMBER OF POLICIES: 5

Fig. 9a - Objective function specifications

CORRELATION MATRIX OF X

		CONST	POP70	AREA	LAT	LONG
1	CONST	1.000000				
2	POP70	0.000000	1.000000			
3	AREA	0.000000	0.069297	1.000000		
4	LAT	0.000000	-0.321259	-0.162328	1.000000	
5	LONG	0.000000	-0.220237	0.782525	0.076358	1.000000

Fig. 9b - Correlation matrix of X

QO MATRIX

1	0.114D 01				
2	0.536D-06	0.118D 01			
3	0.496D-06	0.237D 00	0.997D 01		
4	0.268D-05	-0.172D 01	-0.252D 01	0.242D 02	
5	0.619D-05	-0.377D 01	0.390D 02	0.593D 01	0.249D 03

Fig. 9c - [Q0] matrix

SO MATRIX

1	0.875D 00				
2	-0.793D-06	0.110D 01			
3	0.169D-06	-0.195D 00	0.331D 00		
4	-0.121D-06	0.466D-01	0.343D-01	0.496D-01	
5	-0.573D-07	0.461D-01	-0.556D-01	-0.584D-02	0.136D-01

Fig. 9d - [S0] matrix

Q0*S0

ROW	FIRST COL	1, LAST COL	5		
1	0.100D 01	-0.529D-22	-0.847D-21	-0.569D-21	0.0
2	-0.529D-22	0.100D 01	0.888D-15	0.971D-16	0.888D-15
3	-0.132D-22	0.139D-16	0.100D 01	0.971D-16	0.444D-15
4	0.265D-22	-0.304D-16	-0.971D-16	0.100D 01	-0.222D-15
5	0.132D-22	0.260D-17	0.153D-15	-0.139D-16	0.100D 01

Fig.9e - Product of [Q0] and [S0] matrices

The next part of the output from the RUN command is controlled by the FLGW flag. If it is set, the values of the policy weights and the variance of the policy coefficients are printed. Also, the lower-left hand portion of the policy matrix beyond the NBCF numbered row is produced for the user's inspection (see Fig. 10a on page 55).

The final printout resulting from the RUN command shows several tables of miscellaneous counts. These tables include

1. Desired plan sizes as input on the RUN command.
2. Counts of sets, set increments, and points by strata, along with the average costs of the associated sets.
3. Current status data on each plan.

The control flag FLGD must be set for the second table to be printed (see Fig. 10b on page 55).

The PRINT command prints the resulting statistics of any build-up or build-down process. It provides a plan-by-plan comparison of the current set assignments. For each plan, the following outputs are presented:

1. A table of counts by plan with associated costs, and total efficiency and cost effectiveness statistics.
2. Statistics indicating the status of the variates of each plan policy to the total population
3. A listing of the identifiers of the sets which are assigned to the plan.

The measures of the total relative efficiency of the plan are presented in several forms immediately following the plan counts (see Fig. 11a). Using the same statistical heading found on the output, the given statistics are calculated by means of the illustrated computations shown below:

$$\text{PHI-NORM}_g = \sum_{m=1,\text{MODELS}} W_m(\text{trace}([D]_m[S]_{gm}))/(\text{expected trace}[D]_m[S]_{gm})$$

$$EFFCY_g = \sum_{m=1,MODELS} \frac{W_m(\text{average set-cost})(\text{expected trace}([D]_m[S]_{gm}))}{(\text{trace}([D]_m[S]_{gm}))(\text{average set cost for plan sets})}$$

$$PHI_g = \sum_{m=1,MODELS} W_m(\text{trace}([D]_m[S]_{gm}))$$

where WM_m is the weight assigned to m^{th} objective function. WM is used to distinguish the weight of Modal m from the weights on policies within each Model that are included implicitly in $[D]_m$.

The statistics in the second part show how well each policy variate within each plan was prepresented when it is compared with the total population (see Fig. 11b) The "Z-Test" shows the departure of the policy mean for the plan from that of the policy mean for the total population relative to the population standard error. The "STND-DEVS" heading indicates standard deviations of policy variates for the plan divided by policy standard deviations of the set population. The control flag FLGM must be set to obtain both of these outputs.

The set identifiers of the sets assigned to each plan are printed beside the "SELECTED SETS" heading. The print control flag FLGID must be set appropriately for this output to appear (see Fig. 11c on page 56).

POL	VAR-NAME	W(POL)	VAR(A0(POL))	POLICY VECTOR
1	CONST	100	0.7083334	
2	POP70	100	1.104684	
3	AREA	100	0.3310848	
4	LAT	100	0.4955480E-01	
5	LONG	100	0.1356166E-01	

Fig. 10a - Policy weights and variance of policy coefficients

PLAN SIZES: 12 12 12 12

STRATA	SETS	INCR	PTS	AVE COST
1	24	24	24	1.00
2	24	24	24	1.00

PLAN	MODEL	AVAIL SETS	INCR	INCR-TO-BE	DET-Q
1	1	0	0	12	0.5206765D-05
2	1	0	0	12	0.5206765D-05
3	1	0	0	12	0.5206765D-05
4	1	0	0	12	0.5206765D-05

Fig. 10b - Miscellaneous counts

INPUT PRINT

PLAN 1 **************************

	SETS	INCR	PTS	COST	AVE COST
OLD	0	0	0	0	0.0
NEW	12	12	12	12	1.000000
TOT	12	12	12	12	1.000000

PHI-NORM 0.9429667 EFFCY 1.060482 PHI 0.6629425E-01

Fig. 11a - Counts by plan with associated costs

Z-TEST	***************	0.3237323	-1.1001067	-1.0073616	-1.2174939
STND-DEVS	2.0000000	1.0385402	0.8799058	0.9431554	0.8539575

Fig. 11b - Plan statistics

SELECTED SETS 3 9 11 12 20 22 23 24 28 29 43 44

Fig. 11c - Identifiers of selected sets

The final output of the PRINT command shows the unnormalized and normalized versions of the [H] matrix. The two matrices are of identical plan row by policy column format, where the unnormalized matrix is divided by the expected [H] matrix for a uniform random sample to produce the normalized [H] matrix (see Fig. 12).

The DUMPS command is mainly used as a debug feature when the FSM is being run. When executed, it prints set status data for all sets included in the [X] matrix. It may appear at any point in the model command stream, and reflects current set parameters existing at the point where the command is executed (see Fig. 13).

Use of the FLGDB flag with any build-up or build-down command produces an output that shows the actual selected order of sets during the process. This feature is used primarily for debug purposes, although it does provide some feedback to the new user illustrating the internal logic of the model (see Fig. 14).

An output data set is created if the flag, FLGSF=.TRUE.; this data set contains the selected sets and the associated plan assignments. The format of the data set is as follows:

Variable Name	Position	Format	Description
SID	1- 5	I5	Set Identifier
SSIZE	6-13	I8	Set Size
SINCR	14	I1	No.Set Increments
SPLAN	15-18	I4	Plan Assignment
SSTRAT	19-34	I16	Strata

MATRIX OF UNNORMALIZED VARIANCES

PLAN	CONST	POP70	AREA	LAT	LONG	MEAN	STD
1	0.106461	0.147972	0.068980	0.005493	0.002565	0.066294	0.056657
2	0.090062	0.106818	0.033299	0.003592	0.001181	0.046990	0.043827
3	0.093940	0.161427	0.054027	0.006248	0.002027	0.063534	0.059435
4	0.087260	0.086551	0.020202	0.007467	0.000783	0.040452	0.038439
MEAN	0.094431	0.125692	0.044127	0.005700	0.001639	0.054318	0.048879
STD	0.007339	0.030256	0.018744	0.001406	0.000698	0.010889	0.016267

MATRIX OF NORMALIZED VARIANCES

PLAN	CONST	POP70	AREA	LAT	LONG	MEAN	STD
1	0.853392	0.805307	1.252568	0.666452	1.137120	0.942967	0.217708
2	0.721940	0.581334	0.604656	0.435792	0.523539	0.573452	0.094368
3	0.753019	0.878531	0.981048	0.758014	0.898688	0.853860	0.087352
4	0.699476	0.471035	0.366831	0.905871	0.347058	0.558054	0.214305
MEAN	0.756956	0.684052	0.801275	0.691532	0.726601	0.732083	0.043332
STD	0.058835	0.164661	0.340362	0.170583	0.309608	0.169375	0.232971

Fig. 12 - Matrix of unnormalized/normalized variates

```
*INPUT* DUMPS
 SEQ    INDENT  INCR  SIZE  PLAN  STATUM  COST
   1         2     1     1     5      30    11
   4         7     1     4     1       8    11
   7        17     1     2     4       6    11
  10        41     1     1     1      15    11
  13        57     1     2     5       6    15
  16        71     1     1     3       5    11
  19        76     1     1     1      30    11
```

Fig. 13 - Data status of sets in [X] matrix

```
*INPUT* LOCAL    6       9        4       9        5          16
FROM     0, TO    5,STRATA    6, ID   66, EFF -0.9976737E-02
FROM     0, TO    1,STRATA    6, ID   47, EFF -0.7284109E-02
FROM     0, TO    3,STRATA    6, ID  227, EFF -0.2439186E-01
FROM     0, TO    5,STRATA    6, ID  220, EFF -0.2656237E-02
FROM     0, TO    4,STRATA    6, ID  282, EFF -0.2307809E-02
FROM     0, TO    5,STRATA    6, ID  249, EFF -0.2439186E-01
```

Fig. 14 - Order of selected sets

Appendix A

FSM INSTALLATION AND COSTS

The Finite Selection Model was written in a free format computer language called MORTRAN. The MORTRAN source is converted into equivalent FORTRAN statements by the MORTRAN processor. Thus, if the user wants to install an FSM at his computer facility, he may use the MORTRAN source or FORTRAN source to create object code.

If the user understands MORTRAN, the source is well commented, compact, and has a structure that makes it generally easier to read than FORTRAN. In order to use this source, the user must have access to a MORTRAN processor and a FORTRAN compiler.

The FORTRAN object language generated by the MORTRAN processor, being machine generated, is uncommented, unstructured, and more cumbersome in some aspects than the human-generated FORTRAN source. However, it does provide greater portability to the model because it is totally independent of MORTRAN. Also, it is cheaper to convert the FORTRAN source code to object code, since the MORTRAN processor step is totally eliminated. Figure 15 shows the JCL required to convert MORTRAN and FORTRAN to object code on the IBM 370/158 computer at the Rand Computer Center.

Once the user has generated the object code from the FSM source code, he should create an FSM load module that can be executed directly without translating the source statements. Sample costs involved in creating the load module at the Rand Computer Center are shown in Table 5. Usually the costs of the MORTRAN processor are less than 25 percent of the FORTRAN step. These costs can be eliminated entirely on model reruns if regeneration of the load module is unneccessary and the model is rerun by executing the FSM load module.

```
//Job card
// EXEC MORTP,MACS=MFFSM,LEVEL=L,REGION=96K   Invokes the MORTRAN
      .                                         processor and generates
      .                                         the FORTRAN source
     MORTRAN Source
      .
      .

//EXEC FORTCL,REGC=140K,PARMC=SOURCE,                          Invokes
//     LIBL='k.k1632.ar012.KDLIB',PARML='XREF,LIST,OVLY'       FORTRAN
//SYSIN DD DSN=&&MORTFORT,DISP=(OLD,DELETE)                    compiler and
//DDDEL DD DSN=name of load module,DISP=(MOD,DELETE),          generates
// UNIT=USER,VOL=SER=PRIV01,SPACE=(TRK,(5,3,1))                object
//LKED.SYSLMOD DD DSN=name of load module,DISP=(,CATLG),       code which
// UNIT=USER,VOL=SER=PRIV01,SPACE=(TRK,(5,3,1))                is used to
//LKEDOLD DD DSN=k.k1632.ar012.FSMA(PROG),DISP=SHR             create the
 INCLUDE LKEDOLD(PROG)                                         FSM load
 ENTRY MAIN                                                    module.
 OVERLAY ONE
 INSERT DEF
 OVERLAY ONE
 INSERT PRTHMT
 INSERT CPINIT
 INSERT MKS
 INSERT PRT
 INSERT XVECT
// EXEC PGM=PROG,REGION=650K,COND=(4,LT)                       Invokes the
//STEPLIB DD DSN=name of load module,DISP=SHR                  execution of
//FT01F001 DD DSN=name of data base,DISP=SHR                   the FSM load
//FT02F001 DD DSN=name of output data set (optional)           Module
//FT05F001 DD *
    .
    .
    .
   FSM Input stream
    .
    .
    .
//
```

Fig.15 - JCL used to create and execute FSM load module

Table 5

REPRESENTATIVE COSTS OF LOAD MODULE PREPARATION

Step	Cost
MORTRAN Processor	$ 1.66
FORTRAN Compiler	7.00
Linkage Editor	1.92
	$10.58

Requirements involved in executing the load module are more difficult to determine in terms of necessary memory, execution time, and ultimate costs. Necessary memory is allocated dynamically at execution time by the model and depends on a variety of input runtime parameters. Memory (in number of bytes) required to hold the model data arrays is calculated as follows:

```
NCORE=8(NBCF)(NPTS+NPOL+NBCF) + 4(NPLANS+1)
      +(((NBCFMX+1)(NBCFMX/2))(4(NPOL)+(NPLANS)(NPOL))(2)
          (NPLANS)( 8(NBCFMX+1)(NBCFMX/2)+4)MODELS;
```

The definitions of the variables in the NCORE equation are given in Section VI.

The NCORE equation defines only the dynamically allocated data array section of the total memory requirement. The total memory necessary to run the model is shown in Table 6.

Table 6

CALCULATION OF MEMORY REQUIREMENTS

Memory Requirement Components	Size in Bytes
NCORE arrays	See NCORE equation
Load module	58K - 60K
I/O buffers for input data file	2*file blocksize

Execution time and cost are also closely dependent on the values of the parameters used in the NCORE equation (particularly the number of independent variables NBCF) and the actual build-up and build-down requests in the input stream that produce the total number of set

assignments during the run. Also, the search length parameter (MAXIT) can make a great deal of difference in execution time for a model run because it limits the number of searches involved when a set is being selected for a particular plan. If the number of sets to be selected is n, then the total number of sets evaluated by FSM through the cost effectiveness calculation will be bounded by the inequality

$$\text{number of CE} \leq n * \text{MAXIT calculations.}$$

Various model runs are tabulated in Table 7 to allow cost comparisons to be made under the varying memory and time requirements demanded by differing applications of the FSM.

The costs in Table 7 were obtained several years ago on an older computer than the Rand's current 370/158. More up-to-date figures would reflect cost reduction factors of between 2 and 4 for the 370/158 runs of the FSM.

Table 7

REPRESENTATIVE COSTS OF FSM RUNS

Run Number	Number of Samples Chosen	MODELS Value	NBCF Value	NBCFMX Value	NPOL Value	NPLANS Value	NPTS Value
1	1 FSM selection 1 Random selection	1	25	25	27	5	800
2	1 FSM selection	1	19	5	20	5	50
3	1 FSM selection 1 Random selection	7	25	4	27	32	1550
4	1 FSM selection 1 Random selection	1	25	25	27	12	2415
5	1 FSM Selection 1 Random Selection	1	25	25	27	12	2360

Run Number	NSETS Value	NCORE value	Sets Assigned	Search Length	Execution Time	Core (Region) Allocated	Cost
1	320	202752	320	10	112.95	300K Requested 276K Used	$17.59
2	50	27532	50	50	2.50	180K Requested	.68
3	625	413844	87	20	64.95	600K Requested 484K Used	10.20
4	915	567052	348	10	160.22	650K Requested 624K Used	24.86
5	913	555980	319	20	191.37	650K Requested 614K Used	29.35

Appendix B

AN APPLICATION OF THE FSM

In this application of the Finite Selection Model, four balanced groups are selected from the free pool of 48 states, using four independent variables: longitude, latitude, area and population. The 48 states are stratified on the basis of the existence of a seaport. For convenience, equal number of states were assigned to each strata; Illinois and Ohio were specified as seaport states to achieve this balance. The variables, area and population, were scaled to satisfy the requirements of regresion analysis.

The point vector (vs) for a given observation (state) is

vs_1 = 1.0, the constant term

vs_2 = Log(population)

vs_3 = area

vs_4 = latitude

vs_5 = logitude

The [X] matrix before transformation would be

$$[X] = \begin{bmatrix} x_{11} & \cdots\cdots & x_{15} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ x_{n1} & \cdots\cdots & x_{n5} \end{bmatrix}$$

where $x_{nk} = vs_k$ for the n^{th} observation.

The [X] matrix is transformed by subtracting the means of the independent variables from their respective elements in the [X] matrix; after transformation the elements of the [X] matrix would be defined as follows:

$$x_{nk} = vs_k - vs_k$$

The input routine, written in FORTRAN IV, and the JCL required to build the load module are shown in Fig. 16 (see page 68). Figure 17 on page 70 illustrates the input stream required to execute the FSM -- the necessary problem defintion and program command inputs. Figure 18 on page 74 is a listing of the data base.

Figure 19 is the output generated by the FSM. In this example three selections are generated. Initially the four groups of twelve states are selected by using a selection order matrix. To generate the second selection, there is a build-down step that reduces each group by half -- the least satisfactory six in each group are released to the free pool. The four groups are then built up once again to twelve. In the third selection, the 48 states are released to the free pool and the four groups are generated randomly (MAXIT=0). For comparison purposes, the matrices of normalized variances for the three selections can be found on pages 75, 79 and 81. Figure 20 is a map of the United States, illustrating the second and best selection.

The state labels at the end of Fig. 19 (VT, PA, ..., etc.) were hand typed. The FSM does not now have the ability to auto-matically print such labels.

```
//K1632R08 JOB (3743,50,185),'FSM.APPLIC',CLASS=F
// EXEC FORTCL,REGC=140K,PARMC=SOURCE,PARML='XREF,LIST,MAP,OVLY',
//      LIBL='k.k1632.ar012.KDLIB'
//FORT.SYSIN DD *
      SUBROUTINE DEF(D,P,X,XT,V,VS,VSTD,S,Q0,ZS,LABS,PNSETS,PINCR,PPTR,
     &PTEMP,PINCBE,PITER,SCOST,SSIZE,SFP,SINCR,SID,SPTR,STEMP,SPLAN,
     &SSTRAT,WP,EVARA,WTEMP,WM,HMAT1,HMAT2)
      LOGICAL FLGW,FLGID,FLGD,FLGM,FLGA,FLGDB,FLGSF
      REAL*8 TOP,BOTTOM,T,TT,DET
      COMMON TOP,BOTTOM,EFF,NBCF,INSERT,NPOL,NBCFX,NPOLX,NLTX,NPLANS,
     &NPTS,NSETS,MODELS,MAXIT,NPZERO,NIT,FLGW,FLGID,FLGD,FLGM,FLGA,
     &FLGDB,FLGSF,SMULT,NP0,COST0,WMSUM
      INTEGER PNSETS(NPZERO),PPTR(NPZERO),PTEMP(NPZERO)
      INTEGER PINCR(NPLANS),PINCBE(NPLANS),PITER(NPLANS)
      INTEGER SCOST(NSETS),SSIZE(NSETS),SFP(NSETS),SINCR(NSETS),
     &SID(NSETS),SPTR(NSETS),STEMP(NSETS),SPLAN(NSETS),SSTRAT(NSETS)
      REAL*8 D(NLTX,MODELS),Q0(NLTX,MODELS),S(NLTX,MODELS,NPLANS),
     &X(NBCFX,NPTS)
      REAL*8 ZS(NBCFX,NBCFX),LABS(NPOLX)
      REAL*8 XT(NBCFX),V(NBCFX),VS(NBCFX),VSTD(NBCFX),P(NBCFX,NPOLX)
      INTEGER WP(NPOLX,MODELS),WM(MODELS)
      REAL HMAT1(NPOLX,NPLANS,MODELS),HMAT2(NPOLX,NPLANS,MODELS)
      REAL WTEMP(NPOLX),EVARA(NPOLX,MODELS)
C***********************************************************************
      REAL*4 POP70,AREA,LAT,LONG,CPORT
      REAL*8 NAME
C***********************************************************************
C
C            AN APPLICATION OF THE FINITE SELECTION MODEL
C
C     FOUR BALANCED GROUPS OF STATES ARE SELECTED FROM THE FINITE
C     GROUP OF 48 STATES ON THE BASIS OF POPULATION, AREA, LATITUDE,
C     AND LONGITUDE.
C***********************************************************************
C               INPUT ROUTINE  -  MAY,1976
C***********************************************************************
C     INITIALIZE COUNTERS AND VARIABLES
C
C     NPTS.........THE NO. OF POINTS
C     NSETS........THE NO. OF SETS
C     NBCF.........NO.OF INDEPENDENT VARIABLES (SEE PROBLEM DEF.INPUTS)
C     VS(NBCF).....A POINT VECTOR
C     V(NBCF)......VALUE OF EACH VARIABLE IN THE POINT VECTOR AVERAGED
C                  OVER THE TOTAL NO.OF POINTS IN THE FREE POOL (NPTS)
C***********************************************************************
      NPTS = 0
      NSETS = 0
      DO 10 I = 1,NBCF
      V(I) = 0.0
10    VS(I)= 0.0
```

Fig. 16 - FSM input routine

```
C*************************************************************************
C     READ RECORD FROM INPUT FILE
C*************************************************************************
C     DESCRIPTION OF INPUT RECORD
C
C        CC        DESCRIPTION
C
C      1 - 2       SET ID
C      3 -10       STATE NAME
C     11 -16       POPULATION (1970)
C     17 -22       AREA
C     23 -28       LATITUDE
C     29 -34       LONGITUDE
C     35 -40       SEAPORT (=1, YES; =0, NO;)
C*************************************************************************
20    READ (1,30,END=100) ID,NAME,POP70,AREA,LAT,LONG,CPORT
30    FORMAT (I2,A8,5F6.0)
      WRITE (6,35) ID,NAME,POP70,AREA,LAT,LONG,CPORT
35    FORMAT ( ' INPUT ',I4,A8,5F10.4)
C*************************************************************************
C     INITIALIZE SET VARIABLES
C*************************************************************************
      NSETS = NSETS + 1
      SCOST(NSETS) = 1
      SSIZE(NSETS) = 1
      SID(NSETS)   = ID
      SINCR(NSETS) = 1
      SPLAN(NSETS) = 0
      SSTRAT(NSETS)= 1.0 + CPORT
CC************************************************************************
C     SET UP POINT VECTOR; ELEMENTS MUST BE IN THE SAME ORDER AS
C     POLICY LABEL INPUTS. SCALE SELECTED VARIABLES.
C*************************************************************************
      NPTS = NPTS + 1
      VS(1) = 1.0
      VS(2) = ALOG(POP70)
      VS(3) = SQRT(AREA)
      VS(4) = LAT
      VS(5) = LONG
C*************************************************************************
C     CREATE NEXT ROW IN [X] MATRIX AND ADD POINT VECTOR TO SUMMATION
C     VECTOR;  PROCESS NEXT RECORD.
C*************************************************************************
      DO 40 I = 1,NBCF
      X(I,NPTS) = VS(I)
40    V(I) = V(I) + VS(I)
C
      GO TO 20
C*************************************************************************
```

Fig. 16 - FSM input routine - continued

```
C*****************************************************************************
C     AFTER ALL DATA HAS BEEN READ AND PROCESSED, PRINT 10 OBSERVATIONS
C     FROM THE [X] MATRIX.
C*****************************************************************************
100   WRITE (6,110)
110   FORMAT(' [X] MATRIX, 10 OBSERVATIONS BEFORE TRANSFORMATION ')
      CALL PRTMAT (X,NBCF,10)
C*****************************************************************************
C     COMPUTE AVERAGE FOR EACH VARIABLE AND SUBTRACT FROM ELEMENTS
C     IN THE [X] MATRIX
C*****************************************************************************
      T = 1.0/NPTS
      DO 120 I = 1,NBCF
120   V(I) = V(I)*T
C
      DO 130 I = 1,NPTS
      DO 130 J = 2,NBCF
130   X(J,I) = X(J,I) - V(J)
C*****************************************************************************
C     PRINT 10 OBSERVATIONS FROM [X] MATRIX AFTER TRANSFORMATION
C*****************************************************************************
      WRITE (6,140)
140   FORMAT(' [X] MATRIX, 10 OBSERVATIONS AFTER TRANSFORMATION ')
      CALL PRTMAT (X,NBCF,10)
C*****************************************************************************
C     SET UP POLICY MATRIX
C*****************************************************************************
      DO 150 I = 1,NBCF
      DO 150 J = 1,NBCF
150   P(I,J) = 0.0
C
      DO 160 I = 1,NBCF
160   P(I,I) = 1.0
      REWIND 1
      RETURN
      END
//LKED.SYSLMOD DD DSN=K.K1632.AR012.FSMY(PROG),DISP=(,CATLG),
// UNIT=USER,VOL=SER=PRIV01,SPACE=(TRK,(5,3,1))
//LKEDOLD DD DSN=K.K1632.AR012.FSMA(PROG),DISP=SHR
  INCLUDE LKEDOLD(PROG)
  ENTRY MAIN
  OVERLAY ONE
  INSERT DEF
  OVERLAY ONE
  INSERT PRTHMT
  INSERT CPINIT
  INSERT MKS
  INSERT PRT
  INSERT XVECT
/*
```

Fig. 16 - FSM input routine - continued

```
//K1632R08 JOB (3743,50,185),'FSM.LOAD',CLASS=F
// EXEC PGM=PROG,REGION=300K,COND=(4,LT)
//STEPLIB DD DSN=K.K1632.AR012.FSMY,DISP=SHR
//FT01F001 DD DSN=K.K1632.AR012.STATES48,DISP=SHR
//FT05F001 DD *
 &IN NBCF=5,NPOL=5,NPLANS=4,NPTS=48,NSETS=48,MAXIT=10,MODELS=1,
 NBCFMX=5,FLGID=T,FLGW=T,FLGM=T,FLGD=T,FLGDB=T,&END
CONST    100     1
POP70    100     2
AREA     100     3
LAT      100     4
LONG     100     5
MODEL  1
RUN  001   12   12   12   12
RANDOM
MANUAL
   0  4  2
   0  3  1
   0  2  2
   0  1  1
   0  1  2
   0  2  1
   0  3  2
   0  4  1
   0  3  2
   0  4  1
   0  1  2
   0  2  1
   0  2  2
   0  1  1
   0  4  2
   0  3  1
   0  1  1
   0  4  2
   0  2  2
   0  3  1
   0  3  2
   0  2  1
   0  4  1
   0  1  2
   0  2  1
   0  3  2
   0  1  2
   0  4  1
   0  4  2
   0  1  1
   0  3  1
   0  2  2
   0  2  2
   0  4  2
```

Fig. 17 - FSM input stream

```
    0  3  1
    0  1  1
    0  1  2
    0  3  2
    0  4  1
    0  2  1
    0  3  2
    0  1  2
    0  2  1
    0  4  1
    0  4  2
    0  2  2
    0  1  1
    0  3  1

PRINT
MANUAL
    1  0  1
    1  0  2
    1  0  2
    1  0  1
    1  0  1
    1  0  2
    2  0  1
    2  0  2
    2  0  2
    2  0  1
    2  0  1
    2  0  2
    3  0  1
    3  0  2
    3  0  2
    3  0  1
    3  0  1
    3  0  2
    4  0  1
    4  0  2
    4  0  2
    4  0  1
    4  0  1
    4  0  2
    0  1  2
    0  3  1
    0  2  1
    0  4  2
    0  4  1
    0  2  2
    0  3  2
    0  1  1
    0  3  2
```

Fig. 17 - FSM input stream - continued

```
   0  1  1
   0  4  1
   0  2  2
   0  2  1
   0  4  2
   0  1  2
   0  3  1
   0  4  2
   0  1  2
   0  2  1
   0  3  1
   0  3  1
   0  3  2
   0  2  2
   0  1  1
   0  4  1

PRINT
MAXIT  0
LOCAL  0    0    0    0    0
RANDOM
MANUAL
   0  4  2
   0  3  1
   0  2  2
   0  1  1
   0  1  2
   0  2  1
   0  3  2
   0  4  1
   0  3  2
   0  4  1
   0  1  2
   0  2  1
   0  2  2
   0  1  1
   0  4  2
   0  3  1
   0  1  1
   0  4  2
   0  2  2
   0  3  1
   0  3  2
   0  2  1
   0  4  1
   0  1  2
   0  2  1
   0  3  2
   0  1  2
   0  4  1
```

Fig. 17 - FSM input stream - continued

```
    0  4  2
    0  1  1
    0  3  1
    0  2  2
    0  2  2
    0  4  2
    0  3  1
    0  1  1
    0  1  2
    0  3  2
    0  4  1
    0  2  1
    0  3  2
    0  1  2
    0  2  1
    0  4  1
    0  4  2
    0  2  2
    0  1  1
    0  3  1

PRINT
//FT06F001 DD SYSOUT=A
//
```

Fig. 17 - FSM input stream - continued

Id	Name	Population in thousands	Area in thousands of sq.mi.	Lat.	Long.	Seaport
01	ME	997.	30.9	44.43	69.62	1
02	NH	742.	9.0	43.18	71.47	1
03	VT	446.	9.3	44.03	72.8	0
04	MASS	5706.	7.8	42.28	71.38	1
05	RI	951.	1.0	41.75	71.43	1
06	CONN	3041.	4.9	41.48	72.87	1
07	NY	18268.	47.8	41.5	74.72	1
08	NJ	7193.	7.5	40.48	74.4	1
09	PA	11813.	45.0	40.45	77.18	1
10	OHIO	10664.	41.0	40.53	82.67	1
11	IND	5202.	36.1	40.18	86.28	0
12	ILL	11128.	55.7	41.22	88.38	0
13	MICH	8890.	56.8	42.8	84.03	1
14	WIS	4429.	54.5	43.67	88.92	0
15	MIN	3815.	79.3	45.23	93.65	0
16	IOWA	2832.	55.9	41.98	93.08	0
17	MO	4688.	69.	38.52	92.02	0
18	ND	620.	69.3	47.45	99.5	0
19	SD	668.	76.	44.13	98.95	0
20	NEB	1488.	76.5	41.18	92.53	0
21	KAN	2249.	81.8	38.45	96.65	0
22	DEL	551.	2.0	39.47	75.55	1
23	MD	3938.	9.9	39.15	76.78	1
24	VA	4659.	39.8	37.6	77.97	1
25	WV	1751.	24.1	38.72	80.97	0
26	NC	5098.	48.8	35.57	79.67	1
27	SC	2597.	30.2	34.05	81.03	1
28	GA	4607.	58.1	33.13	83.8	1
29	FL	6848.	54.1	27.85	81.72	1
30	KY	3231.	39.7	37.8	85.27	0
31	TEN	3937.	41.3	35.75	86.5	0
32	ALA	3451.	50.7	32.98	86.77	1
33	MISS	2220.	47.3	32.6	89.65	1
34	ARK	1932.	51.9	34.9	92.35	0
35	LA	3652.	44.9	30.72	91.47	1
36	OK	2567.	68.8	35.58	96.93	0
37	TX	11236.	262.1	31.07	97.52	1
38	MON	698.	145.6	46.88	10.82	0
39	IDA	718.	82.7	44.2	14.82	0
40	WYO	334.	97.2	42.67	06.77	0
41	COL	2223.	103.8	39.43	05.12	0
42	NM	1023.	121.4	34.57	06.1	0
43	ARIZ	1792.	113.4	33.28	11.68	0
44	UTAH	1066.	82.1	40.55	11.82	0
45	NEV	493.	109.9	37.63	16.75	0
46	WAS	3413.	66.6	47.35	21.52	1
47	ORE	2101.	96.2	44.73	22.55	1
48	CAL	20007.	156.4	35.47	19.47	1

Fig. 18 - Data base

```
FINITE SELECTION MODEL OF 22 NOV 75 ** WEDNESDAY, JULY 21, 1976
&IN
NBCF=          5,NBCFMX=          5,NPOL=          5,NPLANS=      4,
MODELS=          1,NSETS=        48,NPTS=         48,
FLGW=T,FLGID=T,FLGD=T,FLGM=T,FLGA=T,FLGDB=T,FLGSF=F,MAXIT=       10,
NCORE=       5312
&END

SEQ  NAME    WEIGHTS
  1 CONST      100
  2 POP70      100
  3 AREA       100
  4 LAT        100
  5 LONG       100

THE NUMBER OF SETS INPUT:      48

THE NUMBER OF POINTS INPUT:    48

SEQ  LABEL       DEF-MEAN        MEAN            STD
   1 CONST      0.9999999       1.000000        0.3725290D-08
   2 POP70       7.868913      0.4690238D-06     1.014841
   3 AREA        7.282001      0.4340412D-06     2.953730
   4 LAT         39.34624      0.2345219D-05     4.605330
   5 LONG        90.91457      0.5418931D-05     14.76103

*INPUT* MODEL    1
*INPUT* RUN     001      12       12       12       12

NUMBER OF SETS      48
NUMBER OF INCR      48
NUMBER OF POINTS    48   48
AVE COST PER INCR    1.000000


INITIAL COMPUTATIONS FOR MODEL  1

   THE NUMBER OF BETA COEFFICIENTS:     5
   THE NUMBER OF POLICIES:              5

   CORRELATION MATRIX OF X

                  CONST     POP70     AREA      LAT       LONG
    1  CONST    1.000000
    2  POP70    0.000000  1.000000
    3  AREA     0.000000  0.069297  1.000000
    4  LAT      0.000000 -0.321259 -0.162328  1.000000
    5  LONG     0.000000 -0.220237  0.782525  0.076358  1.000000
```

Fig. 19a - FSM output for selection 1

DETERMINANT FOR Q0 IS 20924.84

Q0 MATRIX

1	0.114D 01				
2	0.536D-06	0.118D 01			
3	0.496D-06	0.237D 00	0.997D 01		
4	0.268D-05	-0.172D 01	-0.252D 01	0.242D 02	
5	0.619D-05	-0.377D 01	0.390D 02	0.593D 01	0.249D 03

S0 MATRIX

1	0.875D 00				
2	-0.793D-06	0.110D 01			
3	0.169D-06	-0.195D 00	0.331D 00		
4	-0.121D-06	0.466D-01	0.343D-01	0.496D-01	
5	-0.573D-07	0.461D-01	-0.556D-01	-0.584D-02	0.136D-01

Q0*S0

ROW FIRST COL 1, LAST COL 5

1	0.100D 01	-0.529D-22	-0.847D-21	-0.569D-21	0.0
2	-0.529D-22	0.100D 01	0.888D-15	0.971D-16	0.888D-15
3	-0.132D-22	0.139D-16	0.100D 01	0.971D-16	0.444D-15
4	0.265D-22	-0.304D-16	-0.971D-16	0.100D 01	-0.222D-15
5	0.132D-22	0.260D-17	0.153D-15	-0.139D-16	0.100D 01

POL	VAR-NAME	W(POL)	VAR(A0(POL))	POLICY VECTOR
1	CONST	100	0.7083334	
2	POP70	100	1.104684	
3	AREA	100	0.3310848	
4	LAT	100	0.4955480E-01	
5	LONG	100	0.1356166E-01	

PLAN SIZES: 12 12 12 12

STRATA	SETS	INCR	PTS	AVE COST
1	24	24	24	1.00
2	24	24	24	1.00

PLAN	MODEL	AVAIL SETS	INCR	INCR-TO-BE	DET-Q
1	1	0	0	12	0.5206765D-05
2	1	0	0	12	0.5206765D-05
3	1	0	0	12	0.5206765D-05
4	1	0	0	12	0.5206765D-05

INPUT RANDOM

RANDOM SEED IS: 1175534387

Fig. 19a - FSM output for selection 1 - continued

```
*INPUT* PRINT

PLAN     1 *************************
      SETS  INCR   PTS       COST   AVE COST
 OLD     0     0     0          0  0.0
 NEW    12    12    12         12   1.000000
 TOT    12    12    12         12   1.000000

PHI-NORM   0.9429667     EFFCY    1.060482     PHI    0.6629425E-01

Z-TEST   ***************   0.3237323  -1.1001067  -1.0073616  -1.2174939

STND-DEVS      2.0000000   1.0385402   0.8799058   0.9431554   0.8539575

SELECTED SETS   3    9   11   12   20   22   23   24   28   29   43   44
               VT   PA  IND  ILL  NEB  DEL   MD   VA   GA   FL ARIZ UTAH

PLAN     2 *************************
      SETS  INCR   PTS       COST   AVE COST
 OLD     0     0     0          0  0.0
 NEW    12    12    12         12   1.000000
 TOT    12    12    12         12   1.000000

PHI-NORM   0.5734521     EFFCY    1.743824     PHI    0.4699044E-01

Z-TEST   ***************  -0.2293727  -0.7596340   0.8347575  -0.2157446

STND-DEVS      2.0000000   0.9183838   0.9669321   1.1070158   1.0220426

SELECTED SETS   2    4    5    7   15   18   21   31   35   36   42   46
               NH MASS   RI   NY  MIN   ND  KAN  TEN   LA   OK   NM  WAS

PLAN     3 *************************
      SETS  INCR   PTS       COST   AVE COST
 OLD     0     0     0          0  0.0
 NEW    12    12    12         12   1.000000
 TOT    12    12    12         12   1.000000

PHI-NORM   0.8538593     EFFCY    1.171153     PHI    0.6353366E-01

Z-TEST   ***************  -0.7488827   1.0913131   0.8318910   1.1468849

STND-DEVS      2.0000000   0.8945565   0.7914601   0.9564834   0.9213183

SELECTED SETS   6   13   16   17   19   26   32   33   38   40   41   47
             CONN MICH IOWA   MO   SD   NC  ALA MISS  MON  WYO  COL  ORE
```

Fig. 19a - FSM output for selection 1 - continued

PLAN 4 *************************

	SETS	INCR	PTS	COST	AVE COST
OLD	0	0	0	0	0.0
NEW	12	12	12	12	1.000000
TOT	12	12	12	12	1.000000

PHI-NORM 0.5580537 EFFCY 1.791942 PHI 0.4045242E-01

Z-TEST *************** 0.6545303 0.7684300 -0.6592788 0.2863594

STND-DEVS 2.0000000 1.0975353 1.1985858 0.8882263 1.0911372

SELECTED SETS	1	8	10	14	25	27	30	34	37	39	45	48
	ME	NJ	OHIO	WIS	WV	SC	KY	ARK	TX	IDA	NEV	CAL

MATRIX OF UNNORMALIZED VARIANCES

PLAN	CONST	POP70	AREA	LAT	LONG	MEAN	STD
1	0.106461	0.147972	0.068980	0.005493	0.002565	0.066294	0.056657
2	0.090062	0.106818	0.033299	0.003592	0.001181	0.046990	0.043827
3	0.093940	0.161427	0.054027	0.006248	0.002027	0.063534	0.059435
4	0.087260	0.086551	0.020202	0.007467	0.000783	0.040452	0.038439
MEAN	0.094431	0.125692	0.044127	0.005700	0.001639	0.054318	0.048879
STD	0.007339	0.030256	0.018744	0.001406	0.000698	0.010889	0.016267

MATRIX OF NORMALIZED VARIANCES

PLAN	CONST	POP70	AREA	LAT	LONG	MEAN	STD
1	0.853392	0.805307	1.252568	0.666452	1.137120	0.942967	0.217708
2	0.721940	0.581334	0.604656	0.435792	0.523539	0.573452	0.094368
3	0.753019	0.878531	0.981048	0.758014	0.898688	0.853860	0.087352
4	0.699476	0.471035	0.366831	0.905871	0.347058	0.558054	0.214305
MEAN	0.756956	0.684052	0.801275	0.691532	0.726601	0.732083	0.043332
STD	0.058835	0.164661	0.340362	0.170583	0.309608	0.169375	0.232971

Fig. 19a - FSM output for selection 1 - continued

```
*INPUT* PRINT

PLAN      1 ***********************
       SETS   INCR    PTS        COST   AVE COST
 OLD      0      0      0           0  0.0
 NEW     12     12     12          12   1.000000
 TOT     12     12     12          12   1.000000

PHI-NORM    0.9142929      EFFCY      1.093741      PHI     0.5492642E-01

Z-TEST    ***************    0.7458307   -0.2365507   -0.1350055   -0.3860169

STND-DEVS       2.0000000    1.0920295    0.9770620    0.9325900    1.0398458

SELECTED SETS    3    4     9    11    19    20    23    26    29    41    44    48
                VT MASS    PA   IND    SD   NEB    MD    NC    FL   COL  UTAH   CAL

PLAN      2 ***********************
       SETS   INCR    PTS        COST   AVE COST
 OLD      0      0      0           0  0.0
 NEW     12     12     12          12   1.000000
 TOT     12     12     12          12   1.000000

PHI-NORM    0.6147553      EFFCY      1.626663      PHI     0.5155688E-01

Z-TEST    ***************   -0.7171593   -0.3251449   -0.0232733    0.3156320

STND-DEVS       2.0000000    0.9171915    1.0258099    1.1155080    1.0177511

SELECTED SETS    5     7    15    18    21    22    27    31    32    42    43    46
                RI    NY   MIN    ND   KAN   DEL    SC   TEN   ALA    NM  ARIZ   WAS

PLAN      3 ***********************
       SETS   INCR    PTS        COST   AVE COST
 OLD      0      0      0           0  0.0
 NEW     12     12     12          12   1.000000
 TOT     12     12     12          12   1.000000

PHI-NORM    0.7945634      EFFCY      1.258553      PHI     0.5582284E-01

Z-TEST    ***************    0.1205047    0.4334995    0.2703728    0.3527249

STND-DEVS       2.0000000    0.9355173    0.8140381    0.9256215    0.9506916

SELECTED SETS    6    12    13    17    24    25    28    33    36    38    40    47
              CONN   ILL  MICH    MO    VA    WV    GA  MISS    OK   MON   WYO   ORE
```

Fig. 19b - FSM output for selection 2

PLAN 4 **************************

	SETS	INCR	PTS	COST	AVE COST
OLD	0	0	0	0	0.0
NEW	12	12	12	12	1.000000
TOT	12	12	12	12	1.000000

PHI-NORM 0.5805639 EFFCY 1.722463 PHI 0.4794985E-01

Z-TEST **************** -0.1491689 0.1281984 -0.1120860 -0.2823341

STND-DEVS 2.0000000 1.0101269 1.1432595 1.0111759 0.9747766

SELECTED SETS	1	2	8	10	14	16	30	34	35	37	39	45
	ME	NH	NJ	OHIO	WIS	IOWA	KY	ARK	LA	TX	IDA	NEV

MATRIX OF UNNORMALIZED VARIANCES

PLAN	CONST	POP70	AREA	LAT	LONG	MEAN	STD
1	0.086870	0.106272	0.071156	0.008003	0.002332	0.054926	0.042162
2	0.089125	0.129825	0.033773	0.003740	0.001321	0.051557	0.050326
3	0.085662	0.134226	0.051079	0.006160	0.001987	0.055823	0.049849
4	0.086136	0.127527	0.019184	0.005869	0.001033	0.047950	0.050180
MEAN	0.086948	0.124463	0.043798	0.005943	0.001668	0.052564	0.047313
STD	0.001332	0.010775	0.019415	0.001512	0.000516	0.003103	0.009974

MATRIX OF NORMALIZED VARIANCES

PLAN	CONST	POP70	AREA	LAT	LONG	MEAN	STD
1	0.696352	0.578361	1.292077	0.970890	1.033788	0.914293	0.253201
2	0.714428	0.706543	0.613273	0.453711	0.585823	0.614755	0.095010
3	0.686668	0.730497	0.927517	0.747377	0.880760	0.794563	0.092823
4	0.690462	0.694040	0.348360	0.712051	0.457908	0.580564	0.149135
MEAN	0.696977	0.677360	0.795306	0.721007	0.739570	0.726044	0.040549
STD	0.010650	0.058637	0.352547	0.183483	0.228839	0.135723	0.206821

Fig. 19b - FSM output for selection 2 - continued

```
*INPUT* MAXIT    0

PLAN     1 *************************
      SETS  INCR   PTS       COST   AVE COST
 OLD    0     0     0           0  0.0
 NEW   12    12    12          12   1.000000
 TOT   12    12    12          12   1.000000

PHI-NORM     1.345613      EFFCY   0.7431554      PHI    0.1225169

Z-TEST    ***************  -0.3753461   1.3682200  -2.5193013   0.0915062

STND-DEVS       2.0000000   1.0761048   1.0544951   0.9593690   0.8793636

SELECTED SETS    3    17    24    26    27    29    32    36    37    40    41   45
                VT    MO    VA    NC    SC    FL   ALA    OK    TX   WYO   COL  NEV

PLAN     2 *************************
      SETS  INCR   PTS       COST   AVE COST
 OLD    0     0     0           0  0.0
 NEW   12    12    12          12   1.000000
 TOT   12    12    12          12   1.000000

PHI-NORM   0.6638227      EFFCY    1.506426      PHI    0.5197546E-01

Z-TEST    ***************  -0.9909526   0.0747141   0.5246310   0.4850106

STND-DEVS       2.0000000   0.8945696   0.8849388   0.9669712   1.0152546

SELECTED SETS    1    10    12    20    21    22    25    33    35    38    44   47
                ME  OHIO   ILL   NEB   KAN   DEL    WV  MISS    LA   MON  UTAH  ORE

PLAN     3 *************************
      SETS  INCR   PTS       COST   AVE COST
 OLD    0     0     0           0  0.0
 NEW   12    12    12          12   1.000000
 TOT   12    12    12          12   1.000000

PHI-NORM     2.630330      EFFCY   0.3801804      PHI    0.1493002

Z-TEST    ***************   0.3111342  -1.5049829   0.7925008  -1.5305540

STND-DEVS       2.0000000   0.9860463   0.9962640   0.9085692   0.7781922

SELECTED SETS    4     5     7     8    15    18    19    23    28    30    31   42
              MASS    RI    NY    NJ   MIN    ND    SD    MD    GA    KY   TEN   NM
```

Fig. 19c - FSM output for selection 3

PLAN 4 **************************

	SETS	INCR	PTS	COST	AVE COST
OLD	0	0	0	0	0.0
NEW	12	12	12	12	1.000000
TOT	12	12	12	12	1.000000

PHI-NORM 0.9812235 EFFCY 1.019135 PHI 0.7008731E-01

Z-TEST *************** 1.0551719 0.0620512 1.2021775 0.9540430

STND-DEVS 2.0000000 0.9594463 0.9207147 0.8741018 1.1692818

SELECTED SETS	2	6	9	11	13	14	16	34	39	43	46	48
	NH	CONN	PA	IND	MICH	WIS	IOWA	ARK	IDA	ARIZ	WAS	CAL

MATRIX OF UNNORMALIZED VARIANCES

PLAN	CONST	POP70	AREA	LAT	LONG	MEAN	STD
1	0.251416	0.310555	0.037423	0.010844	0.002348	0.122517	0.131245
2	0.091522	0.119822	0.041875	0.005504	0.001155	0.051975	0.046928
3	0.152809	0.431697	0.142380	0.005210	0.014406	0.149300	0.154121
4	0.119855	0.136900	0.084177	0.007864	0.001641	0.070087	0.056028
MEAN	0.153901	0.249743	0.076463	0.007356	0.004887	0.098470	0.093274
STD	0.060333	0.128859	0.042205	0.002261	0.005512	0.039145	0.066425

MATRIX OF NORMALIZED VARIANCES

PLAN	CONST	POP70	AREA	LAT	LONG	MEAN	STD
1	2.011326	1.686752	0.678181	1.312931	1.038889	1.345614	0.469078
2	0.732179	0.650801	0.758863	0.666442	0.510832	0.663823	0.086358
3	1.222471	2.344728	2.580239	0.630812	6.373407	2.630328	2.003740
4	0.958840	0.743560	1.525477	0.952177	0.726068	0.981224	0.289532
MEAN	1.231204	1.356460	1.385690	0.890590	2.162298	1.405248	0.417385
STD	0.482669	0.699889	0.764849	0.273812	2.438522	0.747308	1.210716

Fig. 19c - FSM output for selection 3 - continued

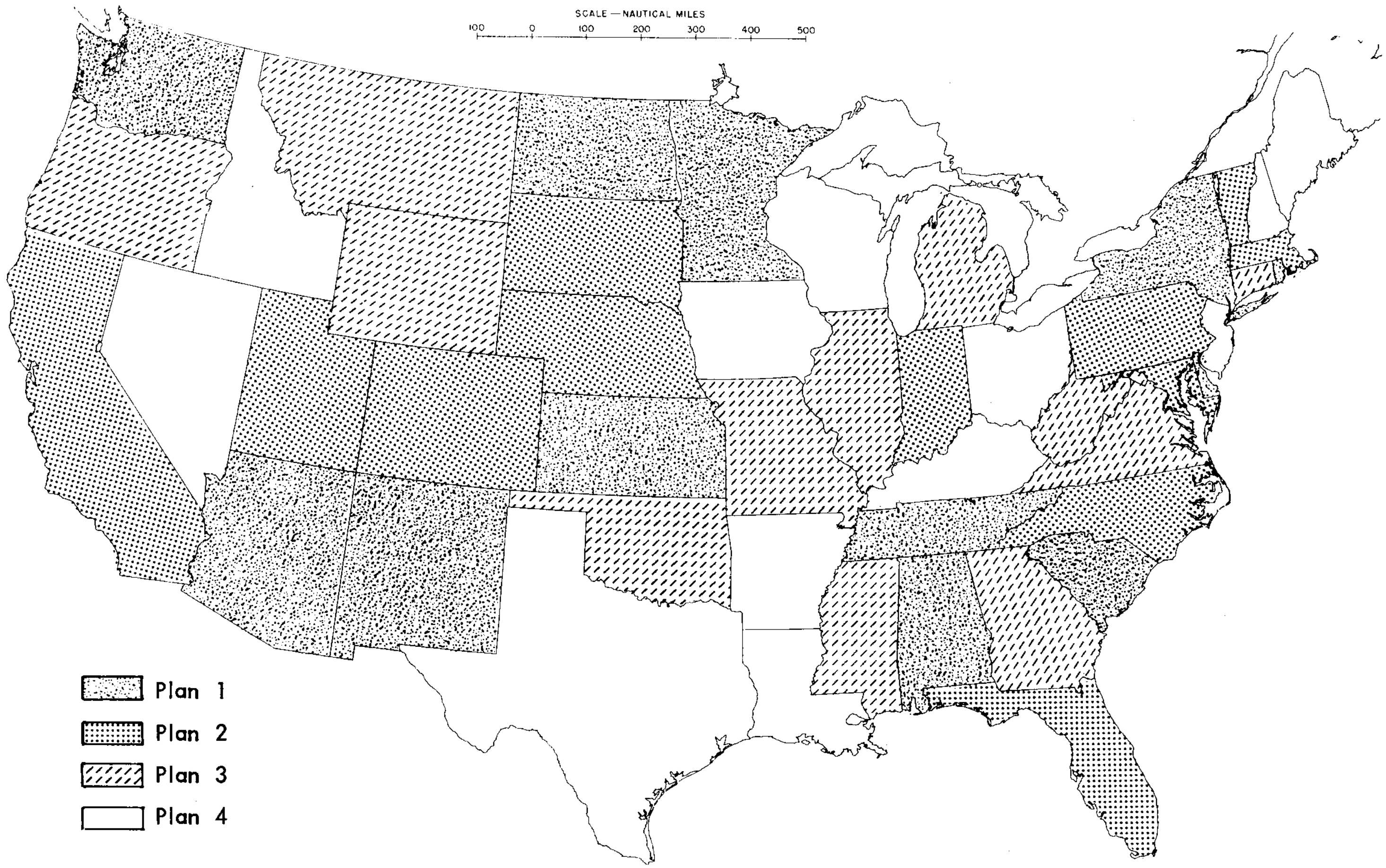

Fig.20— Graphic representation of selection 2

REFERENCES

1. Conlisk, John, and Harold Watts, "A Model for Optimizing Experimental Designs for Estimating Response Surfaces," Proceedings of the Social Statistics Sections, American Statistical Association, 1969.

2. Morris, Carl, "A Finite Selection Model for Experimental Design of the Health Insurance Study" in Proceedings of the Social Statistics Sections, American Statistical Association, 1975; and in the Journal of Econometrics (11), Sept.1979.

3. Newhouse, Joseph P., The Health Insurance Study -- A Summary, The Rand Corporation, R-965-1-OEO, March 1974.

RAND/N-1183-HEW