

HYDROLOGIC MODEL SELECTION IN A
DECISION MAKING CONTEXT
by
Robert Edmund Lovell

Reports on Natural Resource Systems<br>Report No. 26

June 1975

# Collaborative effort between the following Departments: <br> Hydrology and Water Resources <br> Systems and Industrial Engineering 

The University of Arizona Tucson, Arizona 85721

In memory of Chester C. Kisiel

This report constitutes the Doctor of Philosophy dissertation of the same title completed by the author in May 1975, and accepted by the Department of Systems and Industrial Engineering. It is the result of a joint Title II research project on model choice in hydrology No. 14-31-0001-3708, supported in part by the United States Department of the Interior, Office of Water Resources Research, as authorized under the Water Resources Research Act of 1964. The late Chester C. Kisiel provided fundamental contributions in initiating and co-directing this research.

Lucien Duckstein<br>Jointly with the Departments of Systems \& Industrial Engineering and Hydrology \& Water Resources

## ACKNOWLEDGMENTS

This paper has required the guidance, help and indulgence of many people. Professor Lucien Duckstein guided the entire dissertation and was willing to give me very fast turn-around on my rather slow production.

The late Dr. Chester C. Kisiel was constantly helpful in giving me the hydrologic insights that were needed and in guiding me to much of the related literature in both the systems and hydrologic aspects of the work.

Dr. A. Wayne Wymore's concepts provided the framework for all of the thinking that went into the development of the CHOOSE model and the associated CHOOSE system. Our interaction during seminars and during his review of the early versions of Chapters 3 and 4 were crucial.

Many of the formal optimization concepts that lie behind this work were learned in courses under Dr. Robert L. Baker. I am encouraged by his interest in an immediate application of the CHOOSE method in selecting computer systems.

Dr. Donald G. Schultz took over on very short notice as a committee member in the absence of Dr . Wymore and made valuable comments which could not have been made by those who worked closely with me during the past three years.

All of the faculty and students of both the Systems and Industrial Engineering Department and the Hydrology and Water Resources Department of the University of Arizona were willing to help whenever asked.

My sincere thanks to all of these people.
I would also like to thank my favorite wife, Emily, for her tolerance and encouragement, and for editing the entire manuscript. I would also like to thank my favorite niece, Rosalind, who provided additional encouragement and sometimes comic relief from the rigors of the writing of this paper.

Special thanks to Esther Taylor for transforming a mixture of poor handwriting and mumbled dictation into an uncommonly accurate and presentable final copy, and for accommodating my very erratic working hours. With her help the completion was advanced by perhaps six months.

The research associateship awarded jointly by the Systems and Industrial Engineering Department and the Hydrology and Water Resources Department was essential to this activity.

This study was supported principally by funds from the Title II research grant on model choice questions in hydrology and water resources by the Office of Water Resources Research, U.S. Department of the Interior.

## TABLE OF CONTENTS

Page
LIST OF TABLES ..... viii
LIST OF ILLUSTRATIONS ..... ix
ABSTRACT ..... x

1. INTRODUCTION ..... 1
Models ..... 2
The Overall Modelling Problem ..... 6
Preview of Remaining Chapters ..... 8
Specialized Notation and Definitions ..... 8
2. ORIGINS OF THE MODEL CHOICE PROBLEM ..... 13
The Model Overchoice Problem ..... 15
Some Model Choice Schemes ..... 16
Model Structure Choice in Hydrology ..... 19
Design Considerations for the CHOOSE System ..... 22
3. DESIGNING THE CHOOSE MODEL ..... 24
A Set Theoretic Approach to the Choice Problem ..... 24
Mathematical Programming Models ..... 26
The Mathematical Programming, CHOICEMODEL, $M$ ..... 28
Additional Considerations in MultipleCriteria Problems31
Information Retrieval Models ..... 33
A CHOICEMODEL I for Information Retrieval ..... 33
Moving Towards CHOOSE ..... 35
Aspiration Levels, Satisfactums, and Goal Programming ..... 36
Candidates, Attributes and Objectives ..... 36
CHOOSE vs. Information Retrieval ..... 39
CHOOSE vs. Mathematical Programming. ..... 40
Selecting the Directions for Compromise ..... 41
Value Judgments ..... 41
Decoupling or Decomposition ..... 43
The Resulting Feasible Region ..... 45
The Existence of Feasible Solutions ..... 48
A Note on Dominance ..... 51
A Prototype CHOOSE Model ..... 52
Additional CHOOSE Considerations ..... 54
Relationship to Fuzzy Sets and Approximation Spaces ..... 59
The CHOOSE Model ..... 60
4. IMPLEMENTING CHOOSE MODELS ..... 63
The Technology ..... 63
Division of Man/Machine Responsibilities ..... 64
Time Schedule for Interaction ..... 65
Human Limitations in Computer Interaction ..... 66

## TABLE OF CONTENTS--Continued

Page
Input/Output Specifications ..... 67
Inputs ..... 68
Outputs ..... 68
The CHOOSE Programs ..... 68
Auxiliary Systems for Building CHOOSE Models ..... 69
The Makeup System ..... 69
The CODEM System ..... 70
Overview of CHOOSE System Model Development ..... 70
5. APPLICATION OF THE CHOOSE METHOD ..... 73
Example for Home Sound Reproduction Equipment ..... 74
Application to Watershed Models ..... 79
6. CONCLUSIONS, RECOMMENDATIONS, AND EXTENSIONS ..... 82
Possible Extensions to CHOOSE ..... 82
Possible Fields of Application for CHOOSE ..... 84
Potential Program Improvements ..... 86
APPENDIX A: USING THE CHOOSE SYSTEM ..... 87
Interactive Instruction Manual ..... 88
Steps ..... 88
Step 1. The Interactive Terminal ..... 89
Step 2. Establishing Contact with the Computer ..... 89
Step 3. Preliminaries ..... 89
Step 4. Section I of CHOOSE. Eliciting Input Data ..... 92
Step 5. Section II of CHOOSE. Review of Input ..... 97
Step 6. Section III and IV of CHOOSE. Summary and Encoding of Input ..... 97
Step 7. Section V of CHOOSE. Interaction ..... 99
Step 8. Termination ..... 106
Strategy for Interactive Negotiations ..... 106
The Interactive Phase ..... 110
Bringing Up Programs for Interaction ..... 112
APPENDIX B: THE EXAMPLE MAIN PROGRAM - TPDEK8 ..... 114
APPENDIX C: THE CHOOSE SUBPROGRAM SET ..... 120
The NEED Array ..... 121
The MODEL Array ..... 123
A Pattern of Statements Involving BRANCH ..... 124
Subroutine CHOOSE ..... 125
Supporting Subprograms ..... 136
Subprogram BRANCH ..... 136
Subprogram CHAR ..... 136
Subprogram INEED ..... 136
Subprogram QUEST ..... 145
Subprogram QUERY ..... 145Page
Subprogram TABLE ..... 145
Subprogram CODEIN ..... 145
Subprogram DISPL ..... 145
Subprogram SPECIAL ..... 150
Subprogram QFILE ..... 153
Subroutine Calling Structure ..... 153
APPENDIX D: COMPLETE PROBLEM METHODOLOGY ..... 154
APPENDIX E: PROGRAM MAKEUP ..... 159
Input ..... 159
Output ..... 160
APPENDIX F: PROGRAM CODEM ..... 171
Output ..... 172
APPENDIX G: TAPE DECK CHOICE MODEL ..... 180
CHOOSER's Viewpoint ..... 180
The EVALUATOR's Viewpoint ..... 182
The Complete Main PROGRAM TPDEK8 ..... 183
REFERENCES ..... 186

## LIST OF TABLES

Table Page

1. Percentage Chance of Finding Feasible Solution . . . . . . . 51
2. Sample Responses in Choosing an 8-Track Tape Deck . . . . . . 77
Figure Page
3. CHOICEMODEL Equivalences ..... 27
4. Feasible Space without Value Judgments ..... 46
5. Feasible Space with Value Judgments ..... 47
6. A Diagram of the Relationships of the Primary Ratings ..... 58
7. The CHOOSER as both System User and System Component ..... 66
8. Abbreviated Flow Chart for CHOOSE Methodology ..... 71
9. Consumers Union Ratings of 8 -Track Tape Decks ..... 75
10. Sample of Organized Information for Watershed ..... 80
11. Step 2. Establishing Connection ..... 90
12. Step 3. Preliminaries ..... 91
13. Standard Responses and Their Meanings ..... 93
14. Section I. Opening and Method 1 of Entry ..... 94
15. Using the Method 2 Entry Option ..... 96
16. Sections II and III Illustrated ..... 98
17. Section IV. Coded Output Example ..... 99
18. Brief Outline of the Rating Scheme ..... 100
19. Section V. Example of Interaction ..... 101
20. Step 7. Section V. Interaction ..... 104
21. Step 8. Termination ..... 107
22. CHOOSE Subprogram Calling Structure ..... 152
23. Step by Step Process for Project TPDEK8 ..... 155
24. Input Cards for MAKEUP Using TPDEK8 Example ..... 162
25. Computer Output for MAKEUP Using TPDEK8 Example (Block Data) ..... 162
26. Computer Output for MAKEUP Using TPDEK8 Example (Main Program) ..... 163
27. Computer Listing of Typical Header Card and Candidate Cards for Input to CODEM ..... 174
28. Segment of Line Printer Output from CODEM ..... 175
29. Typical Input Set for CODEM with TPDEK8 ..... 184
30. Sample of CODEM Output for TPDEK8 ..... 184


#### Abstract

The problem of selecting appropriate mathematical models for use in studying hydrological phenomena has created a situation in which the choice of suitable models by hydrologic practitioners has become exceedingly complex. The extensive comments in the literature indicate that neither the traditional system of technical journals nor the more modern computer-based retrieval schemes have really solved the problem. Further examination shows that similar problems have arisen in many fields, hence a well organized attack on the specific problem of hydrologic model choice can have a more general application. The present problem is identified as a requirement to codify and make accessible to users information in a more directly user oriented format.

The problem of model choice arises at several levels, ranging from decision on what fundamental structure to use, to choice of parameters, and on to model calibration and validation. This paper is focused on a scheme to aid in model structure choice.

The essential ingredients of model structure choice, and indeed of many choice processes, are extracted and embedded in a generalized set theoretic mathematical notational framework in order to give some insight into the nature of the problem. Within this framework the specialized features of the model choice problem are analyzed, and a specialized model is developed for assisting in model choice and all problems similarly situated.

These considerations lead to the development of a finite vector of objective statements with codified responses prepared by a panel of qualified researchers who are willing and able to construct the essential information in a user oriented format. It is required that the panel not only couch their information in objective oriented terms but that they also generate value judgments for the individual components. In this way, those using the system can take advantage of the expert opinions embedded in the model while, at the same time, tailoring the choice to meet their own specific needs and aspirations. This results in what is defined as a mathematical CHOICEMODEL.

The implementation of a system for interactive computation of the CHOICEMODEL is described in detail, and the associated computer programs are presented in appendices.

A detailed instruction manual is given, and the implementation of the method is illustrated by an easily understood model of the ingredients of the problem of selecting an 8-track stereo tape deck for home use. The plan is outlined whereby hydrologic choice models can be developed within the CHOICEMODEL system by a selected panel of expert EVALUATORs .


## INTRODUCTION

What the world doesn't know is how to handle what it does know.

Xerox Corporation (1971)
The principle above, expounded by Xerox, represents the underlying philosophy of this dissertation. A great body of scientific knowledge has accumulated over the years, but all too often the information is either not available when needed, or those who could profit from it are unaware of its existence and applicability. The result is that previous work is often replicated, some very useful insights and findings are never applied in situations where these would be welcome and needed, and other work is lost, never to be rediscovered.

The present partial-sum of human knowledge is certainly well beyond the capacity of any individual to command. It is easy to believe that this has been true for thousands of years. It has most certainly been true since the time of Leibnitz ( $1646-1716$ ) who is reputed to be the last person to have "had a full command of all the intellectual activity of his day" (Wiener, 1948, p. 8). Now we can safely say that no one "knows everything," even about any field of specialization.

From the point of view of practical application of knowledge and ideas; the best we seem able to achieve now is an adequate understanding of the fundamental principles of one field of inquiry and an acquantance with closely related fields. This knowledge sometimes permits an investigator who is oriented to problem-solving to bring the best available tool to bear on a particular problem. But now even this is often inadequate. Even the broad knowledge of a specialty and related fields does not insure that a person will be able to retrieve and focus on the best of current knowledge to apply to a particular problem. And, failing that, the problem may not be treated in optimal, or near optimal, fashion. It now seems that more effort must be expended to increase the level of help available to individuals and groups in retrieving and applying information appropriate to their specific problems.

One area where the problem of adequate dissemination of knowledge seems to be particularly pressing is in the choice of mathematical models for phenomena under study. This dissertation will focus on what may be done to aid model users and potential model users in choosing suitable model structures. This will be done by the development of a scheme which, in the context of the particular situation in which the model is to be used, will restrict attention to a narrow set of candidate models. It will then help guide the user toward a near-optimal final choice of model structure based on his specific requirements and resources.

The scheme will be seen later to be applicable to situations in winich the major problem is better choice among alternatives in a much broader sense. However, the context in which these procedures were developed was the choice, by an intelligent but (perhaps) uninformed user, of suitable mathematical models for attacking his specific problem. Since the problem of model choice is in the background of all subsequent discussion, an overview of models and their roles in problem solving seems appropriate.

## Mode1s

The fundamental assumption of system theory is that the external behavior of any physical or conceptual phenomena can be described in terms of a suitable mathematical model (Booth, 1967, p. 2). The model can then be used for many purposes, such as prediction of future behavior under given input conditions, determination of optimum inputs to achieve particular sets of goals, and prediction of changes in behavior of the real system under study by carrying out studies of the behavior of a suitably modified model.

The concept of a model is an old one. While the traditional use of the word may bring to mind scaled physical copies, it now has the much broader meaning of "a representation of the system under study" (Churchman, Ackoff and Arnoff, 1957, p. 155). Such a definition is too broad for use here, however, and it will be convenient to restrict the meaning to mathematical models. By mathematical models we will mean symbolic descriptions using both numbers and generalized sets to exhibit formally the concepts and interrelationships of the process under investigation. Since the hydrologic examples in this study are limited to models which can be programmed for predictive work on medium to large scale digital and hybrid computers, the term model will often be used to mean mathematical models directly adaptable for such use.

The role of models is fundamental to any sort of human understanding of systematic inquiry. Rosenblueth and Wiener (1945, pp. 316321) state:

No substantial part of the universe is so simple that it can be grasped and controlled without abstraction. Abstraction consists in replacing the part of the universe under consideration by a model of similar but simpler structure. Models, formal or intellectual on the one hand, or material on the other, are thus a central necessity of scientific procedure. . . . The ideal model is the universe itself. If you . . . could comprehend it you wouldn't need an abstract mode1. But the human mind is finite.

The fixing and formalizing of these ideas since that time have made it fashionable to describe all human thought processes as model-building activities and to couch all scientific ideas and results in the same type of wording. We now speak (not-disjointly) of models of many types, e.g., iconic models, analog models, physical models, simulation models,
cost models, optimization models, conceptual models, stochastic models, deterministic models, and mathematical models.

Wymore's (1971, pp. 3-5) position is similar with respect to the role of models in research, but he also extends the ideas to engineering.

I tend to believe that the role of models in research is so important that one might define research as the process of the development of models. . . .

Since I am also interested in engineering, it is important to me to point out that the objective of engineering is not necessarily the development of models. The objective of engineering activity is to produce some thing. . . . In order to design this thing the engineer typically makes use of models which have been developed in science or, lacking such models already developed, he may have to develop his own models. But the ultimate objective of model development is different in engineering than it is in science and hence, the criterion for what constitutes a satisfactory model may be different for the engineer than it is for the scientist.

The engineer is always faced with much more explicit economic and time constraints than is the scientist or, at least the way the engineer typically regards his economic and time constraints implies a different attitude towards complexity. The engineer can only deal with complexity to the extent that his resources in time and money available on the contract in force at the moment will allow him. The scientist, on the other hand, seems usually to be much more powerfully motivated to deal with complexity more "honestly": He wants his model to encompass every aspect of the phenomenon under study that he can perceive. . . .

The engineer, presumably, does as good a job as he can within his constraints.

Researchers in many fields have found a powerful tool in the modern electronic digital computer. The large memory and computational capacities of these computers have opened the way for extensive formal treatment of questions arising in many fields formerly unreachable by mathematical methods. In using this new tool, they have found it necessary to develop mathematical models suitable for treatment by these computers, or to disinter relevant old ones. In the past fifteen to twenty years researchers in such diverse fields as social sciences, business, and natural sciences, have produced many such models and used them with varying degrees of success.

But in each of these fields there is a strong feeling of incompleteness of the work. Each reports a significant gap between theory and practice. Researchers are almost to a man disappointed in the failure of practitioners in their respective fields to make effective use of their concepts and models in order to develop insight into their every day problems.

In the management science area this is felt strongly. Slevin and Schultz (1972) summarize it well in their proposal for a conference of the implementation of Operations Research/Management Science models.

Operations research/management science (OR/MS) has developed over two decades as a mature field of academic study and research. Important advances have been made in mathematical programming, stochastic modeling, and general systems analysis. The result of this work is an inventory of OR/MS models and systems that has great potential in solving management problems. Despite the promise of OR/MS as a problem-solving discipline, the reality of 2972 shows a significant gap between management science theory and managerial application. Thus, the potential of $O R / M S$ is largely unrealized. (Emphasis supplied.)

A serious problem of this type also is reported to exist in the field of mathematical modelling of hydrologic phenomena. Dracup, Mobasheri and Cardenas (1970, pp. 4-6) have perhaps sumnarized this best as follows:

While much research effort has been expended in identifying and developing the use of systems analysis techniques in the analysis of water resource systems, few practicing engineers have adopted the use of these techniques. This apparent anomaly is an outgrowth of two major deficiencies that have become apparent with the evolution of systems analysis in water resources:

1. A communications gap between researchers and practicing engineers has developed to astonishing proportions. . . .

That the communications gap exists is apparent to all familiar with water resources. . . . The fact remains that many excellent research efforts have been successfully completed. The practitioner remains in need of analytical tools and techniques. Yet the research results that exist have not yet evolved to the stage of being applied . . . . It is the purpose of this report to delineate the directions that must be taken in near-term research to eliminate this alarming hiatus. (Emphasis supplied.)

Perhaps the first problem a practitioner encounters in attempting to develop a model to attack a particular problem is the great number and diversity of model structures from which to choose. In many fields the sheer number of models which have been developed or postulated in the last two decades is enough in itself to cause concern. Further, many models, though of potential practical usefulness, may be presented in obscure fashion, in incomprehensible language and poor notation. This illustrates that there can be a human factors problem of presenting or "marketing" a model in an appropriate way. If the potential model builder also feels that he must obtain a detailed understanding of each of the many models before he can make. a rational or
reasonable (if not optimal) choice, he will be further inhibited in his attempts to use the most applicable research results. He may very well feel that it is simpler to forget about the vast literature and retreat to traditional rules of thumb. Or he may, without any substantial use of prior work for foundation, develop his own model to fit his own circumstances (or at least his perception of his own circumstances). Indeed, his own pride may propel him in this direction in any case. This, at best, is likely to cost more time and money, and, at worst, is likely to produce a model in which many important considerations previously discovered by others are overlooked. It may also add another model of marginal value to the library of available models.

The would-be hydrologic modeller, therefore, may find it useful to have access to some reasonable, quick and inexpensive way to locate the models best suited to his needs and to eliminate (or simply fail to call attention to) those models which are deficient in some important aspect for the problem under consideration.

The use of mathematical models for digital computer investigations of problems in hdyrology has been accepted on a large scale. Viessman (1970) reported that, among the 50 states, plus Puerto Rico, 24 were currently using mathematical or simulation models in their state water planning agencies. He further reported that "about half of them are making use of various types of models. The level of sophistication and originality varies considerably. It seems clear however, that modelling techniques will play an increasingly important role in state water planning efforts."

Interest is not limited to the United States; the interest in mathematical models in hdyrology is world-wide. Popkin (1975) has compiled a report of the state of research and application of only stochastic methods in hdyrology which incorporates contributions from 24 countries active in such work. Many of such activities are monitored and/or guided by agencies of the United Nations such as the UNESCO Working Group on Representative and Experimental Basins (Ibbitt, 1972), and the Food and Agriculture Organization of the United Nations (1972).

While Viessman notes some distress with the high cost of systems analysis studies, the overall emphasis seems to be on improving techniques for model choice, model building and model manipulation to produce better results and to expand the range of situations which can be treated by these methods. There may be some who would avoid these newer methods and remain with unaided professional judgment and intuition, but the complexity of the problems for which some sort of answers are now needed seems to be reinforcing the trend to the development of models amenable to manipulation by large digital computers or hybrid computers.

Even in those circumstances where the educated intuition and judgment of experienced professionals would be adequate for a particular problem, newcomers to the field cannot be expected to have the requisite background. Use of appropriate models, then, can aid in solving the
immediate problem as well as providing the experience and educating the discretion of the inexperienced as they develop the maturity of judgment needed.

## The Overall Modelling Problem

The problems of creating mathematical models for particular applications can be considered at many levels. Although this study will be concerned almost exclusively with model choice (in the limited sense of "model structure choice" as defined below), it will be useful to describe briefly each of the general steps of the modelling process.

1. Model Design: We may consider that model design is that period of the modelling process in which fundamentally different model elements and interrelationships are tested. There are two basic sources of model ideas for this step. One is to begin with first principles based on knowledge and insights into the physical processes which are known or suspected to be taking place within the system under investigation. The other is to see what structure must be built into the mathematical description in order to permit it to produce the input-output relationships known to have existed in the past, or thought to be required. That is, internal structure is imputed from behavior. In either case the selection or development of model structures and their proper verification by the steps described below is an important process and is the part of the process which has received most of the research attention to date.
2. Modél Structure Choice, or simply 'Model Choice": Formally, here, model (structure) choice is defined as the selection of the fundamental mathematical structural elements to which parameters are to be assigned. A simple example would be the choice of a linear or a nonlinear stochastic model for streamflow sequence generation.

Since researchers are presumed to have developed a family of model structures to meet many needs, for the practicing engineer the modelling process would normally begin with "model structure choice." For the purpose of this dissertation, the process also will begin here, since we are developing model structure choice criteria specifically for those fields in which a substantial effort has already been devoted to model design. We are, then, seeking methods to move the most suitable models from research to engineering.
3. Model Building, or Parameterization and Coupling: After the basic model structure(s) has been selected, it is necessary to assign values from suitable sets to the parameters and to specify the linkages between the various elementary models which will make up the system. In any sort of a complex situation this part of the process is quite difficult, since parameter values existing in the field environment cannot, in general, be measured directly. They must be imputed from other observed behavior. As an example, important parameters of subsurface water flow are storage capacity of the subterranean structure and the transmissivities in various directions. These parameters cannot
be measured directly since any attempt to obtain samples for test disturbs the structure so that test results are no longer valid. The parameters are generally estimated from measurements made of the water level change (draw down) experienced in a series of wells during certain controlled pumping tests; that is, they are "imputed" on the basis of what values must be present to produce the observed result.
4. Model Calibration: The calibration of the model is a process where the model is operated in conjunction with the selected computer in such a way as to insure that the model represents the real world sufficiently well to make it useful for treatment of the problem at hand. This is really an interactive phase with step 3, model building, since, often, the assignment of parameter values cannot be done a priori with any degree of confidence.

If the first four steps of the modelling process are accomplished by imputing structure, parameters, and linkages from input/output observations, the process is called system identification (Astrom and Eykhoff, 1970; Sagar, 1973). In hydrology (Emsellem and deMarsily, 1969), and in other fields, it is often referred to as the inverse problem.
5. Model Validation: Model validation is a continuing aspect of model use wherein the calibrated model is under surveillance to see that it is, and continues to be, an adequate representation of the phenomenon under study. It can be considered an extension of the calibration step.

Steps 3, 4, and 5 can be similar in many respects in that, in general, they are usually accomplished by use of problem-oriented data. In some cases, there is an arbitrary partition of the data into sets to be used for each of the three steps. That is, sometimes data used during the model building and model calibration and model validation are indistinguishable except in the manner of their use. In a philosophical sense a distinction may be made between calibration and validation in that the modeller himself must, in some way, establish the ability of the model to serve the intended purpose (calibration), while others (users) who may enter the scene at step 5, can validate the model.

One additional difficulty of definition arises in that the term "model choice" is often used to encompass the steps 2, 3, and 4 of the modelling process as defined above. That is, models having identical structures but differing from each other only in the values of one or more parameters are, in some sense, different models. Hence, the process of choosing parameters would be considered as a "model choice" question. For purposes this study, however, the term "model choice" is restricted to mean on step 2 of the process described above--that is, selection of model acture.

Johnston (1969) and kates (1962) in the context of model (structure) choice for hydrology have referred to the assistance needed by users as "broadening their range of choice" and freeing them from "the prison of experience." The work described here is aimed directly at this problem.

## Preview of Remaining Chapters

Chapter 2, Origins of the Model Choice Problem, gives an overview of the nature of the modelling process and methodology of model development as depicted by several of the important contributors to the art. It also reviews some of the formal approaches to model choice questions proposed by others. Here the concept is developed that the important problem of model structure choice is prerequisite to effective model building or model parameterization.

Chapter 3, Designing the CHOOSE Mode1, develops the conceptual framework for embedding the information that is needed to guide choices of model structure. This framework is seen to contain some important features from the concepts of multiple objective mathematical optimization and of information retrieval. The CHOOSE model, however, represents simply a function--a mapping of inputs into outputs without considerations of present state or elapsed time. To make a model useful requires a system for interaction and computation.

Chapter 4, Implementing CHOOSE Models, shows how the requirements of the human CHOOSERs and of the CHOOSE models lead to an overall computer based interactive system for placing the needed information at the hands of the CHOOSER. Also certain auxiliary systems are developed which are useful in preparing a CHOOSE model for a particular application and for placing it into a form for use by the CHOOSE system.

Chapter 5, Applications of CHOOSE Methodology, contains a detailed example of an application of the CHOOSE techniques to a simple situation. This example allows the CHOOSE concepts to be grasped without becoming engulfed by the details of a complicated real world application. Chapter 5 also cites the types of model structure information needed for hydrologic applications and refers to several frameworks offered by others to organize such information. These attempts at organization appear to be well suited for applications to CHOOSE procedures.

Chapter 6 gives the conclusions, recommendations and suggestions for extension of the CHOOSE methodology.

## Specialized Notation and Definitions

At the outset it will be useful to define several terms as they are used here.

CHOOSER: The term CHOOSER is used here to indicate the person or group who has the requirement for making an appropriate choice or decision among alternatives in the context of a particular problem that has arisen. In the multiple objective optimization or a decision making terminology he would ordinarily be called a Decision Maker (DM). Here, however, the choice that is made is not necessarily irrevocable in the sense of decision theory.

EVALUATOR: The term EVALUATOR refers to those who follow the guidelines in this dissertation to establish the evaluation criteria and/or the specific evaluations of the various alternative candidates. In the context of the problems under study here, CHOOSERs and EVALUATORs are disjoint sets.

CHOOSE Model: The CHOOSE model for a particular application is the set theoretic model developed for a particular problem in accordance with the criteria of Chapter 3, and the methodology of Appendix D.

CHOOSE System: The CHOOSE system refers to the complete set of programs and related hardware facilities for both generating CHOOSE models and making them usable by CHOOSERs.

In order to be able to speak precisely and concisely about the various theoretical aspects of the problem, the pattern of notation developed by Dr. Wymore (1967, 1970, 1972, 1973) in his studies of generalized systems has been adapted. Certain of the set and function theoretic definitions are defined below as used here. For a complete set of definitions, refer to Appendix A of his Systems Engineering Methodology for Interdisciplinary Teams.

The set and function theoretic definitions conclude Chapter 1. Those who are well acquainted with this notation, or who do not expect to read in detail the symbolic representation used in Chapter 3, can omit the balance of this chapter, without loss of continuity.

Sets: Set is an undefined term. If $A$ is a set and $x$ is an element of $A$ or, equivalently, if $x$ belongs to $A$, the fact is denoted $x \in A$ but is not otherwise defined. The fact that $x$ does not belong to $A$ is denoted $x \notin \mathrm{~A}$.

Subsets: If $A$ and $B$ are sets, then $A$ is a subset of $B$ if for all $x \varepsilon A, x \varepsilon B$. The fact that $A$ is a subset of $B$ is denoted $A \subset B$. The set $A$ is equal to the set $B$ if and only if $A \subset B$ and $B \subset A$. The fact that the set $A$ is equal to the set $B$ is denoted $A=B$. The set of all subsets of the set A is a set denoted SUBSETS(A), defined as follows: $\operatorname{SUBSETS}(\mathrm{A})=\{\mathrm{B}: \mathrm{B}$ is a set and $\mathrm{B} \subset \mathrm{A}\}$.

Complements: The complement of the set $A$ in the set $B$ is denoted $B \sim A$ and is defined as follows: $B \sim A=\{x: x \in B$ and $x \notin A\}$. The empty set is denoted $\emptyset$ and is defined as follows: $\emptyset=A \sim A$ for any set A.

Intersections: If $A_{1}, A_{2}, \ldots, A_{n}$ are all sets, then the intersection of the sets $A_{1}, A_{2}, \ldots, A_{n}$ is denoted $A_{1} \cap A_{2} \cap \ldots \cap A_{n}$ and is defined as follows: $A_{1} \cap A_{2} \cap \ldots \cap A_{n}=\left\{x: x \in A_{1}\right.$, and $x \in A_{2}, \ldots$, and $\left.x \in A_{n}\right\}$.

Numbers: The set of all real numbers is denoted $R$. The set of all positive real numbers is denoted $\mathrm{R}^{+}$. The set of all non-negative real numbers is denoted $\mathrm{R}^{++}$. The set of all non-negative integers is denoted
$\mathrm{I}^{--}$. The set of all positive integers is denoted $\mathrm{I}^{++}$. The set of all integers from $n$ through $m$ inclusive, where $n \leq m$ is denoted $I[n, m$.

Vector Products: If $A_{1}, \ldots, A_{n}$ are all sets, then the vector product of the sets $A_{1}, \ldots, A_{n}$ is denoted $A_{1} X \ldots X A_{n}$ and is defined as follows:

$$
\begin{array}{rl}
A_{1} & X \ldots X A_{n} \\
= & A_{1} \text { if } n=1 ; \\
= & \left\{\left(a_{1}, \ldots, a_{n}\right): a_{i} \varepsilon A_{i} \text { for each } i \varepsilon I[1, n]\right\} \\
& \text { if } n \varepsilon I[2, \infty) .
\end{array}
$$

Alternatively the notation $\underset{i=1}{\mathrm{X}} \mathrm{A}_{\mathrm{i}}$ will be used to represent the vector $i=1$
product. Where each set $A_{i}$ making up the vector product is the same set $A$, the notation $A^{n}$ will be used to represent the vector product.

Projections: If $A_{1}, \ldots, A_{n}$ are all sets, and $i \varepsilon I[1, n]$ and $a \varepsilon A_{1}$ $X \ldots X A_{n}, a=\left(A_{1}, \ldots, a_{n}\right)$, then the ith coordinate of $a$ is $a_{i}$ and the projection of a into its ith coordinate is denoted $\pi_{i}$ (a) and is defined as follows: $\pi_{i}(a)=a_{i}$.

Functions: If $A$ is a set and $B$ is a set, then the set of all functions defined on A with values in B is denoted FUNCTIONS (A, B) and is defined as follows: FUNCTIONS (A,B) $=\{\mathrm{f}: \mathrm{f} \varepsilon$ SUBSETS (AXB), and for every $a \varepsilon A$ there exists $b \varepsilon B$ such that $(a, b) \varepsilon f$, and if (a,b) $\varepsilon f$ and $(\mathrm{a}, \mathrm{c}) \varepsilon \mathrm{f}$ then $\mathrm{b}=\mathrm{c}\}$. If $\mathrm{f} \varepsilon \mathrm{FUNCTIONS}^{(\mathrm{A}, \mathrm{B})}$ and $\mathrm{x} \varepsilon \mathrm{A}$, then the value of the function $f$ at $x$ is denoted $f(x)$ and is defined as follows: $f(x) \varepsilon B$ and $b=f(x)$ if and only if ( $x, b) \varepsilon f$.

Range: If $f \varepsilon \operatorname{FUNCTIONS}(\mathrm{~A}, \mathrm{~B})$, then the range of f is denoted $\operatorname{RANGE}(\mathrm{f})$ and is defined as follows: $\operatorname{RANGE}(\mathrm{f})=\{\mathrm{b}: \mathrm{b} \varepsilon \mathrm{B}$, there exists $a \varepsilon A$ such that $b=f(a)\}$.
\#. If $A$ is a finite set, then the number of elements in $A$ is denoted \# (A).

Vector Inequalities: The symbols $<,>, \leq, \geq, \notin \neq, \downarrow$, and $\neq$ have their customary meaning when comparing two real numbers. It will be useful here to define a notion of vector inequality as used in this paper. If $\underline{a}$ and $B$ are $n$ vectors with real components, then $a \leq b$ is defined to mean that $\pi_{i}(\underline{a}) \leq \pi_{i}(\underline{b})$ for all i $\varepsilon I[1, n]$. It should be clear that $\mathfrak{a} \neq b$ does not imply that $\mathfrak{a} \geq \underline{b}$. The symbol $\geq$ is not defined or used for comparing vectors here. The symbol $\ddagger$ simply means that for some component $j$ that $\pi_{j}(\underline{a}) \varepsilon \pi_{j}(\underline{b})$. Note that this
definition for vector inequality is an alternate form of representation of a tradeoff ordering defined below.

Orderings: Let A be a set. The set of orderings over A is denoted ORDERINGS(A) and is defined as follows: ORDERINGS(A) = $\left\{\alpha: \alpha \in \operatorname{FUNCTIONS}\left(\mathrm{A}^{2},\{F A L S E, T R U E\}\right)\right.$ such that for every a $\varepsilon \mathrm{A}, \mathrm{b} \varepsilon \mathrm{A}$, and $\mathrm{c} \varepsilon \mathrm{A}, \alpha(\mathrm{a}, \mathrm{a})=T R U E$, and, if $\alpha(\mathrm{a}, \mathrm{b})=T R U E$ and $\alpha(\mathrm{b}, \mathrm{c})=T R U E$, then $\alpha(\mathrm{a}, \mathrm{c})=T R U E\}$. Any element of $\operatorname{ORDERINGS}(\mathrm{A})$ will be called an ordering, a set ordered by •, or often simply an ordered set.

An ordering $\alpha \varepsilon \operatorname{ORDERINGS}(\mathrm{A})$ is partial if for all a, b $\varepsilon \mathrm{A}$, if $\alpha(\mathrm{a}, \mathrm{b})=T R U E$ and $\alpha(\mathrm{b}, \mathrm{a})=T R U E$, then $\mathrm{a}=\mathrm{b}$ (antisymmetry). Similarly, an ordering $\alpha$ is simple if and only if for $a 11 \mathrm{a}, \mathrm{b} \varepsilon \mathrm{A}$, at least one of the following occur: $\alpha(\mathrm{a}, \mathrm{b})=T R U E$, or $\alpha(\mathrm{b}, \mathrm{a})=T R U E$. Elements from a finite set that are simply ordered can be ranked numerically (ties permitted) in the common meaning of the term. An ordering is total if it is both partial and simple. Elements from a finite set that are totally ordered can be ranked numerically without ties.

Equivalence Relation and Partitions: If $\alpha \varepsilon \operatorname{ORDERINGS(A)~and~}$ $\alpha(a, b)=\alpha(b, a)$ for every $a, b \varepsilon A$ (symmetry) then $\alpha$ is an equivalence relation. Note that this symmetry condition distinguishes between an equivalence relation and a partial ordering for which the anti-symmetry condition applies. An equivalence relation partitions the set A into disjoint subsets of equivalence classes. Each element of an equivalence class has behavior equivalent to the others in the same class (subset) in the sense that any one element of the class can represent the entire class. That is, they are "tied" in their behavior for the ordering relation being used.

An ordering relation can be both an equivalence relation and a partial ordering relation only trivially; that is, only if $\alpha(a, b)=$ $\alpha(\mathrm{b}, \mathrm{a})=\operatorname{TRUE}$ for all $\mathrm{a}, \mathrm{b} \varepsilon \mathrm{A}$, or if $\alpha(\mathrm{a}, \mathrm{b})=\operatorname{TRUE}$ only when $\mathrm{a}=\mathrm{b}$.

Optimality: If A is ordered by $\alpha$ and there exists an element $a^{*}$ $\mathrm{a}^{*} \varepsilon$ A such that for $\mathrm{all} \mathrm{b} \varepsilon \mathrm{A}, \alpha\left(\mathrm{b}, \mathrm{a}^{*}\right)=T R U E$, then $\mathrm{a}^{*}$ is optimal with respect to $\alpha$.

If A is partially ordered by $\alpha$ and has an optimal element a*, it is a unique element called the upper bound, and the partial ordered set is an upper semi-lattice.

Maximality: If $\alpha \varepsilon$ ORDERINGS (A), and $\mathrm{a}^{\prime \prime} \varepsilon \mathrm{A}$, then $\mathrm{a}^{\prime \prime}$ is maximal with respect to $\alpha$ if and only if for all $\mathrm{b} \varepsilon \mathrm{A}, \alpha\left(\mathrm{b}, \mathrm{a}^{\prime \prime}\right)=F A L S E$ implies $\alpha\left(\mathrm{a}^{\prime \prime}, \mathrm{b}\right)=F A L S E$. Note that if optimal elements are present in A, then there are no elements in.A that are maximal but not optimal.

Tradeoff Ordering: Let A be a set not empty and let $\alpha, \beta$, $\gamma \varepsilon$ ORDERINGS(A). Then $\gamma$ is a tradeoff ordering between $\alpha$ and $\beta$ if and only if: for $a, b \varepsilon A$
(1) $\alpha(\mathrm{a}, \mathrm{b})=\beta(\mathrm{a}, \mathrm{b})=T R U E$ then $\gamma(\mathrm{a}, \mathrm{b})=$ TRUE;
(2) $\alpha(\mathrm{b}, \mathrm{a})=\beta(\mathrm{b}, \mathrm{a})=T R U E$, and $\alpha(\mathrm{a}, \mathrm{b})=F A L S E$
or $\beta(\mathrm{a}, \mathrm{b})=F A L S E$, then $\gamma(\mathrm{a}, \mathrm{b})=F A L S E$.

Note that this corresponds to the vector inequality defined above if comparisons are made on the individual elements of the vector on the basis of $\leq$.

## CHAPTER 2

## ORIGINS OF THE MODEL CHOICE PROBLEM

> Everyone complains of his memory, and no one complains of his judgment.

La Rochefoucauld (Bartlett, 1941)
A memory lapse, if we can accept La Rochefoucauld's statement, is a more acceptable human failing than poor judgment. A person apparently is more willing to admit to overlooking something than he is to admit that he made a bad choice among the various alternatives which he actually considered. Yet the opportunity loss which is occasioned by making even the best possible choice from a set of alternatives limited by memory may be high--much higher, perhaps, than that of making a suboptimal choice from a larger domain of alternatives.

Poor memory, however, is not the only mechanism which restricts the domain of choice. The failure to search for additional alternatives, the lack of information on where to look, the lack of creativity in establishing alternatives, and the lack of knowledge of the existence of alternatives which are created by others also limit the number of alternatives which come under consideration.

Here, the problem of model choice is under study. The goals are to decrease excessive dependence on memory, to encourage potential model users to enlarge the domain of choice, to show them where to look, and to make them aware of the existence of the method described herein for model choice. The intent is not to suppress creativity but to see that creativity is not wasted on duplicating the efforts of others.

There is a great deal of variation in the complexity of mathematical models. They range from the simplest algebraic equations to mathematical descriptions which may require several hundred pages. Mathematical modelling of systems seems to be the least rigorous and most subjective part of the whole process of systems analysis and design. However, once the model in the mind of the modeller is reduced to a firm, communicable mathematical form, the investigator can turn to many formal methods of manipulating it to seek answers to the questions which originally gave rise to the model. Or he may endeavor to answer additional questions which may have emerged during the development of the model.

Help in modelling is hard to achieve in itself, since it is the artistic part of the process. Several writers have suggested steps to follow in the modelling process (Astrom and Eykhoff, 1970; Wymore, 1971, 1973), but, for the most part, the art of modelling is taught by example (Wymore, 1967, Chapter 3). Usually this is done by the description of an overly simplified situation or by the description of models which have worked well in specific situations. All of this is not very encouraging to those who may lack "artistic" insight yet who are, nevertheless, the consumers or "needers" of models.

Klir (1972, p. 9) seems to feel that improvements in systems modelling methodology will come in time. With respect to Wymore's (1967 and 1972) general systems theory Klir states, "Unfortunately, there has not been enough time to elaborate the methodology associated with the approach. In particular, very little has been done in regard to systems synthesis, which is of primary importance in engineering." He also considers his own theory (Klir, 1972, Chapter 7) and that of Mesarovic (Klir, 1972, Chapter 8) to be inadequately developed from a methodological point of view. Subsequently, however, Wymore (1973) made some significant steps in development of methodology.

While a general modelling methodology may be useful framework for discussing, or teaching modelling, there is no immediate prospect for a largely automated procedure. Hence, one of the problems facing systems engineers today, in the development of models of phenomena in areas where mathematical descriptions have not been applied previously with any degree of success, is a genuine lack of a formalized way in which to proceed. But adequate models are essential if the power of modern digital computers is to make itself felt strongly in newer fields, and if engineering is to transfer its methodology to complex problems that so far have not been treated in engineering terms.

Hall (1971, pp. 1-13) takes a strong position "against a universal model methodology." In reference to "recurrent attempts to provide a unifying methodology for all scientific modelling" he continues:

The idea that such a unified methodology exists and can be discovered by sufficient reflection is almost an article of faith within the scientific community. Discussions of this proposition usually place special emphasis on the common basis all scientific work has in the scientific method, which has been defined with varying degrees of precision by philosophers of science. From this point it is usually assumed that the present balance between scientific method and creativity, between craftmanship and insight, can be progressively shifted toward greater systematization until some operational plan is identified which can be routinely followed to a solution in any problem situation.

In this paper, I will put forward a case against this proposition as a feasible or even a desirable goal for scientific activity. ... [I]nstead of asking for universal models, we should seek means of drowing forth competing models so as to accelerate a dialectic process of model extension.

A methodology of modelling, if it existed, would consist of a complete program for acquiring, structuring, relating, and generalizing experience.

I suggest that the answer lies in a deliberate and conscious rejection of the idea of a unified modelling methodology.

Following Hall's suggestion the focus here will be on "drawing forth competing models" on a very selective basis for determination of their usefulness in solving a specific problem of model application, rather than attempting to develop a universal modelling methodology.

## The Model Overchoice Problem

The development of digital computers with large internal memory and high computational speeds has given a great impetus for researchers to formalize models on a large scale. Klir (1972, p. 4) outlines the role of computers in this process. "For systems theorists the computer is a tool as basic and essential as the microscope is for the biologists. Either of these tools enhances enromously the ability of the human being in a particular area."

This has led to a rapid and somewhat disorderly development of models of all types: conceptual models, mathematical models, mathematical systems descriptions, computer programs, simulation models, etc. The literature in some fields contains frequent announcements of new models or refinements of models previously demonstrated or postulated.

The proliferation of models at the academic and research levels, in many fields, has created a number of difficulties for the potential user. Many models have not found broad acceptance and widespread use by practitioners. This may be due to several reasons, among them: some models, in fact, are not very applicable to real problems; some models are unproven or fail to give reliable results; others, while useful for research leading to understanding, may be too complicated. for routine field use. Also, it appears that many investigators have found it easier and perhaps more satisfying to create a new model rather than to build on activities of others.

But many claim that valid and useful models are not more widely used for more mundane reasons: the models are not marketed in an appropriate manner; the potential users have not kept up with research efforts; the users would rather stay with familiar (less useful and perhaps potentially dangerous) models than learn the newer ones; the users would prefer to develop their own models for a number of very human reasons.

In any case, it appears that a real problem exists and something does need to be done. Woolsey (1972) has written a strong "expose" of the problem as it exists in operations research and management science. His suggestions for improvement cover professional societies, education and individuals. But the recurrent theme through all of his suggestions is that researchers, no matter how abstract their work, should have sufficient practical orientation in their backgrounds so that they will
recognize which of their findings have practical implications and what the problems of reduction to practice might be. Further, they should undertake to communicate these results to the appropriate audiences in much more readable and usable format.

It appears that the burden of closing the gap between researchers and users must fall largely on the researchers. It is they who have the desire and an inherent obligation to take the added steps necessary to insure that their results are made available in suitable form to those who have need of them. The CHOOSE system presented here assumes that the researchers will take the needed extra steps.

Note, however, that no methods of presentation or organization will help those users who--for a variety of reasons, ranging from laziness to incompetence, to a satisfaction that the methods to which they have become accustomed in the past will serve their future needs--are unwilling to learn of potentially better methods and the improved sources for retrieving the information when needed.

## Some Model Choice Schemes

Many papers present methods or procedures for model choice, each in its own rather limited context. Among those reporting such procedures are Kashyap (1971), Smallwood (1968), Young (1971). The decisions in these cases are often made on the basis of a numerical measure of effectiveness which is easily calculated. Kashyap (p.1) for example, divides model building into three parts:
(a) the choice of a number of a priori models with undetermined parameters;
(b) the estimation of parameters in them using the available set of observations; and
(c) the choice of a particular model from among the various models obtained above, according to a suitable criterion, and the check of the model to see whether it reproduces the desired statistical characteristics of the observed process.

Further, he states,
In the ideal case, we would like the output of the model to reproduce all the important statistical characteristics of the observed stochastic process such as the prediction error, correllegram, power spectrum, and the range-lag characteristic (the so-called $\mathrm{R} / \sigma$ graph) representing the average characteristics of the extreme values of the stochastic process. Since such a construction seems very hard at the present juncture, we will be content in constructing models which reproduce some of the important statistical characteristics of the observed process mentioned above which are relevant for the conditions under which they will be used.

This shows that Kashyap recognizes that the choice of a suitable model must be guided by its intended use. His subsequent development is based, however, strictly on comparing models on the basis of one attribute which is selected for model performance. We further note that his model choice procedures involve the mimicking of input-output relationships established through observation of the real system.

The problem presented in this study is broader than the one in the Kashyap paper because we also wish to consider those situations in which the data base does not contain sets of input-output information over a time scale. That is, we will be concerned with models for (1) situations with no present physical reality, that is, they exist only conceptually; and (2) existing real systems for which no past history is available, the only data available being those which can be measured directly or imputed from surveys of present conditions.

In addition, we note that, in the context of limited resources, it will often be necessary to make one model perform several jobs adequately, rather than to perform one job in some "optimal" way. It will not be convenient, in general, to have a separate model for each of several performance criteria of interest.

Smallwood (1968), in treating model selection as a decision-making process, outlines the following steps:

1. Formulate the requirements for the model.
2. Hypothesize a form for the model. This is the point requir ang $^{\text {a }}$ the most insight and creativity and the part which is least understood. At this point the model form constitutes a continuum of models--the "model space."
3. Affix parameters to the model form hypothesized. (Each set of parameters assigned or assignable to the model form creates a different model--a "point" in the model space.) This is done by a) assuming that a model of the hypothesized form has produced the data, b) assigning a cost function which specifies the cost of using a particular set of parameters for each possible value of the true parameters, and c) selecting the set of parameters which, for the present state of information, minimizes the cost of using the parameters. Here, Smallwood introduces the term metamodel to mean a model space of candidate models presumed to have included the true one.
4. Test and evaluate the models. If necessary, return to step 3 for additional parameter choices. If necessary, return to step 2 and postulate alternative forms for the model.

Clearly, Smallwood's approach also falls short of our needs because of the input-output data requirements.

Young (1971) offers an approach for choosing among systems when input characteristics are uncertain. He proposes using probabilistic inputs to evaluate by simulation all of the candidate models. Again,
the decision is based on some comparison of simulated outputs with observed outputs. As pointed out before, this is not always possible.

Sìmulation, usìng digital, analog, or hybrid computers, is a powerful tool and is used in many situations. It is sometimes preferred in solution of problems where analytic treatment is possible, but it finds its biggest uses in exploring system performance in those circumstances untreatable by analytic methods. However, the accuracy of results obtained by simulation using probabilistic inputs is often very limited; reproducibility of results within two or five percent often being difficult to establish. This occurs for a number of reasons, probably the most important of which is that it is impossible to generate a finite stream of truly random inputs. Finite streams can always be shown to exhibit some degree of sequential correlation. Attempts to reduce this problem often result in increasingly long computer runs. And, again, where probabilistic inputs are to be generated in an attempt to reproduce with some confidence the inputs which would be supplied by accident of nature or design of individuals, the selection of a process generator may become as important a part of the model choice process as the model structure itself. Simulation is, by nature, a brute force process which does not yield elegant results. It is not a sharp enough tool for many decisions.

It, therefore, seems that, for the most part, it would be impractical, as Young proposes, to evaluate a great number of candidates by the tedious process of simulation with the expectation of reaching clear cut choices.

To summarize previous comments, the three procedures described miss the mark on several counts. In the first place, we are not so much interested in postulating new model forms or structures as we are in examining critically the model forms postulated or already in use for particular fields.

Second, we will be looking at situations where comparisons of model performance with data may not be possible. These cases will arise in situations where model structures must be selected and parameters assigned on the basis of the simplest "present state" information which has been gathered on site. There may be neither time-indexed input information nor time-indexed output information.

Third, the combinatorial problems would appear to make the detailed examinations, which are required by Smallwood and Young, unreasonably time consuming.

Fourth, we will, in most cases, be unable to develop a simple one dimensional decision criterion, whether it be to optimize a single attribute or a one dimensional objective function made up by some convenient weighting of a set of single dimension attributes.

In other model choice problems, questions arise with respect to the method of generating new model structure ideas. This is quite appropriate in many fields where the underlying mechanisms of system
behavior are not understood. Mylks (1971), for example, created a framework for interactive generation of models for improving health delivery in urban areas. His thesis represents a rather formalized way of "brainstorming" to produce additional model forms. Again, however, the concern in this thesis is not towards the methods of creating new conceptual models or new insights into internal behavior. Rather, it is concerned with improving the selection of model structures from the vast library of those made available by research in recent years.

Another aspect of the modelling question was introduced by Wymore (1970, pp. 1/13-1/15). He takes a broader view of model choice, stating that the model should be as large and all encompassing as the resources available permit. His thesis is that models have a way of growing; posing new questions to the investigator as work progresses. It is, therefore, useful to have the model as broadly based as possible so that it can take on the added responsibilities without substantial revision. This makes one of the important criteria for model selection: is it a model on which one can build, one which can grow with changing needs?

## Model Structure Choice in Hydrology

While the preceding material has referred to the general model choice question--including that of selecting model parameters--only a restricted part of the problem will be treated here. For the reasons given in Chapter 1 Step 2, Model Structure Choice (page 6) has been selected as the most appropriate for treatment in this paper. The problems of model parameterization are quite distinct from the model structure choice problem and have an extensive literature.

The problem of model choice now seems to be particularly acute in the field of hydrology since researchers in hydrology were among the first to see great possibilities in a mathematical reorientation of their activities. Their frontal attack and a tentative assessment of the results was put very succinctly by Dracup, Mobasheri and Cardenas (1970, p. iii) as follows:

There is probably no precedent in the field of water resources for the manner in which researchers have seized upon and applied the techniques of systems analysis to water resource problems. The realization that there were now available recently-developed mathematical techniques which might be applied to "optimally allocate limited resources among competing activities," resulted in an almost frantic drive on the part of researchers to apply these techniques. . . . The results, after almost two decades of work, have been less than satisfactory.

The greatest danger may arise from not using formal models at all, but simply settling for the simplest of mental models of the kind that arise from a brief "man on horseback" type of survey.

One dramatic example of inadequate water resource planning of this type took place in India. Akbar selected Fatehpur Sikri as the site for a new capital which he founded in 1569 beside a lake 20 miles from Agra. The city of 200,000 had to be abandoned within 16 years for lack of an adequate water supply. The lake was completely dry. The area now supports a population of a very few thousand, most of whom are involved in maintaining the palaces, mosques and other structures for visitors. Here the lack of adequate insight into the water problems led to the waste of a tremendous amount of human resources (Time, 1973).

In approaching the model choice question, it does not seem desirable to try to organize hydrologic models into something equivalent to the periodic table of the elements or the evolutionary structures used in biology. There is no single thread of relationships; there is merely an n-tuple of attributes. Rather, it seems more appropriate to group models into the most broad classification possible. Examples of such broad classifications are watershed, stream pollution, lake pollution, reservoir, thermal pollution and groundwater models.

Once a classification scheme is selected, certain attributes can be listed for each of the specific models. This information, including data on whether the specific model possesses a certain attribute (and, perhaps, to what degree), can be developed for each model.

Models must be accepted or rejected on the basis of usefulness in providing insight into the problem at hand. This places model choice in the context of use rather than in some attempted generalized measure of effectiveness at a higher level of abstraction. But models are in some sense like life insurance policies. No one policy is suitable for all clients. Some method is needed to educate the discretion of the CHOOSERs so that the choice they make will be near optimal in the sense of their particular requirements.

In order to talk more precisely about the problem, one example from the field of hydrology--watershed models (including rainfall-runoff models)--has been selected below for more detailed discsusion. This class of hydrologic models comprises those associated with surface water flow in natural or man-made channels. It has been rather traditional to refer to these in two broad classifications. The term streamflow model seems to refer most often to models which generate flow information on the basis of statistical projections from past performance (Fiering, 1967). The term rainfall-runoff models applies to event-based models which accept some input event and predict the output on the basis of that specific input. See, for example, Fogel, Duckstein and Sanders (1971); Yakowitz and Denny (1973).

There are many watershed models of both types. Almost every new issue of many of the hydrologic journals discloses some new model for a particular application. Over 20 such models are codified in the report by the World Meteorological Organization (1972). An example of one of them is given in Figure 8 on page 80 . The proposal under which the present research was conducted (Kisiel, Qashu and Duckstein, 1970) contains an extensive bibliography with references leading to many models such as the
following: the Stanford model, the Purdue model, the Soil Conservation Service family of models, various models involving instantaneous unit hydrographs, streamflow synthesis models, the University of Kentucky model, and non-linear models such as the Chiu-Huang model.

All of the above classifications represent families of models which under different choices of parameters yield essentially different detailed models. However, the first problem facing the potential user is some way to locate the models of potential value to him, and to select from those the appropriate model structure. All of this should take place before parameterization is considered.

What seems to be needed, then, is some new effort, new marketing, new direction, or new emphasis which will get the required information into the hands of potential users promptly and without difficulty. Wholesale re-education of users should not be required. They should be able to move, almost effortlessly, simply on the basis of the existence of suitably organized information to the specific candidates (results, model, or whatever) that offer the most promise for their specific requirements. All of this, of course, is based on the premise that no one model structure is so all encompassing and pervasive that it completely dominates the others. It also presumes that no such dominant model will be developed in time to solve the present problem.

In seeking some sort of framework to organize information regarding watershed models from the user's point of view, it would seem appropriate to organize the information in a finite number of performance oriented attributes. This could lead to such user oriented attributes (statements and/or questions) as follows:

1. Mode1 Performance
a. Predicts timing of flood peaks
b. Predicts volume of discharge
c. Predicts complete runoff hydrograph
d. Predicts 24-hour (monthly, annual) runoff, etc.
2. Input Requirements
a. Area survey information
b. Slope estimates or survey
c. Infiltration model to be incorporated
d. Estimates of roughness coefficients
e. Rainfall records
f. Runoff records
g. Rainfall input (complete hyetograph)
h. Time scale
3. Resource Requirements
a. Funding
b. Skilled personnel
c. Computer
d. Time schedule

Clearly, some of the questions (attributes, queries) will lend themselves to complete answers by a binary choice; that is, yes or no, the model "does" or "does not" have the stated property: Others, however, will require answers which somehow indicate the degree or level of response assignable to the question. In some cases the response may represent the "degree of belief" which a binary question has as to whether it will take one or the other of the two values. In any case, the questions must be posed in such a way that some merit ordering is possible for each of the answers.

With this background in hydrologic model structure choices, the discussion can now return to a general statement of the design considerations for the CHOOSE system.

## Design Considerations for the CHOOSE System

As noted in Chapter 1 the "CHOOSE system" refers to the complete set of concepts and facilities to be designed for attacking the choice problems defined above. The guidelines used in designing such a system can be summarized briefly as follows:

1. The final system should be something that users or potential users could actually be persuaded to try: something that might actually appeal to those facing the problem as a "good" way to approach it.
2. It should be something that the user could try with very little individual preparation. This means the system selected should almost be self-teaching.
3. The format should embed the required information in the language of the interests and portable knowledge of the user.
4. It should be something the user could work with, set aside, work with again at his convenience, etc., until he was satisfied with the result.
5. The system should be organized so that it could be easily updated by the research organization implementing it. It should be easy to add new candidates, readjust attribute information on individual candidates based on experience or new findings, or to add new classification and selection criteria.
6. Assuming a computer is to be used, the system should not require any significant knowledge of computers or computer programming by either the researchers or the users. Further, it should be economical in the use of central memory storage or execution time in the computer used.
7. The system should prevent, or tend to prevent, loss of appropriate candidates due to any of the following:
a. Oversight. No qualified candidate should fail to be brought to the potential user's attention.
b. Information shedding. No candidate should be lost, ignored, or otherwise neglected simply because the capacity of the user to learn about and retain knowledge of a large number of candidates is limited. The typical human response to an information overload, information shedding (filtering or omission) (Sheridan and Ferre11, 1974, pp. 133-134), is to be avoided.
c. Premature rejection. In many selection processes, many candidates are rejected for further consideration on the basis of an early screening. They fail to meet some criterion considered minimal at that time for the problem under study. The problem here is that, as goals, constraints, minimum levels of satisfaction, etc., for each of the attributes evolve during an interactive learning process, certain candidates that were rejected early in the search process might again be worthy of consideration. These should be caused to reappear for consideration if the evolving selection criteria indicate.
8. Ideally, the framework should be applicable without change to a broad range of problems similarly situated. It should not be necessary to modify the basic framework or the computer programs to accommodate problems in different fields.
9. Costs should be reasonable and requirements for hardware and specialized personnel should be kept to a minimum.

With these thoughts in mind, the development of the CHOOSE system is covered in Chapters 3 and 4 following. Chapter 3 develops in detail the concept of the computational CHOICEMODEL, and Chapter 4 describes the implementation of the model so that the appropriate computations and interaction, when required, can take place.

DESIGNING THE CHOOSE MODEL
I realized also that the mind can neither contain all the knowledge it seeks, nor find solutions to all the problems it encounters.
al-Ghazali (ca. 1100)
The problem identified in earlier chapters is that of organizing and presenting specialized information in a way that will enhance its usefulness and create a broader range of application. Towards this end this chapter identifies the CHOOSE problem as occupying a place somewhere intermediate between mathematical programming formulations and information retrieval structures.

For the particular class of problems under study, we can abstract the essence of the problem presented in Chapter 2 in the following form:

1. The basic classes of problems under consideration are those for which a substantial amount of work has been done (or can be done) producing a large set of candidates (models, individuals, etc.) for possible selection.
2. The candidates can be evaluated and compared on the basis of intended application by a mixture of objective descriptions and subjective, but presumably valid, expert opinions.
3. Those choosing from the candidates must make their choices (or at least narrow their range of choice) without substantial effort. This implies that they are interested in making an optimal or near optimal choice based largely on evaluations of those whose opinions they are willing to accept with little question.
4. Those choosing have the ability and background to answer a set of questions specifically designed to direct their attention to the candidates most likely to succeed for their specific application. The questions asked need not be in detailed quantitative form.

In order to design a process for aiding choice makers it will be useful to have a framework for discussion of the ingredients of the problem. Such a framework is presented in a set theoretic notation using the definitions given at the end of Chapter 1.

A Set Theoretic Approach to the Choice Problem
There are many aspects to a choice problem. First, it is desirable to consider as many alternatives as practical limitations allow.

When considering prospective candidates, it is not always desirable or possible to know a priori that they are good enough to be chosen. Therefore a choice model should have some constructs for including such questionable candidates and for establishing their feasibility or acceptability at a later tìme.

Secondly, in selecting subsets of candidates not just any subset will do. The subset or individual candidate is usually chosen on the basis of some perception of the concept best. And since the candidates themselves do not often exhibit their own capabilities and limitations directly, it is necessary to create some measures of expected performance of the candidates with respect to perceived needs. This will sometimes require a mapping of candidate attributes to some objective space.

The final problem is to use the measures of expected performance in some way to make the best choices. The minimum rational basis for such a decision is that there are no choices available that are better than the one chosen--or those in the subset chosen. Mathematically speaking, this implies that the acceptable candidates are to be ordered by an ordering relation that directs the set; that is, one that contains an optimal element. However, if they are to be best, the concept of best--whatever that may mean to the CHOOSER--must be embedded in the selected ordering relationship.

Note that if the set of acceptable candidates has more than two members, there will be more than one ordering that can produce the same choice. Hence the ordering relationship associated with a particular choice is not unique. This simply reflects the concept that the minimum conditions for choice do not necessarily imply any position concerning the relative positions of other candidates in the choice structure.

Since the modelling of the choice problem implies that the choice is not known, it may be that none of the candidates are acceptable. A model of the choice problem should make provision for this contingency. Further, the act of creating a choice model does not insure that any best elements can be identified by the model. Hence, it will be desirable for the model to partition the set of acceptable candidates into two subsets; one for those that are maximal on the basis of the chosen ordering criteria, and one for those that are not. If a clear cut set of best choices is not defined by the model, then the choice process may begin anew with a second model to develop choices from those in the "maximal but not best" category.

A set theoretic framework, a CHOICEMODEL, in which to embed these ideas is defined below. As with other artifacts, however, it is up to the person (modeller) who generates the various sets to see that the sets of the artifact do provide a valid description of the real choice problem.

## Definition: A CHOICEMODEL $C$ is a 4-tuple of sets,

 $C=\overline{(C, g, f, \alpha)}$, where1) C is a set not empty;
2) $\mathrm{g} \varepsilon$ FUNCTIONS (C, $\{$ NOTACCEPTABLE, ACCEPTABLE $\}$ );
3) $\mathbf{f} \varepsilon$ FUNCTIONS $\left(C^{\prime}, F\right)$, where $\mathrm{C}^{\prime}=\{\mathbf{x}: \mathbf{x} \varepsilon \mathrm{C}, \mathrm{g}(\mathbf{x})=\operatorname{ACCEPTABLE}\}$, and $F$ is a set not empty; and
4) $\alpha \in O R D E R I N G S(R A N G E(f))$.

The interrelationship between the definitions, notation and intuitive concepts involved are given in Figure 1. Only the elements of the 4 -tuple need be defined in order to establish $C$. The elements $C^{\prime}, C^{\prime \prime}$ and $C^{*}$ defined in Figure 1 are implied in the model definition and are exhibited only after the computations embedded in the model are performed. Note particularly that the method of definition requires $C^{*} \subset C^{\prime \prime} \subset C^{\prime} \subset C . \quad$ If $C^{\prime}$ is empty, no choice is possible. There is no candidate that meets the requirements established by $g$.

If $C^{\prime}$ is a finite set then certain special properties arise. If $C^{\prime}$ is not empty, then $C^{\prime \prime}$ cannot be empty since every ordered set has maximal elements. In fact, every element may be a maximal element. If $C^{\prime \prime}$ has only one member, then $C^{*}=C^{\prime \prime}$, since a single maximal element will also be an optimal element. However, if $\mathrm{C}^{\prime \prime}$ has more than one member, there is no assurance that C* is not empty. C* will have members only if all the members of $C^{\prime \prime}$ are equivalent as well as maximal; that is, for all $x, y \in C^{\prime \prime}, \alpha(x, y)=T R U E$ and $\alpha(y, x)=T R U E$. This is equivalent to saying that if $\mathrm{C}^{*}$ is not empty, it is the same set as $\mathrm{C}^{\prime \prime}$.

The set F is a convenient intermediate set that need not be formally defined; all that is necessary is that $\mathrm{F} \in \operatorname{RANGE}(\mathrm{f})$. But sometimes it will be convenient to define it in some broad way. For example, in the multiple objective mathematical programming problem of the next
section, it will be convenient to let $F=R^{n}$. At other times it will be convenient to define other auxiliary sets to simplify the overall description. In particular, it will be useful to define a set $A$, an attribute space into which the elements of $C$ can be associated with attributes meaningful for the choice problem at hand. These considerations will become more clear with the examples to follow.

## Mathematical Programming Models

Classical optimization techniques have a lot of appeal since they do lend a mathematical elegance to decision problems amenable to such treatment. Classical optimization encounters some difficulties when multiple criteria are involved, and thus gives way to the concepts of goal programming where objectives, constraints and optima become aspiration levels, goals and satisfactums.

The mathematical programming approach consists of quantifying the various considerations and combining them in mathematical equations or inequations, and then "solving them" so that the optimal choice or "program" emerges.

| Concept | Denotation | Symbol |
| :---: | :---: | :---: |
| The set of candidates or prospective candidates for $C$ | CANDIDATES ( C ) | C |
| The considerations for feasibility for members of $C$ | constraints (C) | g |
| Feasible or admissible candidates of $C$ | OKCANDIDATES (C) | $C^{\prime}$ |
| Objective functions or goal functions of $C$ | goalfunctions (C) | f |
| The set on which the quality of goal attainment for $C$ is to be measured | GOALMEASURES (C) | F |
| Choice criteria for C | choiceordering (C) | $\alpha$ |
| Choices made by C | CHOICES (C) | $C^{\prime \prime}=\left\{x: x \in C^{\prime},\right.$ <br> $f(x)$ is maximal <br> with respect to $\alpha\}$ |
| The best choices made by $C$ | BESTCHOICES (C) | $C^{*}=\left\{x: x \in C^{\prime}\right.$ <br> $\mathrm{f}(\mathrm{x})$ is optimal <br> with respect to $\alpha\}$ |

Fig. 1. CHOICEMODEL Equivalences

In the notation defined at the close of Chapter 1 , let $n, j \varepsilon I^{+}$, $\mathrm{m} \varepsilon \mathrm{I}^{++}, \mathrm{g} \varepsilon \operatorname{FUNCTIONS}\left(\mathrm{R}^{\mathrm{n}}, \mathrm{R}^{\mathrm{m}}\right) ; \underline{\mathrm{b}} \varepsilon \mathrm{R}^{\mathrm{m}}, \mathrm{F}=\{\underline{\mathrm{x}}: \underline{\mathrm{g}}(\underline{\mathrm{x}}) \leq \mathrm{b}\}$, and $\ddagger \varepsilon \operatorname{FUNCTIONS}\left(F, \mathrm{~F}^{\mathrm{j}}\right)$. Then the classical optimization problem, if $F$ is not the empty set, is to find $\underline{x} \varepsilon F$ such that $\underline{f}(\underline{x})$ is optimized in some sence. (As will be noted later, the "sense" in which optimization is to take place is a key issue.) Such a problem is spoken of as having $j$ objectives, $m$ constraints and $n$ decision variables. The function $\underline{f}$ is called the objective function. The $g$ and $b$ taken together define the constraints of the problem. They constrain the values of $\underline{x}$ to the admissible of feasible region $F$.

If $j=1$, the problem is one of single objective optimization--a maximization or minimization problem--of the type that has been studied for many years and has an extensive literature. See, for example, Chapter 3 of Wilde and Beightler (1967). Certain specialized single objective problems can be defined in the same framework. If the $n$ decision variables are to be non-negative, then the domain of $g$ can be defined as $\left(\mathrm{R}^{++}\right)^{\mathrm{n}}$ rather than $\mathrm{R}^{\mathrm{n}}$. It is probably more in keeping with the way computation is usually organized, however, to modify the domain of $\underline{f}$ by substituting $F \cap\left(R^{++}\right)^{n}$ for $F$. For integer programming problems with non-negativity the domain of $\underline{f}$ becomes $F_{\cap}\left(I^{++}\right)^{n}$ rather than $F$.

If $j>1$ the problem is one of multiple objective optimization. This adds a new dimension to the problem both literally and figuratively. The problem of multiple objective optimization, compromise or trade-off is as old as man. However, its intensive investigation in a mathematical framework is comparatively recent, and the results are not altogether satisfactory. The added problem is that it is no longer adequate to speak of optimization as a maximization or minimization problem: the objective function no longer produces a measure of optimization success but simply a vector of values in which the meaning or sense of optimization is no longer clear. The problem becomes one of somehow deciding what is best, i.e., what is the proper basis for optimization. An overview of the topic and a compilation of selected papers is given in Cochrane and Zeleny (1973).

Before discussing in detail the problems arising when multiple objectives are sought, it will be convenient to express the above concepts in the form of a CHOICEMODEL.

The Mathematical Programming CHOICEMODEL, M
In the CHOICEMODEL framework the concepts of a mathematical programming model can be defined as follows:

Definition: Let $j, n \varepsilon I^{+}, m \varepsilon I^{++}, h \varepsilon \operatorname{FUNCTIONS}\left(\mathrm{R}^{\mathrm{n}}, \mathrm{R}^{\mathrm{m}}\right)$, and $\mathrm{b} \varepsilon \mathrm{R}^{\mathrm{m}}$. Then a mathematical programming model, $M$, is a 4-tuple ( $C, g, f, \alpha$ ), defined as follows:

```
\(C \subset R^{n} ;\)
\(\mathrm{g}=\{(\underline{\mathrm{x}}, o k ?): \underline{\mathrm{x}} \varepsilon \mathrm{C}, \mathrm{ok} ?=\operatorname{ACCEPTABLE}\) if \(\mathrm{h}(\underline{\mathrm{x}}) \leq \mathrm{b}\),
    ok? = NOTACCEPTABLE if \(\mathrm{h}(\underline{\mathrm{x}})\) £ \(\underline{\mathrm{b}}\);
\(f \varepsilon \operatorname{FUNCTIONS}\left(C^{\prime}, F^{j}\right)\), where \(C^{\prime}=\{\underline{x}: \underline{x} \varepsilon C, g(\underline{x})=\)
    ACCEPTABLE\}; and
\(\alpha=O R D E R I N G S(R A N G E(f))\).
```

Note that the above model，like $C$ ，is still largely framework．It will not be＂computable＂until each of the four sets，$C, f, g$ and $\alpha$ are specifically defined for the particular problem．The above simply defines enough of the elements of the 4 －tuples so that the completely general framework is reduced to one of qualified generality for mathematical programming problems．

As a further example，consider the following model of a single objective linear programming problem with non－negativity constraints：

Definition：Let $n \varepsilon I^{+}, m \varepsilon I^{++}$，$A$ be an $m x n$ matrix of real numbers，$F$ be a $1 \times n$ matrix of real numbers，and $b \varepsilon R^{m}$ ．Then a linear programming model with non－negativity constraints，$F$ ，is a 4－ tuple（ $C, g, f, \alpha$ ）defined as follows：

$$
\begin{aligned}
& C=\left(\mathrm{R}^{++}\right)^{\mathrm{n}} ; \\
& \mathrm{g}=\{(\underline{\mathrm{x}}, \mathrm{ok} ?): \underline{\mathrm{x}} \varepsilon \mathrm{C}, \mathrm{ok}\}=\operatorname{ACCEPTABLE} \text { if } \mathrm{A} \underline{\mathrm{x}} \leq \underline{\mathrm{b}}, \\
& \text { ok? = NOTACCEPTABLE if A } \underline{x} \text { 立b }\} \text {; } \\
& f=\{(\underline{x}, F \underline{x}): \underline{x} \varepsilon C, g(\underline{x})=A C C E P T A B L E\} ; \text { and } \\
& \alpha=\{((x, y), z): x, y \in C ; g(\underline{x})=g(y)=A C C E P T A B L E ; \\
& \text { if } f(\underline{x}) \leq f(\underline{y}), z=T R U E ; \text { if } f(\underline{x}) \text { 立 } f(\underline{y}), z=\text { FALSSE }\} .
\end{aligned}
$$

If the feasible set is not empty such a model always has a unique best value．In certain degenerate cases there may be more than one candidate that produces that best value．Here the definitions of A， F and b give a linear programming problem its specific detail．The distinction between parameters of the model and inputs to the model－－ from a computational point of view－－is not important in this case．In some cases，however，the definitions of $A$ and $F$ ，the constraint matrix and objective matrix，respectively，would be quite basic to the problem and would be considered parameters，whereas the amount of resources available，$\underline{b}$ ，would be the variable input that the decision maker would use to explore the behavior of the decision．The output would be the selected $\underline{x}$ and the $F \underline{x}$ associated with it．

## Additional Considerations in Multiple Criteria Problems

Much of the original work in multiple criteria cases was in devising ways of converting multiple objective functions into a single objective function, and then proceeding as in the classical $j=1$ cases. In other approaches, a hierarchy of objectives was used so that optimization could first take place over a primary objective, and if any adjustment of decision variables remained, the attempts would be made to optimize the second criterion, etc. The development of these concepts is thoroughly documented by Johnsen (1968).

It soon became apparent, however, that other approaches were needed. In some cases very small reductions from optimum for a primary objective could sometimes produce valuable improvements in a secondary objective. This produced the problem of determining marginal substitution rates for the various objectives. (But note that if substitution rates are constant over the entire feasible region, the problem could have initially been stated as one with a single objective function.) But marginal substitution rates are often elusive, subjective, investigator dependent, time dependent, and otherwise inaccessible to the modeller/analyst. While there may be reasonable agreement among various modellers and analysts on the formulation of the various objective functions and constraints, the relative importance of the various objectives under different proposed policies (choices, decisions) can only be established by the decision makers. But it is not realistic to ask the CHOOSER to define his preferences throughout $F$ a priori so that the optimization can be conducted as a predetermined experiment. The CHOOSER will generally find it impossible to articulate his own preferences for the large region of $F$ where no feasible solution is likely to be found. In fact, while $F$ may be well defined mathematically, a substantial amount of computation may be required simply to present $F$ to the CHOOSER in a framework in which he can readily react. Hence, if a completely predetermined experiment is sought, it may turn out that the CHOOSER has the added burden of supplying his marginal preferences for alternatives outside of $F$.

For the above reasons, most of the multiple objective optimization schemes that have been implemented avoid predetermined experiments in favor of adaptive experiments; that is, they require input information more than once as the experiments proceed. "In this connection, however, there arises another problem: the problem of successfully organizing the 'analyst-decision maker' dialog. For psychological reasons, the questions put to the decision maker have to be clear and sufficiently simple. The number of these questions as well as the number of 'analystcustomer' dialogs have to be minimized as far as possible (Benayoun and others, 1972, p. 1258)."

Important developments have taken place in recent years both in determining marginal substitution rates and in organizing the dialog. Geoffrion (1967) worked extensively with the $j=2$ case for which tradeoffs can be diagrammed quite easily. Boyd (1970) developed a technique for determining the decision maker's marginal substitution rates for more general cases. Monarchi (1972) developed a computerized scheme for
getting decision maker's preferences dynamically at teletype or cathode ray interactive terminals. Other computer oriented schemes have also been presented; e.g., 1) the STEP Method of Benayoun and others (1970) treats the linear programming problem (where $f$ and $g$ are linear combinations of decision variables) for $j>1,2 \overline{\text { I }}$ the Méthode Électre II of Roy and Bertier (1972) uses ordering relations to determine the regions of concordance and discordance in judgment among a group or panel of decision makers.

Dyer (1973) developed a time-sharing program for attacking multiple objective problems in $\mathrm{R}^{\mathrm{n}}$ space. The particular feature of Dyer's method is that he organizes the man-machine dialog in such a way that only ordinal preferences (or indifferences) must be reported by the decision maker. The computer then takes over the problem of assigning numerical values to the preferences thus obtained. The decision maker retains some control over the size of the decision problem by controlling the step size. The computer program, however, manages the whole process of moving stepwise, in the directions determined in the preference evaluation.

Although Dyer's technique is not applicable to the present problem for reasons given below, his insight into the problems of eliciting information from a decision maker in a time-sharing environment is directly applicable. In particular, he cites Torgerson (1958) as arguing that ordinal comparisons are much less demanding on the decision maker than point estimates. The ease of use by the decision maker is an important element in making the time-sharing process successful. Dyer also refers to a study (Feinberg, 1972) which indicates that "the tradeoffs obtained from a series of ordinal comparisons are more accurate than those obtained from point estimates."

All of the above work has been done in some continuous subset of the decision space $\mathrm{R}^{\mathrm{n}}$. In addition, each of the component numbers of the n-vector $\mathrm{x} \in F$ has the customary meanings associated with numbers. They are numbers that measure things and can be combined meaningfully with elementary arithmetic and algebraic operators into either objectives or constraints. They are not simply numerical symbols representing nonnumeric concepts. Taxonomies of many of the multiple objective concepts
for seeking optima (largely in $\mathrm{R}^{\mathrm{n}}$ space) appear in Roy (1971) and MacCrimmon (1973).

And, of course, all of the above assumes that some feasible solution exists in the first place. That is, the set of feasible solutions is not empty; all constraints are met even though the satisfaction level of the available choices may not be very high. There will be problems, naturally, for which this is not true. Techniques for finding these and knowing when to seek other alternatives are also needed.

## Information Retrieval Models

Quite different from mathematical programming is the concept of automated information search and retrieval, where the'typical library system of cataloging (card indices of titles, authors and subjects) has been supplemented by computer based schemes. Some are quite specialized in that the hierarchy of coding information is developed for a specific field of inquiry. For example, a substantial effort has been made to code library type materials in the field of arid land studies in a specialized computer file at The University of Arizona (Zeitler, 1970). More generally, the schemes involve key words identified with the published material or some sort of computer analysis of the text itself. Whether the text is coded by librarians with appropriate backgrounds or by detailed computer analysis of the text, results attainable in recovery of materials by such retrieval schemes are limited by trade-offs between recall and precision. A high percentage of recall (recovering all the related material) carries with it a low precision (recall of excess amount of material not useful). Salton (1970) reported that recalls of $50 \%$ and precision of $60 \%$ were attainable using these schemes. He feels that interactive searches can produce improvement.

One of the most fruitful ways of upgrading retrieval performance consists in using multiple searches based on user feedback information furnished during the search process. Interactive search methods should then lead to a retrieval effectiveness approaching a recall and precision of about 0.70 . . . [Emphasis supplied].

Even if the improvements envisioned by Salton are attained, the results are not particularly attractive for the present problem. The selection of documents is not in itself the choice or decision desired, and the high rate of selection of inapplicable documents would seem to compound rather than ameliorate the problem. However, Salton does suggest what is required to improve the results.
. . . a retrieval system is designed to serve a large, sometimes heterogeneous user population. Since users may have different needs and aims, and since their search requests may range from survey or tutorial type questions to very detailed analytical queries, an excessively specific analysis may be too detailed for most users.

While "an excessively specific analysis" may be too detailed for most users, in the present case, when the users constitute a well defined group the specific analysis concept is appealing. A CHOICEMODEL for one such scheme follows:

## A CHOICEMODEL I for Information Retrieval

The CHOICEMODEL I for information retrieval is based on the concept of a single computation based on input supplied by the CHOOSER. To use
such a model iteratively, it must be embedded in a dynamical system so that the model computations can be accomplished as many times as necessary.

One method for information retrieval is based on identifying each document (book, manuscript, journal, letter, audio tape, video tape, motion picture film, or whatever source of information it is of interest to catalog) with. certain key words that relate to the contents of the document. For example, the proposal for the research grant under which the larger part of this study was done is identified by the following set of key words: validation, time series analysis, systems analysis, computer simulation, network design, water resource management, stochastic processes, statistical inference, cost-benefit analysis . A document search can then be established on the basis of key words that the CHOOSER identifies as closely related to his needs. A document that is associated with several of the key words of CHOOSER interest is likely to be pertinent. It may also be assumed that the lack of certain key words would make the document useless. The number of key words directly associated with the CHOOSER's inquiry might, in fact, act as some measure of the likelihood that the document is pertinent.

A CHOICEMODEL representation, $I$, of this is presented below. Each of the elements is defined after some discussion of the underlying rationale.

The candidate set can be as broadly described as desired, although there is no assurance that all the documents described will actually be coded and actually be present in the model. Further, there is no need, from the mathematical description point of view, to attempt to limit the number of candidates by prescreening in any way.

Hence

$$
\begin{aligned}
C= & \{x: x \text { is any reproducible document considered of } \\
& \text { potential value in the treatment of the present } \\
& \text { problem }\} .
\end{aligned}
$$

An intermediate attribute space will be notationally convenient. Therefore let $A=\{k: k$ is a key word associated with any member of C$\}$. The desired attribute space, A, then will be the set of all subsets of A. In this way the subset of key words associated with any $\mathrm{x} \varepsilon \mathrm{C}$ will be the attribute of interest. Thus $A=\{a: a \subset A\}$.

To identify each of the candidates with the appropriate subset of key words is a task usually assigned to the writer of the document (as in the above example), but it can be done by catalogers or others. In any case the information retrieval scheme, if it is to be of any immediate use to a CHOOSER, must have the function $c \in \operatorname{FUNCTIONS}(\mathrm{C}, \mathrm{A})$ defined and embedded in the model in advance.

Suppose now that the only documents that the CHOOSER considers worthy of retention in $C^{\prime}$ (i.e., admissible or acceptable) are those containing two particular key words. Let the subset of $A$ (i.e., the element of A) containing only those two words be identified as $a_{2}$.

Then

$$
\begin{aligned}
\mathrm{g}= & \left\{(\mathrm{x}, \text { ok }): \mathrm{x} \in \mathrm{C}, \text { ok? }=A C C E P T A B L E \text { if } \mathrm{c}(\mathrm{x})=\mathrm{a}_{2},\right. \\
& \text { ok? } \left.=\text { NOTACCEPTABLE if } \mathrm{c}(\mathrm{x}) \nmid \mathrm{a}_{2}\right\} .
\end{aligned}
$$

and, as always, $\mathrm{C}^{\prime}=\{\mathrm{x}: \mathrm{x} \varepsilon \mathrm{C}, \mathrm{g}(\mathrm{x})=$ ACCEPTABLE\}.
Suppose now that the CHOOSER decides that the number of certain additional key words of particular interest to him would constitute a useful score of best. Best in this context will mean that the documents with the highest number of selected key words are considered to be most likely to be of direct interest to the CHOOSER. In this sense it constitutes an "expected value" of usefulness. Let $a_{10} \varepsilon A$ be such a subset of key words; i.e., a subset of $A$ containing 10 elements. Then

$$
\begin{aligned}
f= & \left\{(x, z): x \in C^{\prime}, z=\#\left(c(x) \cap a_{10}\right)\right\} ; \text { and } \\
= & \left\{((x, y), z): s, y \in C^{\prime} ; \text { if } f(x) \leq f(y), z=T R U E ;\right. \\
& \text { if } f(x) \notin f(y), z=F A L S E\} .
\end{aligned}
$$

There the CHOOSER has depended upon a substantial amount of prior work by others to define all aspects of the choice model, $I$, except $a_{2}$ and $a_{10}$. Implementation of the model consists of designing and building a physical system, $z^{R E A L}$, that accepts the required inputs, performs the computations implied by the model, and produces the desired outputs. If the model $I$ has been implemented, the CHOOSER need only define $a_{2}$ and $a_{10}$ at the input of the device and receive his list of acceptable candidates simply ordered by his criteria. If he is not satisfied with the results obtained, he can reenter the system with different constraints or objectives to obtain new information.

## Moving Towards CHOOSE

Only a small subset of the world's decision or choice problems lend themselves to the numerical treatment required for mathematical programming. Many problems, the present one of model choice included, deal with concepts and images which must be expressed largely in non-numeric symbols and functions defined on these symbols. If such symbols cannot be mapped into a suitable numeric space and prepared for computation in the familiar arithmetic of real numbers, they cannot be treated by the mathematical programming methods lescribed above. However, the mathematics of arbitrary sets has been developed extensively the 1st few decades, and there are many symbolic mathematical techniques that can be considered. The challenge is to see to what extent the properties of numbers can be enlisted so that mathematical programming techniques can be of some assistance in solving the problem.

Aspiration Levels, Satisfactums, and Goal Programming
In attempting to find some intermediate position between mathematical programming and information retrieval schemes some consideration of the role of goals, aspirations and objectives is needed.
"It belongs to an educated man to seek such certitude in each thing as the nature of the thing allows (Aristotle, in Saucedo and Schiring, 1968)." Since the ingredients of the multiple criteria problem are so elusive, it is frequently desirable not to push the optimum concept too far. Continuing the search for optima beyond the point where the preciseness of the mathematical formulation is meaningful is a mathematical exercise rather than a problem solving exercise.

The approach taken now is to reach some satisfactum; a solution that meets all constraints and is as close to an optimum as one can reasonably expect to come within the time available for analysis, given the uncertainties of the evaluation techniques. The process usually involves examination of a sequence of changing "aspiration levels" for the various objectives so that the resulting mix of feasible aspiration levels is acceptable to the decision maker. Such a practice often involves adjusting some goals in attempts to see what influence such changes have on other goals. Hence the term goal programming is also used.

Eilon (1972) points out that goals and constraints become almost interchangeable under these conditions. If. an attempt is made to maximize one of several objective functions at the expense of the others, then one or more of the others may fall below acceptable levels. To overcome this it becomes necessary to convert these objectives into constraints in order to see that they are met at some minimum acceptable level. In the same way, if what was initially an important objective is satisfied well above some optimistic aspiration level, some reduction of that objective may be entertained if it will lead to significant improvements in others. In this case the original objective may become a constraint in that a limit may be established on how low it will be permitted to go. As will be seen later, some advantage is taken of this interchangeability of goals and constraints in the CHOOSE procedures.

Candidates, Attributes and Objectives
As noted by MacCrimmon (1973, pp. 1-2) there is a fundamental distinction between attributes of candidates and the ability of candidates to meet objectives. In the context of multiple objective and multiple attribute decision problems, his definitions are as follows:

Multiple attribute decision problems deal with choosing among a set of alternatives which are described in terms of their attributes. Choosing among automobiles described by such attributes as initial cost, size, horsepower, acceleration, and fuel economy is, for example, a typical multiple attribute decision problem. Most of the techniques available for dealing with these multiple attribute situations require information about (a) the decision maker's preference among
values of a given attribute (how much does he prefer a 2 -mile per gallon fuel saving over a 5 -mile per gallon fuel saving) and (b) the decision maker's preference across attributes (how much more important is cost than acceleration). The multiple attribute techniques either directly ask the decision maker for an assessment of the strengths of these preferences or they infer them from his past choices.

Multiple objective decision models, on the other hand, recognize that attributes of alternatives are often just means to higher ends--the decision maker's objectives. These techniques, then, require (a) preference information about the decision maker's objectives and (b) information about the instrumental relationship between objectives and attributes. Preferences among attributes are thus derived from the preferences among objectives and the functions relating attributes to objectives. In multiple objective models an alternative can be described either in terms of its attributes or in terms of the extent to which it achieves the objectives of the decision maker. In our automobile example, the decision maker's relevant objectives may be financial security, prestige, dependable transportation, etc. A multiple objective model would require priorities on values of financial security versus prestige and also the linkages which relate the extent to which cost, size, acceleration, etc., contribute to prestige.

In summary, the attributes are the things that can be used to describe the candidates whereas the objectives are those end requirements that must somehow be satisfied. Presumably there exists some mapping between attributes and objectives so that the performance objectives can be stated in terms of the attributes. Once objectives are analyzed in some way then a backward mapping leads to the selected candidate. This leads to a generalized relationship which can be illustrated by the following mapping:


The arrows to the right indicate those mappings which must be defined in creating the choice framework. The mappings to the left show the steps that must be taken after some objective is selected in order to determine which candidate is indicated by that choice.

In the mathematical programming CHOICEMODEL $M$, the generalized procedures can be considered as being compressed to the following form:


The implication here is that rather than the set of available candidates giving rise to attributes, the attributes themselves over which the objective choice is made do in fact constitute the candidate set directly. The backward mapping stops when the policy vector (attribute vector producing the optimal as measured by the objective) is decided upon. The corresponding candidate is presumed to exist. For example, in Dyer's model (1973) the choice is made strictly on the basis of numerical attributes developed by the CHOOSER. In his example of developing marginal substitution rates in the choice of an automobile, he considers that such trade-offs exist, for example, between horsepower and gasoline mileage in a continuum and that the problem is solved once the horsepower and gasoline mileage are selected. No provision is made for the final backward mapping step to find an actual candidate that has the desired attributes.

In the information retrieval CHOICEMODELI a little different situation exists in that the candidates are mapped into an attribute space and the choice seems to be made by working directly with the attributes rather than any formally defined objective space. This is illustrated as follows:


The person attempting to retrieve information has to work in a framework which ends with attributes. He must search in terms of attributes, often with little knowledge of how his choices will influence his meeting his objectives.

Since it will not be possible to implement the basic CHOICEMODEL C in a completely generalized way, some compromises will be required to create the procedures to attack the present problem. Therefore an attempt must now be made to retreat from generality in a slightly different
direction than either of the two just described, but in a manner that allows the choice process to proceed for the important cases of the present problem. The retreat in this case will be an attempt to suppress the attribute space so that the following form arises:


The suppression of the attribute space is the key step, and it places the complete burden on the EVALUATORs of presenting all of the information to the CHOOSER in objective form. This means that researchers (EVALUATORs, applications engineers, etc.) must adopt this viewpoint and try to formulate all of their measures of candidate performance in terms of user needs rather than quantities that are convenient to measure and display.

## CHOOSE vs. Information Retrieval

In attempting to eliminate the attribute space it will be useful to see just which aspects of information retrieval concepts need to be carried forward into the CHOOSE model.

The user of a retrieval scheme is still required to devote considerable time to study, analysis and filtration of the useful documents retrieved; and to rejection of those retrieved that are not pertinent. The CHOOSER must create his own judgment criteria, evaluate his own requirements against those of competing candidates and generate his own rationale for selection.

The problems identified in Cahpter 2 indicate that this is too much to expect. Information retrieval schemes have been in existence for some time. Numerous key word files centered around hydrologic material have been developed. However, their chief use seems to be for educational purposes and for building foundations for future research. They are not as helpful to users or potential users of the concepts as many would like.

Even though the information retrieval and document search schemes have not proved extremely helpful, a substantial sharpening of the focus to specific choices could be attainable, as Salton (1970) suggests, by making the analysis extremely specific. Note, however, even though the concept of making an extremely specific analysis in an information retrieval context can be used, since the search will be for model structure rather than documents, the choice will not necessarily lead to journal articles or text books but may lead to packaged programs, mass storage files at various locations or data bases existing in almost any form.

In viewing the CHOOSE situation from a mathematical programming point of view three points can be made. First, there are multiple objectives/ constraints/goals in these choice problems. There will always be some need to balance resources of time, supplies, labor, capital investment, etc., against the insights which the model is expected to produce. All of these requirements must be determined by the CHOOSER; they cannot be decided by the EVALUATORs who make the detailed analyses of the items that are placed in the associated CHOOSE data base.

Second, in mathematical programming problems the choice or decision is made by selecting those feasible values of the decision variables which produce the desired result. Each decision variable is, within limits, presumed subject to the control of the decision maker, and each can be made independently. It is this property that makes them decision variables. However, in the present case, the decisions come in "packages"; with one choice all of the attributes (variables on which objective functions are defined) are selected. So the points in the domain space are restricted to those sets of points represented by the possible decisions. Even though you might like to make each decision individually, you must end up picking, in the present choice situation, one of the candidates that best (in some sense) meets the objectives.

Further reference to Dyer (1973) should clarify this. His illustration of eliciting trade-offs in the selection of an automobile considers three factors: cost, horsepower, and miles per gallon. His interactive dialog then produces a feasible mix of these three elements which represents the buyer's preferences. But automobiles do not exist in a continuum of prices, horsepowers, and fuel consumption rates. The problem remains, even after the satisfactum of the decision maker is found, to select the automobile to purchase. If the problem were the inverse, that of deciding what feasible combination of cost, power, and fuel consumption should be embedded in a new design, the situation would be different. In this case the decision would probably be made on the basis of a collection of a large number of opinions; in fact, several decisions are made in such cases resulting in a multiple product line.

Third, mathematical programming is based on numbers; whereas information retrieval schemes are generally developed on words and other nonnumeric symbols. Further, the numbers used in mathematical programming represent measures of the decision variables and objectives. They are not simply ordered symbols representing non-numeric concepts.

Many people, and particularly engineers and scientists, feel comfortable with numbers. Lord Kelvin (Beakley and Chilton, 1973, p. 235) summarized the attitude quite nicely with "If you can measure that of which you speak, and can express it by a number, you know something of your subject; but when you cannot measure it, your knowledge is meager and unsatisfactory."

Since the CHOOSE model is conceived as a blending of the more tractable elements of mathematical programming and information retrieval,
the challenge here will be to find symbols having some of the useful properties of numbers so that some of the mathematical programming techniques can be used.

The things that seem to be necessary to move towards CHOOSE from information retrieval and mathematical programming schemes in a way that suppresses the intermediate attribute space are the topics of the next three major sections.

Selecting the Directions for Compromise
Oliver Wendell Holmes (Highet, 1963) is said to have noted that "no generalization is worth a damn; including this one." This is undoubtedly true here. The generalized model choice framework does not really display a mechanism for solving the problem. If the mechanism for performing the required calculations is not embedded in the model itself, then the model cannot be computed and the results are not attainable.

The role of general concepts is to provide a framework for thinking and to aid in directing compromises which must be made to attain a useful, practical result. The directions of the compromise which must be taken in the retreat from generality will, of course, be guided by the nature of the problem under study. The nature of the CHOOSE problem has been described in preceding sections. These give some suggestion as to what the retreats must be. The next two major sections develop those concepts in greater detail.

## Value Judgments

Having agreed that the ordering relation "is as good as or better than" is to be the guiding theme for comparison, it will be useful to investigate how this can be applied.

In many cases the concepts of good, better and best are really embedded in the text of the description. For example, consider the binary question related to a particular streamflow model, "Does the model predict peak flow?" If we are viewing this in a completely decomposed framework in which each response is evaluated on its own merit without considering the implications on other questions, then an answer of "yes" is almost without question better than an answer of "no".

As a second example consider the following. A set of streamflow models is to be evaluated on the basis of how accurately it predicts the time of a flood peak for a particular type of storm event. The selected finite set of evaluations could be as follows:

1. Does not predict time of event
2. Predicts within +6 hours
3. Predicts within $\mp 1$ hour
4. Predicts within $\mp 10$ minutes
5. Predicts within $\mp 3$ minutes

While it may not be important to an individual model user that timing of flood peak be predicted better than one hour, it is still intuitively obvious that the higher numbered evaluations are in the ordinary sense better than the lower ones. Again this assumes the situation is not confounded by concern for the influence on other questions.

But according to Hamlet (Shakespeare, 1940), "things are neither good nor bad, but thinking makes them so." There are situations where a CHOOSER or an EVALUATOR must take a position on what is good or bad and perform a ranking (total ordering) accordingly. Suppose, for example, one is faced with a question of evaluating streamflow models with respect to the size of basin for which they are suitable. An integer indexed set of symbolic evaluations might be as follows:

1. Useful for urban water catchments up to 100 acres.
2. Useful for suburban catchments from 10-1000 acres.
3. Of general use for undeveloped watersheds from 1000 acres to 100 square miles.
4. Useful for modelling perennial stream basins up to 50,000 square miles.
5. Useful for modelling ephemeral stream flow in basins from 5 to 500 square miles.

Here the concept of what is good or bad is not intujtive at all and would vary substantially depending on the user's viewpoint. Here, to paraphrase an old saying, "best is in the eye of the beholder." Symbolic sets of the above type which cannot be totaliy ordered on the basis of binary relationship "is as good as or better than" are useless in the present context. It is up to the ingenuity of the EVALUATORs to formulate evaluations which will not encounter the logical problem of the above example. If this cannot be done then the CHOOSE procedures will not be directly applicable.

To modify the above example consider now what might be done to the above to make it useful in the CHOOSE context. A set of questions of the following nature can be posed, each of which admits of a set of symbolic responses capable of the required total ordering.

1. Is the model suitable for use in evaluating urban watersheds?
2. Is the model of use in evaluating suburban watersheds?
3. Is the model of use in evaluating undeveloped watersheds?
4. The model is capable of handling watersheds of the following minimum size:
(1) 100 square miles
(2) 10 square miles
(3) 1 square mile
(4) 100 acres
(5) 10 acres
(6) 1 acre
5. The model is capable of handling watersheds of the following maximum size:
(1) 1 acre
(2) 10 acres
(3) 100 acres
(4) 1 square mile
(5) 10 square miles
(6) 100 square miles

Note that questions 1, 2 and 3 are binary questions that are worded so they can be answered yes or no, although it is true that questions 2 or 3 could perhaps be evaluated on the basis of the degree of their ability to do the required tasks and total ordering could be developed on this. For question 4, response (6) can, on the basis of unaided intuition, be considered better than (1). This is simply because the word "minimum" appears in the statement. The corresponding question (question 5) involving "maximum" is required to clarify the situation completely. Note that this question inverts the order of the six responses so that the larger integers representing responses can be considered better than the lower ones.

It is true that a total ordering at the element level would not be required and that partial orderings could be of some value. However, since a fairly large number of partial orderings is not likely to produce a very useful product ordering, it seems best at this point to constrain the direct responses for the CHOOSE model to those that can be totally ordered.

It can be seen that even though the numbers associated with the various symbolic responses are not in fact measures of the various properties, it may be possible to use them as measures in certain limited contexts. This will be particularly true if the step size between the various choices is organized in such a way that to the CHOOSER or EVALUATOR each integer increase intuitively represents about an equal "step" of desirability.

## Decoupling or Decomposition

Although the importance of decoupling or decomposition of questions (attributes) has been touched on in the preceding section, some further development with examples seems indicated.

Note that it was necessary to break the one question in the previous section down into five independent questions in order for the concept of best to be embedded suitably. This unfortunately will often be the case. However, some limitation on the total number of questions will be necessary if a CHOOSER can be expected to interact with the CHOOSE system and get meaningful answers within the space of a few hours. Although experience with the television game, "Twenty Questions," would indicate that some fairly complex situations could be resolved with 20--usually binary--questions, it seems plausible to embed a higher number in the

CHOOSE mode1. For the present configuration of the CHOOSE system, on the CDC 6400 computer the limitation is 59 binary questions and 14 questions capable of up to 8 responses.

As an example of the need for decomposition of questions into those whose replies can be independent of other replies, consider the following. In choosing material for an electrical conductor, the cost of the material in cents per pound could be considered an independent and important question which is capable of being totally ordered, as described in the preceding section. However, that is not the entire story. The conductivity and density of the material are going to be involved in determining how much is required. Conductivities could also be stated as an independent question and the conductivities of various materials again is amenable to total ordering, where the higher conductivities are considered better. A total ordering on material densities would also be possible but here it would be difficult to apply a value judgment. Is higher density better than low? But here the problem arises that the CHOOSER cannot really reach a decision on the basis of independent answers to any of these questions. In a typical CHOOSE situation where the CHOOSER constitutes an intelligent but uninformed user, he will of course seek the lowest cost and the highest conductivity and probably not find any material that meets both of these requirements. In these situations the EVALUATORs should perhaps suppress both of the preceding questions and instead ask a single question again amenable to total ordering as to what is the cost per unit length of a particular conductor per mho of conductivity. A single response to this question--which should be within the capability of an intelligent CHOOSER--would lead more directly to the required material.

Additional questions of course could be used to clarify further the material choices for particular uses. Attributes such as structural properties, corrosion resistance, size limitations, etc., could be embedded in a CHOOSE model for such a situation.

It should be very clear now that the extensions sought to current choice procedures place a substantial burden on the EVALUATORs to generate not only independent questions/attributes, but to generate them strictly in the contxt of the CHOOSER's limited ability to make use of them. In other words the mapping from attributes to objectives is done by the EVALUATORs. The CHOOSERs conduct their deliberations strictly in an atmosphere of objectives.

This is not to imply that a question regarding cost per pound of the material would be completely useless. It may be important to certain CHOOSERs as a secondary objective. What is important in the objective questions for CHOOSE is that the right questions need to be present for the application at hand. It is possible to construct the CHOOSE model so that questions that are simply ignored or unanswered by the CHOOSER will not influence the ensuing analysis. The most important concept is that the CHOOSER must decouple the question in his mind in preparing his responses rather than attempting to equivocate between the questions. [It would be, of course, possible to build such continuous trade-offs between two questions into the model, but at this stage of
development it seems better to stop at this point in developing the attribute/objective space. Trade-offs in a continuum of the nature of those developed by Boyd (1970) and Monarchi (1972) could be embedded at a later date.]

## The Resulting Feasible Region

If each attribute or objective had been associated with a subset of the positive integers without embedded value judgment, the resulting space would have been simply the series of rather randomly placed points in the space $\left(I^{+}\right)^{n}$, where $n$ is the number of attributes or objectives into which the candidate was mapped. For a two-dimensional visualization of this, consider the case of four candidates with the following attributes:

| Candidate |  | Attribute 1 |  |
| :---: | :---: | :---: | :---: |
|  |  |  | Attribute 2 |
| 1 | 2 | 6 |  |
| 2 |  | 6 | 1 |
| 3 |  | 2 | 6 |
| 4 |  |  | 8 |

This is mapped in two-dimensional positive integer space in Figure 2. Here, since no ordering is implied by the numbers associated with the various characteristics of the candidate, one has no way of establishing any criteria for exploring the attribute space in any rational method. The only way to operate in this situation is to view each candidate in its entirety and make some overall value judgment.

However, using the value judgment concept for the various attributes/objectives, the feasible region can be enhanced substantially since each candidate which is evaluated contains in its characteristics (attribute/objective set) all of the points with lower numbers. For example, in Figure 3, candidate \#1, if rated on the same basis as before, now extends the feasible region to include all points lying. in the Cartesian product space $\{1,2\} \times\{1,2,3,4,5,6\}$.

The same is true for each of the other candidates and the feasible space generated by the four candidates is exhibited by all of the x's shown in Figure 3. With this type of space the CHOOSER (on the basis of the iterative procedures described in detail in Chapter 4) is given more of an opportunity to maneuver in the space until he finds the best trade-off between his input and the capabilities of the various candidates.

Briefly then, one considers those choice situations in which the evaluation of the various candidates is to be done on the basis of an n-vector of the decomposed or independent objectives, each element of which can be assigned a numeric value from a small subset of the positive integers such that the larger numbers represent the better ability to meet the associated objective. Each of the candidates is then


Fig. 2. Feasible Space without Value Judgments


Fig. 3. Feasible Space with Value Judgments
evaluated in terms of a vector from the set of all vectors determined by the maximum possible candidate space.

This leads to a vector maximization problem of the type on which much work has been done. See Cochrane and Zeleny (1973) where some of the work has been done along this line.

The selection of the best available vectors turns out to be a problem of defining aspiration level of individual CHOOSERs in such a way that each individual CHOOSER can choose what is best in the context of his particular application. The dynamics of this choice process are described in Chapter 4. The embedding of the above ideas into a mathematical description follows after notes on the probability of feasibility and on dominance.

## The Existence of Feasible Solutions

It appears that there are two fundamentally-different choice situations which arise in practice. The first of these is one in which all the alternatives are known to the individual who needs to make the choice. In this case he can analyze the trade-off questions in the context of his view of the problem at hand and of the contributions which each of the available alternatives would make to the situation. He then chooses, on the basis of some stated or unstated overall satisfactum, the one which he determines best in the context of the time available, importance of the choice, and his state of knowledge of the pros and cons of the situation.

A simple example of this situation would be the foreman of a construction company who is required to make up a special crew with motor vehicle to undertake a bridge repair at some distant point on short notice. He is presumably aware of the capabilities and limitations of each of the men available and of the motor vehicles at his disposal. While he is not likely to make the "optimal" choice of crew and vehicle in everyone's judgment, he will probably make a reasonably satisfactory one. The point here, however, is that, whatever his choice; it is not likely to be infeasible. He will not attempt to assign personnel not available or vehicles which are not available to him.

The second situation is the one of concern here. This is the case in which the CHOOSER does not have an intimate knowledge of capabilities and limitations of what could be made available. This situation arises when the CHOOSER is not the "expert" on the subject matter of the problem but only has vague ideas of what is possible (feasible) in the field of his inquiry. This situation seems to be at the heart of the model (structure) choice questions treated here. If presented with a set of judgment criteria and response sets, and no detailed knowledge of the candidate set, the CHOOSER will be propelled by his desire to do the best job to ask for the most benefit and offer the least of his resources in order to see if the choice is possible on such a basis. He may then be prepared to retreat to a point of less benefit and application of more resources in attempts to arrive at a suitable compromise.

In these circumstances infeasibility is likely to be a bigger hurdle than moving towards optimality from an initial feasible solution. In any case, the treatment of methods of moving towards optimality (or a satisfactum in the case of multi-objective situations) has been covered extensively in the literature, particularly in those situations in which the objective and constraints are such that the problem can be considered in terms of marginal substitution rates of real (and, hence, infinitely divisible) numbers. But in broader questions substantial uncertainty may be present on whether or not any feasible solution exists. While this may appear to be a pessimistic restatement of one of Murphy's Laws (if anything can go wrong, it will), it does correspond with some experience.

To illustrate this second situation with an example of personnel selection an exercise was conducted with the CHOOSE procedures of the Appendices. This was an attempt to assist in the selection of a new head for the Department of Systems and Industrial Engineering at The University of Arizona. The writer, who has had substantial experience in selecting professional and administrative personnel and in evaluating applications, worked informally to develop criteria for selection. He then evaluated all of the then 75 applications on the following basis of the 19 -vector element rating scheme established. Following this, three CHOOSERs (four, counting the writer) established the response vectors, stating their requirements for the position in the context of criteria developed. Although two of the CHOOSERs were members of the selection committee (and, accordingly, had prior knowledge of the qualification levels within reach), none of the candidates met the minimum requirements established for the position by any of the four CHOOSERs. By the rating scheme (described later beginning on page 56), which ranks candidates on a scale of 1 (meets all criteria without question) through 8 (fails to meet many of the tests, including failure by more than one step margin in at least one case), the best rating achieved on any of the evaluations was 5. A rating of 4 is the most marginal of the feasible ratings. A rating of 5 indicates that at least one of the firm minimum requirements was not met.

It is necessary, if the job is to be filled at all from this set of candidates, for the CHOOSER to modify his criteria. He may do this in several ways. He may relax some specification until he finds that some candidates become available. Not being satisfied with that, he may return the relaxed specification to somewhere near the original level and relax one or more of the other specifications until he finds a feasible non-empty set of candidates with which he feels he would be satisfied. He can then enter the optimization phase by varying his criteria, by introducing other criteria not previously incorporated, or by making subjective evaluations. Regardless of his technique for approaching optimality, the toughest task for him has been to establish feasibility. The optimality of the final choice will probably remain elusive even though certain mathematical niceties may be observed in making the final choice.

The question may arise, "Is there any essential difference between the technique just described for establishing the existence of at least
one feasible solution, and of moving toward optimality from a collection of feasible solutions?" In one sense, in the example just given, moving towards feasibility has the aspect of not moving at all but simply retreating from the original goals and constraints to a less desirable set. It constitutes a retreat from original goals and constraints to a new set, attempting to keep the retreat as orderly and as small as possible. Of course, if such a retreat is in itself feasible, it must mean that the original position must be overstated to some degree. If the entire relaxation of goals and constraints ultimately achieved had been available at the outset, and if, in this form, they uncovered a non-empty set of feasible solutions, then the problem would have appeared at the outset as one of traditional moving toward optimality in a feasible set to one final choice (following Roy, 1971). So there may be no essential conceptual difference between moving toward feasibility and moving toward optimality, but simply a question of direction of movement of the constraints, and adjustment of goals. Yet for the class of problems considered here, where the CHOOSER is seeking to do his best without detailed knowledge of what the possible choices are, infeasibility can be expected to be the typical first result. The alternative would be for the CHOOSER to enter the system with absolutely minimal acceptable criteria for each component so that all tentatively feasible solutions would be visible at the outset. This would reverse the problem to the traditional one of moving toward optimality. However, in terms of the human proclivity for trying to get the most for the least, by negotiating from an initially overstated initial position, it may be that moving toward feasibility may be not only the typical situation but also the preferred one.

While it is impossible to state the degree to which the feasibility question would arise in any particular model choice context, it is instructive to speculate on the probability of finding no feasible solutions in the sense that there is no candidate which meets all of the stated requirements at some minimal level. Certain simplifying assumptions are needed. Suppose that we must satisfy, at the minimum level, n criteria; that is, we are evaluating an n-tuple figure of merit to see that all n conditions are met at the minimal level. Suppose further that there are $m$ candidates and that $p$ is the probability that any element of any candidate will be "equal to or better than" the corresponding requirement. Then the probability of failure of a particular candidate on any one point (any one element of the n-tuple) is $1-p$. The probability that the candidate passes each of the $n$ tests is $p^{n}$, and the probability that it fails at one or more points is $1-p^{n}$. Hence, if $p=0.75$ and $n=20$, the probability of an individual candidate's failing is 0.997 or $99.7 \%$.

Now, if there are m candidates, each with the same likelihood of failure, the probability that they will all fail is ( $\left.1-p^{n}\right)^{m}$. Hence, the probability that at least 1 will pass is $1-\left(1-p^{n}\right)^{m}$. Continuing the above example, if there are 3 candidates, the probability of existence of a feasible solution is $0.9 \%$; if the candidate set increases to 100, the probability of finding a feasible solution increases to $27.2 \%$.

The purpose of the above estimates is to develop some insight into the magnitude of the problem. No implication is intended that probabilities play any role in the actual selection processes contemplated by the CHOOSE procedures outlined here. Some extension of the sample computations are contained in Table 1.

Table 1
Percentage Chance of Finding Feasible Solution


Of course it may turn out that, even under the most favorable statements of goals and constraints, there may be no feasible choice in the candidate set. The CHOOSER's problem remains. Unless the candidate set is incomplete, has been prepared improperly or is not up to date, the CHOOSER can assume that the problem has no present solution in the context of his requirements. Additional research, development, synthesis, discovery and/or problem reformulation may be required before the problem is disposed of. And, finally, as noted by al-Ghazali at the beginning of the chapter, there may be no solution.

## A Note on Dominance

In keeping with the concepts of dominance and dominated strategies as they appear in decision theory and game theory (Sheridan and Ferre11,

1974, p. 409), the notion of candidate dominance could be introduced here. Since (as previously established) each candidate or model will be evaluated in terms of finite n-tuple, where each element of the ntuple for a particular candidate is selected from a finite set of responses totally ordered on the basis of "is as good as or better than," a working concept for dominance here would be as follows: a candidate is dominated if some other candidate is at every element as good as the candidate in question, and is for at least one element better than the candidate in question. Such candidates would ordinarily never be selected, and, presumably, all dominated candidates could be discarded.

Such an approach would not appear useful here, however. The reason is quite simple. Certain questions (i.e., elements of the n-tuple) will not be pertinent in some applications. It would be improper to delete from the prospective candidate list those candidates which are dominated because of their better position with respect to those unused questions, when, in fact, they may be just as good with respect to the requirements of a specific application.

Also, it may happen that the CHOOSER, having once selected a dominant model, finds, on further study and evaluation, that the rating supplied by the experts on some point does not coincide with his viewpoint. The candidates or models previously found to be dominated may now become viable choices. On re-entering the CHOOSE process the user should then be able to extract from the system these "second" choices.

Further, the evaluation of a candidate with respect to a particular set of questions and answers is not the entire story. There may be several properties not included in the evaluation scheme which, on more detailed study of a selected subset of candidates, would be of importance. It does not seem appropriate, then, that any candidate should be considered dominated on the basis of the limited evaluation criteria which would be set up in this scheme.

So, instead of determining dominance a priori, it seems preferable to let dominance be determined by the relationship of the interactive user's requirements to the candidates, rather than by the more customary way. In fact, the entire interactive choice scheme might, in some sense, be considered "dynamic dominance determination." This position recognizes that the experts are not omniscient and that the finite representations in this evaluation structure cannot always be the basis for final choice.

## A Prototype CHOOSE Mode1

The ideas presented in the preceding section can be embedded easily in a choice model 4-tuple. This will be done as an intermediate step to the final CHOOSE model. The final one contains additional features in the computation of the ordering which reflects the need for a greater degree of "ordering power" in order to narrow the choice more quickly-that is, with fewer iterations of the model.

The elements of the 4 -tuple of the prototype CHOOSE model, $P$, are defined as they occur in the following paragraphs.

The first step is to describe the general nature of the CHOOSE candidates; the specific candidate set will, of course, be associated with a particular choice problem. For the type of problem under consideration there will be a finite number of candidates. It will be useful to have descriptions of the candidates in three different ways to serve different audiences. Fist, each candidate is assigned a serial number i (from 1 through m ) for cataloging, easy reference, and unambiguous identification. Second, in order that each candidate be identifiable to human CHOOSERs, provision is made for a short distinctive mnemonic name. This is conveniently described by a single vector of individual descriptors $w=$ $\left(w_{1}, w_{2}, \ldots, w_{i}, \ldots, w_{m}\right)$. Third and last, each candidate will have associate with it (not necessarily uniquely) its own $v_{i} \varepsilon V$, where $V$ is the attribute space described in the following way. Let
$\mathrm{n} \varepsilon \mathrm{I}^{++}$, a number representing the number of attributes considered for each of the candidates;
$j=I[1, n]$, an index representing the $j^{\text {th }}$ attribute;
m $\quad \mathrm{I}[1, \infty$, a number representing the number of candidates;
$i=I[1, m]$, an index representing the $i^{\text {th }}$ candidate;
$\mathrm{k}=\mathrm{I}[2, \infty]$, a number representing the maximum number of levels of evaluation for any attirubte in the mode 1 ; and
$k_{j}=I[2, k]$, a number representing the number of levels of evaluation for attribute $j$.

Then $V=I\left[1,1_{1}\right] \times I\left[1, k_{2}\right] X \ldots X I\left[1, k_{j}\right] X \ldots X I\left[1, k_{n}\right]$.
This permits the following definition of CANDIDATES(P):

$$
C=\left\{\left(i, w_{i}, v_{i}\right): i \in I[1, m], w_{i}=\pi_{i}(w), v_{i} \varepsilon V\right\} .
$$

The function $g$ is easily defined. It requires an input vector $\mathrm{b} \varepsilon \mathrm{V}$ to act as a constraint to eliminate undesirable candidates. Then

$$
\mathrm{g}=\{(\mathrm{x}, \mathrm{ok}\}): \mathrm{x} \in \mathrm{C}, \mathrm{ok} ?=\mathrm{ACCEPTABLE} \text { if } \mathrm{b} \leq \pi_{3}(\mathrm{x}),
$$

$$
\left.o k ?=N O T A C C E P T A B L E \text { if } \mathrm{b} \text { 上 } \pi_{3}(\mathrm{x})\right\} .
$$

A simple measure of optimality--although as ill be noted later, not a particularly useful one--is the sum of the gits by which the various attributes exceed the minimum requirements established by the
constraint b . This means that the GOALMEASURES( $P$ ) constitute a finite subset of the nonnegative integers. The goalfunction $(P)$ is defined as follows:

$$
f=\left\{(x, q): x \in C^{\prime} ; q=\sum_{j=1}^{n} \pi_{j}\left(\pi_{3}(x)-b\right)\right\}
$$

The ordering, $\alpha$, to be used is the simple ordering of the subset of the nonnegative integers defined by the "less than or equal to" concept . Formally

$$
\begin{aligned}
\alpha= & \{(x, y), z): s, y \varepsilon C^{\prime} ; \text { if } f(x) \leq f(y), z=\text { TRUE, } \\
& \text { if } f(x) \notin f(y), z=F A L S E\} .
\end{aligned}
$$

The prototype model uses as its input the vector $\mathrm{b} \varepsilon \mathrm{V}$, and its output constitutes a simple ordering of the feasible or admissible alternatives of the candidates present in the model $P$. Note that in this model if the set of feasible candidates, $C^{\prime}$, is not empty, there will be at least one optimal candidate.

Although the prototype model does permit the rejection of some candidates as infeasible and perhaps others as being non-optimal, it does not constitute a complete solution to the present problem. It simply presents its ordering of the candidates based on the sum of the "excess" performance capability over all of the attributes. It evaluates each candidate as though excess performance in any attribute is of equal importance to the CHOOSER. Further, it does not give any idea of "feasible directions" for the user to move should he desire additional iterations of the model to seek feasible candidates (if there are none) or to find the better candidates if the choice is not clear. The next section modifies the prototype CHOOSE model to mitigate these difficulties.

## Additional CHOOSE Considerations

Assuming the correctness of the premises on "probability of feasibility" and the needs for iteration to find the most satisfactory positions for the individual CHOOSERs, the following modifications and extensions to the prototype CHOOSE model have proved to be useful.

1. Rather than having the input consist of a single vector $b \varepsilon V$, the input consists of two vectors: $b$ and bhigh $\varepsilon \mathrm{V}$ such that b < bhigh. This permits the CHOOSER to clarify those things that are important to him. In the b vector he establishes those values for the individual criteria which he considers to be the minimum acceptable. In the bhigh vector he solicits those values he considers to be particularly important to his higher aspiration levels. This permits the model to give emphasis to criteria of particular interest to the CHOOSER.

Rather than constituting an additional burden on the CHOOSER, this actually results in a reduction in effort. The CHOOSER need now pay less attention to making precise choices; he now simply quantifies his uncertainty more quickly by specifying the range of his aspirations. He is under much less pressure to be extremely precise in his deliberations, and hence can proceed more rapidly with preparation of the entry information.
2. A similar problem exists with the candidate vectors $v_{i}$. The EVALUATORs will sometimes be able to evaluate the candidates unequivocally at one level, and sometimes will prefer to specify an extended range of competence that is reachable under favorable conditions. But this was considered to be useful only for those criteria which had more than two levels of response. It was felt that those having only two levels of response (that is, $k_{j}=2$ ) could be evaluated unequivocally by the EVALUATORs since they represent binary choices (Yes, No; True, False; the candidate has this property, the candidate does not have this property). Hence, the candidate vector is expanded to a 4-tuple $\left(i, w_{i}, v_{i}\right.$, vhigh $_{i}$ ) where $v_{i}, \operatorname{vhigh}_{i} \varepsilon V, v_{i} \leq \operatorname{vhigh}_{i}$, and for all $j \varepsilon I[1, n]$ if $k_{j}=2$, then $\pi_{j}\left(v_{i}\right)=\pi_{j}\left(\operatorname{vhigh}_{i}\right)$.
3. Since infeasibility was demonstrated in earlier tests of the procedure to be an important problem in the choice process, some method of finding candidates "almost feasible" was considered important. Thj.s leads to the concept of "degree of infeasibility" which, when combined with the various comparisons available between $b$ and bhigh, and $v_{i}$ and vhigh $_{i}$, leads further to a primary ranking based on level of feasibility and optimality, and a secondary ranking based on measures of excess capacity or degree of deficiency.

The elements of the final CHOOSE 4-tuple ( $H$ ) can now be discussed.
The candidate set, $\operatorname{CANDIDATES}(H)=\mathrm{C}$ is defined as follows:

$$
\begin{aligned}
& C=\left\{\left(i, w_{i}, v_{i}, v h i g h_{i}\right): i \varepsilon I[1, m] ; w_{i}=\pi_{i}(w) ; v_{i}, \operatorname{vhigh}_{i} \varepsilon V ;\right. \\
& \\
& \quad v_{i} \leq \operatorname{vhigh}_{i} ; \text { and for all } j \varepsilon I[1, n] \text { if } k_{j}=2, \text { then } \\
& \\
& \left.\pi_{j}\left(v_{i}\right)=\pi_{j}\left(\operatorname{vhigh}_{i}\right)\right\} .
\end{aligned}
$$

The input information required to establish feasibility and optimality consists of two vectors: b , and bhigh $\varepsilon \mathrm{V}$, where $\mathrm{b} \leq$ bhigh.

$$
F=\{(p, s): p \varepsilon I[1 ; 8], s \varepsilon I\} .
$$

The component $p$ constitutes the primary rating level and the component $s$ constitutes the secondary rating information.

Before going into the formal definitions of $g, f$ and $\alpha$, the meaning and construction of the rating structure will be presented under the 8 primary rating numbers. Note that this information is tabulated in Figure 16 on page of Appendix A in a user-oriented form. Those who prefer the non-mathematical description are referred to that discussion.

1. Outstanding. (bhigh $\left.\leq v_{i}\right)$. Here the highest aspiration level is met or exceeded by the standard (low) candidate rating for each criterion. A secondary rating scheme is based on the excess capacity
measured by $\sum_{j=1} \pi_{j}\left(v_{i}-\right.$ bhigh $)$. The larger the secondary ranking, presumably the better the choice, although in some sense all those candidates with a primary rating $p=1$ are equal. That is to say that the excess capacity or value for a particular criterion may be of no significant value to a particular CHOOSER, and he may prefer to use subjective criteria in choosing from among those candidates for which $\mathrm{p}=1$.
2. Fully Acceptable. $\left(\mathrm{b} \leq v_{i}\right.$ and bhigh $\leq$ vhigh $\left._{\mathrm{i}}\right)$. Candidates with a primary rating of 2 are considered fully acceptable. The secondary rating value for such candidates is $s=0$.
3. Acceptable A. (bhigh $\leq$ vhigh $_{i}$ ). Candidates with a primary rating of 3 are reasonably safe in that optimistic or extended candidate performance (vhigh ${ }_{i}$ ) does meet or exceed the highest aspiration level (bhigh)--and, therefore, also exceeds the entry or minimum level (b). But there are some points that prevent $\mathrm{b} \leq \mathrm{v}_{\mathrm{i}}$. That is, the candidate, working at its standard or basic level of evaluated performance, misses somewhat the minimum acceptable level of performance. The secondary rating for such candidates is $s=0$.
4. Acceptable $B . \quad\left(\mathrm{b} \leq \mathrm{v}_{\mathrm{i}}\right)$. The candidate, i , clearly meets the minimum established requirements even though the higher aspirations, bhigh, are not met by the candidate even when its most favorable evaluation, vhigh ${ }_{i}$, is used. This is poorest of the ratings in which the CHOOSER can have reasonable confidence. The secondary rating value for candidates with a primary rating of 4 is $s=0$. Hence, the element of F associated with wuch a candidate is $(4,0)$.
5. Marginal. (b $\left.\leq \operatorname{vhigh}_{i}\right)$. The most optimistic candidate rating. vhigh $_{i}$, exceeds the minimum requirements, b. This means that full approval is not given but that there is some expectation that the choice
would be satisfactory. Here again the secondary rating of such candidates is zero, and the associated element of $F$ is $(5,0)$.

In going to poorer ratings ( 6,7 and 8 ), an additional distinction is useful. This is the distinction between binary questions for which clear-cut evaluations should be possible, and higher order questions for which some equivocation in evaluation can be permitted. Any candidate which cannot meet all of the mìnimum binary aspirations does not seem worthy of further consideration. It may be thought of as having at least one fatal defect. In order to attack this particular problem it is useful to locate the binary criteria in vector V . Let $\mathrm{k}=$ $\left\{j: k_{j}=2\right\}$, then $k$ is a set of indices that locate the binary attributes in $V$. The $k$-projection of any element of $V$ sill present the binary criteria. To insure that binary criteria are met it will be required that $\pi_{k}(b) \leq \pi_{k}\left(v_{i}\right)$. The complement of the set $k$ in the set $I[1, n]$ is denote $\bar{d} \bar{k}$ and $\bar{d}$ efined $\bar{k}=I[1, n] \sim \mathrm{k}$. With this additional notation, the description of the ranking levels can continue.
6. Bad. $\quad\left(\pi_{k}(b) \leq \pi_{k}\left(v_{i}\right)\right.$, and $\pi_{\bar{k}}\left(b-\{1\}^{n}\right) \leq \pi_{\bar{k}}\left(\right.$ vhigh $\left.\left._{i}\right)\right)$. A rating of 6 indicates that the candidate meets the binary criteria but that the CHOOSER must reduce his minimum requirements on one or more higher order criteria by one level in order to enable this candidate to attain a ranking of 5 . The secondary ranking value is the negative of the count of the number of criteria requiring such adjustment: $-\sum_{j \varepsilon \bar{k}}^{\sum_{j}} \pi_{j}\left(b-v h i g h_{i}\right)$.
7. Very Bad. $\quad\left(\pi_{k}(b) \leq \pi_{k}\left(v_{i}\right)\right.$, and $\left.\pi_{k}\left(b-\{1\}^{n}\right) \leq \pi_{\bar{k}}\left(\operatorname{vhigh}_{i}\right)\right)$. A rating of 7 indicates that the candidate is almost hopelessly out of reach of the CHOOSER's requirements since it fails to meet some higher order criteria by more than one level of response, even though all of the minimum binary criteria are met. A substantial adjustment of the CHOOSER's aspiration levels would be required to make this candidate acceptable. For such candidates the secondary rating criteria is $\mathrm{s}=0$.
8. No Hope . $\left(\pi_{k}(b) \leq \pi_{k}\left(v_{i}\right)\right)$. Candidate with a rating of 8 must be considered "No Hope" unles $\bar{s}$ the CHOOSER's thinking is completely revised. In order for such a candidate to come up for consideration, the CHOOSER must reverse his position on all of the binary aspirations that the candidate cannot meet. For such candidates the secondary rating criteria is $\mathrm{s}=0$.

The eight primary rating levels and their relationships to the vectors $b$, bhigh, $v_{i}$ and vhigh ${ }_{i}$ are shown in Figure 4 on the following page.

The remaining elements of the CHOOSE 4 -tuple can now be defined. If the concepts of the prototype CHOOSE model were continued here, all candidates with a primary rating of 6 or larger would be considered not acceptable. Because of the feasibility problems previously described, a strong need may exist for downgrading aspirations in order to find any

## ASPIRATION <br> LEVEL

(Decreasing downward)

EVALUATION
LEVEL
(Increasing downward)

(7) Very Bad. Test 6 is met for higher order criteria, but fails on one or more binary criteria.
(8) No Hope. Some binary criteria are not met.

Fig. 4. A Diagram of the Relationships of the Primary Ratings
feasible candidate. For this reason the function $g$ will not be used in this model to reject any candidates.

Hence

$$
\mathbf{g}=\{(\mathbf{x}, A C C E P T A B L E): \mathbf{x} \varepsilon C\} .
$$

The goalfunction $(B)$ can now be partially descirbed as follows:
$\mathrm{f} \varepsilon$ FUNCTIONS(C,F).
The exact content of the set $f$ is determined by the aspiration levels $b$ and bhigh supplied by the CHOOSER and the relationships between the vectors $v_{i}$ and $v$ vhigh $_{i}$, which are, respectively, $\pi_{3}(x)$ and $\pi_{4}(x)$ where $x \in C$. The complete set theoretic description of $f$ is deferred to the next section where the full model description is summarized. The description of the set of GOALMEASURES $(H)$ above is the basis for the definition of $f$.

The final element of the choice 4-tuple, choiceordering( $H$ ), reflects a lexicographical ordering of the $\operatorname{RANGE}(\mathrm{f})$ using pair-wise comparisons between elements of $F$. It is described as follows: $\alpha \varepsilon \operatorname{ORDERINGS}(\operatorname{RANGE}(\mathrm{f}))$, such that for all $\mathrm{s}, \mathrm{y} \varepsilon \mathrm{C}^{\prime} \alpha(\mathrm{x}, \mathrm{y})=\operatorname{TRUE}$ if $\pi_{1}(f(y))<\pi_{1}(f(x))$, or if $\pi_{1}(f(y))=\pi_{1}(f(x))$ and $\pi_{2}(f(y)) \geq \pi_{2}(f(x))$; $\alpha(\mathrm{x}, \mathrm{y})=$ FALSE otherwise.

## Relationship to Fuzzy Sets and Approximation Spaces

Note that the development of the 8 valued rating scheme for the various candidates can be considered from alternate viewpoints. Zadeh (1965, 1971) developed the concept of Fuzzy Sets, as opposed to "crisp" sets, which are the type customarily encountered. Crisp sets are those where membership or non-membership in a set is completely deterministic; an element either belongs in the set or it does not. Fuzzy sets, however, consist of a set of ordered pairs where the first element of the ordered pair is the element that would normally appear or not appear in the crisp set in a deterministic basis. The second element indicates the "degree of membership" of the associated element in the corresponding set. The elements for the degree of membership can be any set but, more tractably, ones on which partial ordering or total ordering can be defined. In many cases the set of real numbers [0,1] is used where 0 indicates nonmembership in the set and 1 indicates full membership. In this case, however, we consider the set of integers [1.8]. These constitute the degree of membership criteria where the number 1 indicates full membership in the set of feasible candidates and the number 8 indicates nonmembership.

Hammer (1969) followed a similar idea in his development of generalized "approximation spaces." In somewhat the same way he associated each element with an element from the second set in such a way that the
element from the second set in some way measured how good an approximation the first element was to the element being approximated.

If we consider the input vectors as establishing the CHOOSER's aspirations, then we can consider the 8 valued rating for comparative candidates as measures of their degree of approximation to the requirements. In this case the value 1 represents the best approximation and the value 8 the poorest.

It seems possible that the notation offered by either of these techniques would be of some value in attempting to extend the CHOOSE concepts. However, nothing further will be done here with either, since the present notation seems adequate for the current level of development.

The CHOOSE Model
The set theoretic description for the CHOOSE model can now be assembled. For convenience, some of the definitions are restated here.

Given the following:
$\underline{m}=\mathrm{I}[1, \mathrm{~m}]$, where $\mathrm{m} \varepsilon \mathrm{I}^{+}$;
$\underline{n}=I[1, n]$, where $n \varepsilon I^{+}$;
$\mathrm{k} \in \mathrm{I}[2, \infty) ;$
$\mathrm{k}_{\mathrm{j}} \in \mathrm{I}[2, \mathrm{k}]$ for all $\mathrm{j} \varepsilon \underline{\mathrm{n}}$;
$\underline{\mathrm{k}}=\left\{\mathrm{j}: \mathrm{j} \varepsilon \underline{\mathrm{n}}, \mathrm{k}_{\mathrm{j}}=2\right\} ;$
$\overline{\mathrm{k}}=\underline{\mathrm{n}} \sim \underline{\mathrm{k}} ;$
$\underline{1} \varepsilon\{0,1\}^{\mathrm{n}}$ such that $\pi_{j}(\underline{1})=0$ if $\mathrm{j} \varepsilon \underline{\mathrm{k}}$, and $\pi_{\mathrm{j}}(\underline{1})=1$ if $\mathrm{j} \varepsilon \overline{\mathrm{k}}$;
$V=\sum_{j=1}^{n} I\left[1, k_{j}\right] ;$
$w_{i}=a$ unique descriptor (for all $a, b \varepsilon \underline{m}, w_{a} \neq w_{b}$ unless $a=b$ ) for each i $\varepsilon \underline{m}\}$;
w $\sum_{i=1}^{m} w_{i}$;
b $\varepsilon$ V;
bhigh $\varepsilon \mathrm{V}$ such that $\mathrm{b} \leq$ bhigh.

The CHOOSE Model, then is a 4 -tuple $H=(C, g, f, \alpha)$ where

$$
\begin{aligned}
& C=\left\{\left(i, w_{i}, v_{i}, v h i g h_{i}\right) ; i \varepsilon \underline{m} ; w_{i}=\pi_{i}(w) ;\right. \\
& v_{i} \varepsilon \mathrm{~V} ; \operatorname{vhigh}_{\mathrm{i}} \varepsilon \mathrm{~V}, \mathrm{v}_{\mathrm{i}} \leq \operatorname{vhi} \mathrm{gh}_{\mathrm{i}} \text {; for all } \\
& \left.j \varepsilon \underline{k}, \pi_{j}\left(v_{i}\right)=\pi_{j}\left(v h i g h_{i}\right)\right\}, \\
& \mathrm{g}=\{(\mathrm{x}, \text { ACCEPTABLE }): \mathrm{x} \in \mathrm{C}\} \\
& f=\left\{(x,(p, s)): x \in C^{\prime} ;\right. \\
& \mathrm{p}=1 \text {, if bhigh } \leq \pi_{3}(x) \text {; } \\
& \mathrm{p}=2 \text {, if bhìgh } \mathrm{k}^{2} \pi_{3}(\mathrm{x}) \text {, bhigh } \leq \pi_{4}(\mathrm{x}), \mathrm{b} \leq \pi_{3}(\mathrm{x}) \text {; } \\
& p=3 \text {, if bhigh } \leq \pi_{4}(x), b \neq \pi_{3}(x) \text {; } \\
& p=4 \text {, if bhigh } \underset{\sim}{f} \pi_{4}(x), b \leq \pi_{3}(x) \text {; } \\
& \mathrm{p}=5 \text {, if bhigh } \leq \pi_{4}(\mathrm{x}), \mathrm{b} \neq \pi_{3}(\mathrm{x}), \mathrm{b} \leq \pi_{4}(\mathrm{x}) \text {; } \\
& \mathrm{p}=6 \text {, if } \mathrm{b} \text { 上 } \pi_{4}(\mathrm{x}), \mathrm{b}-\underline{1} \leq \pi_{4}(\mathrm{x}) \text {; } \\
& p=7 \text {, if b-1 } \underline{\xi}^{2} \pi_{4}(x), \pi_{j}(b) \leq \pi_{j}\left(\pi_{4}(x)\right) \text { for all } j \varepsilon \underline{k} \text {; } \\
& p=8 \text {, if } \pi_{j}(b) \nmid \pi_{j}\left(\pi_{4}(x)\right) \text { for all } j \varepsilon \underline{k} \text {; } \\
& s=0 \text {, if } p \varepsilon\{2,3,4,5,7,8\} ; \\
& s=\sum_{j \varepsilon \underline{n}} \pi_{j}\left(\pi_{3}(x)-\text { bhigh }\right), \text { if } p=1 ; \\
& s=\sum_{j \varepsilon \bar{k}}^{\Sigma} \pi_{j}\left(\pi_{4}(x)-b\right) \text {, if } p=6 \text {, and } \\
& \alpha=\left\{((x, y), a): s, y \varepsilon C^{\prime}, a \varepsilon\{T R U E, F A L S E\},((x, y), T R U E) \varepsilon \alpha\right. \\
& \text { if } \pi_{1}(f(y))<\pi_{1}(f(x)) \text {, or } \\
& \text { if } \pi_{1}(f(y))=\pi_{1}(f(x)) \text { and } \pi_{2}(f(y)) \geq 2^{(f(x)) \text {; } ; ~} \\
& ((x, y), F A L S E) \varepsilon \alpha \text { otherwise }\} .
\end{aligned}
$$

The parameters of this model are $\mathrm{m}, \mathrm{n}, \mathrm{k}, \underline{1}$, and w . These parameters are determined by the EVALUATORs. The inputs the model are the vectors $b$ and bhigh and are supplied by the CHOOSER to indicate his preference. As will be seen in Chapter 4, the CHOOSER is permitted to modify his preferences as he views the output. On this basis the CHOICE
model computes new outputs. The output consists of the partial ordering developed as a result of the EVALUATORs' parameters and the CHOOSER's inputs.

Of course the CHOOSE model stated above is simply the framework for defining the inputs, outputs and transformations required between the two. It is now necessary to design a system for implementing this model in a manner that the required inputs can be developed in appropriate form, the model computations performed, and the results displayed to the CHOOSER. The development of this system is the subject of the next chapter.

## CHAPTER 4

## IMPLEMENTING CHOOSE MODELS

It is, of course, only part of the problem to design the model. The CHOOSE model as described in Chapter 3 consists of, in its mathematical essence, a 4-tuple. Note that there is no time orientation implied in the computation. Similarly the exact information that is to be imbedded in the output is not clear and the inputs, while mentioned in the text of Chapter 3, are not formally defined. If a system is to be created for receiving inputs performing required computations in a suitable time frame and for displaying outputs, these additional aspects of the design of the computational system must be considered.

In addition to deciding on the computational requirements of a CHOICEMODEL $H$, some system is required by EVALUATORs to create the appropriate models. This again can be considered as a system design problem with its own input/output sets and its own time orientation. The discussion of the design of this auxiliary system begins on page 69.

In the context of a formal System Design Project (Wymore, 1973), it is necessary to establish both the technology in which the system is to be implemented and the input/output requirements of the system. Rather than giving the full set theoretic formulation of CHOICEMODEL implementation, it should be noted that the major part of the design was in the conceptual model and that its implementation only in computer based systems is considered.

## The Technology

The CHOICEMODEL H developed in Chapter 3 requires a certain amount of computation for any complex problem with a rather large amount of data to be considered. Requirements for the ultimate choice are such that the CHOOSER, if required to define his complete choice structure on the basis of all possible tradeoffs between the various objectives without any concept of where the ultimate choices will lie, would be faced with the essentially impossible task of defining preferences in every region possible of the objective space. But the structure of the questions and answers does imply that any candidate can achieve all of the goals even at some minimum level or that there exists any candidate in the set which meets the 'highest" goal for any one objective. As noted in Chapter 3, this leads to the consideration of the possibility of "progressive definition of goals" in the nature of an adaptive experiment. And, quite naturally, this leads to a division of responsibilities between the computer and the CHOOSER in such a way that computed results must be reviewed by the CHOOSER, who then analyzes the information and creates inputs for additional computation.

Because of the nature of the problem, two aspects of the problem must be considered: first, the proper division of the responsibilities
between the CHOOSER and the computer, and second, the time frame over which the interchange is to take place.

## Division of Man/Machine Responsibilities

Fitts (1962) analyzed the functions of man in complex systems and defined what he considered to be, at that time, the functions that men perform better than machines and what machines perform better than man. Although there have been substantial improvements in computers since 1962, improvements in human beings have been essentially negligible and the 1962 analysis seems still applicable in a qualitative way. A brief review of Fitts' criteria will guide the choice between machine responsibilities and CHOOSER responsibilities in the CHOOSE system.

1. Information-Handling Capacity.

Man has a very low information-handling rate. If he is asked to perform several different functions at once he most usually accomplishes them serially since he cannot perform two things requiring information processing at the same time. In contrast the computer's ability to process information is essentially high and is increasing rapidly over the years.
2. Information Storage and Retrieval Capacity.

While people differ widely in the amount of information they can store, the accuracy with which it is stored is limited. On the other hand, the computer is capable of storing large amounts of information and retrieving it with complete accuracy.

## 3. Versatility.

In spite of man's limitation for information-handling, he has an ability to handle a great variety of information processing tasks. He can switch rapidly from one form of processing to another as he perceives needs for the change. Computers can perform this work only when previously programmed to make the various changes.

## 4. Adaptation and Learning.

Man has an unusual ability for learning skills and for long term retention of important information and, particularly, for adaptation to the peculiar requirements of a new task or new environment. Computers are presently inadequate in this respect.

## 5. Speed-Accuracy Tradeoff.

Man has the unique ability to interchange speed for accuracy. Again computers can be programmed to do this but the decision rules for the change must be created initially in the program by some human operator. The computer cannot make the transitions on its own.

## 6. Reliability.

Man's performance varies as a function of alertness, fatigue, stress, interest, etc., whereas the computer's ability to perform can be kept at a high level at all times.

## 7. Judgment.

Judgment is another attribute possessed in varying degrees by humans but not at all by computers. Humans are able to "bring previous experience to bear on unique situations. They improvise, see relationships, select appropriate procedures, and usually arrive at decisions within the time and environmental limitations of the particular problem facing them at the moment. As a result, we often decide to include men in new systems because we assume that they will be able to react intelligently to situations which cannot be anticipated at the time the system is designed."
8. Dealing with Unreliable Information.
'Throughout his lifetime, man selects and smoothes information, estimates probabilities, anticipates alternative outcomes, and in various other ways learns to make decisions in the face of excess, missing, or unreliable information." The computer has this capacity only to the extent that it is first programmed by some human to do this.

## Time Schedule for Interaction

In the context of computer modelling Kemeny (1973, p. 382) puts the whole motivation for interactive time-sharing computer use very nicely.

Computer modeling is an outstanding example of a situation in which an occasional shot at the computer, or a 20 minute turn-around time, is totally useless. One of the great breakthroughs in time-sharing systems is the capacity for research scientists to converse with a computer. With a 20minute turn-around, however, you might as well have somebody else do your computing for you. You are never going to do the work for which computer models are ideal if you don't have the opportunity to sit at a terminal, vary the parameters of your data, watch the results come out, and, if the answers raise still more questions, to begin exploring the implications fully.
. . . But beyond the obvious advantage of a time-sharing system, the fact that you literally work with the system in the same way you work with a mathematical model with paper and pencil, makes a time-sharing system absolutely crucial for the development of computer models. (Emphasis supplied.)

If we substitute the words "computer-aided decision making" for "computer models" in the above, we have a pretty good picture of interactive motivation for the choice problems under consideration here.

Human Limitations in Computer Interaction
In organizing the system in which to embed the component system CHOOSE, the human limitations for such action must be considered. The following items seem appropriate:

1. If the interaction proceeds at a pace which will not be frustrating to the CHOOSER, the solicitations of information from the CHOOSER by the computer must be presented concisely and in the language and portable knowledge of the CHOOSER.
2. Computer responses should be reasonably prompt and unequivocal.
3. The CHOOSER should not be required to maintain extensive handwritten notes on the side as interaction takes place. The computer should perform this storage function and the information should be recallable by the CHOOSER as needed.


Fig. 5. The CHOOSER as both System User and System Component
4. As noted in the design guidelines at the end of Chapter 2, the system should be organized so that the CHOOSER could interact at his convenience, set the system aside with partially completed results if desired, and re-enter at his convenience to continue the work until he is satisfied with the results.

With this technology the choice was then made to use a computer based interactive system so that the CHOOSER himself becomes embedded in the overall system in a way that he can progressively define his preferences as he sees the intermediate results of his initial choices. This role is depicted in Figure 5. The computer plays the role of storing information in required form, performing the required calculations, and displaying the intermediate outputs. The process will terminate any time the CHOOSER is satisfied to let the last standing intermediate outputs become his final criteria.

With this division of responsibility between the CHOOSER and the system established, the input/output specifications for the "machine" portion of the CHOOSE system can be analyzed.

## Input/Output Specifications

Considering the non-human portion of the CHOOSE system, that is, the portion of the system in which the CHOOSER generates inputs and reacts to outputs, the CHOICEMODEL can be analyzed in terms of input/output requirements in the manner described below.

From the selection of an interactive mode as indicated in the technology section above, it is clear that the CHOOSER himself becomes an overall part of the system. It will, of course, not be possible to exercise much design influence on the CHOOSERs themselves. The CHOOSERs will be "designed" by their total educational experience prior to playing the role contemplated for them in the CHOOSE system. Therefore, the input/output criteria will focus on the actual CHOICEMODEL $H$ computations. Looking at the 4 -tuple item by item, we see the following:

C - the candidate set which is embedded in the model (CANDIDATES ( $H$ )) by EVALUATORs
g - the constraints ( $H$ ) on acceptability which, for purposes of this particular model, do not enter into the computation since candidates are not rejected but are only assigned numerical readings.
f - the goalfunctions $(H)$ which depend for their assignment of points in RANGE (f) on the elements of the candidate set and on the two vectors, b and bhigh, which are supplied by the CHOOSER.
$\alpha$ - the choiceordering $(H)$ which is dependent on the input. From this it is seen that the inputs to the model are the two vectors b and bhigh. It appeared desirable that these inputs be permitted to enter into the system in several different ways,
depending on the degree of preparation of the CHOOSER for selecting the values. Accordingly two methods were designed for entering the vector inputs an element at a time, and one method was built in to permit the inputs to be entered a complete vector at a time.

## Inputs

Inputs for this system will be those interactive messages that are required to establish the various aspirations of the CHOOSER with respect to the various components of the input vectors. It will be useful to build in error detection routines and appropriate display tables so that the information can be entered efficiently and accurately.

## Outputs

The primary output of the system is, of course, the sequential presentation of the candidates as totally ordered by $\alpha$ (choiceordering ( $H$ )). A number of other outputs were also determined to be needed. Prompting messages were required to elicit inputs and to point out inconsistencies in data supplied. Further, other outputs were needed which would permit the CHOOSER to generate specific additional inputs, establishing the type of output desired, and modifying inputs as the need arises.

## The CHOOSE Programs

Once the decision was made to use interactive computation in the input/output framework of CHOICEMODEL $H$, the problem then became one of developing the required programs. It will not be useful here to recite in detail the programming decisions that were made in order to implement the design features referred to earlier.

In brief summary, however, the special considerations that entered into program implementation were as follows:

1. Although the particular computer available for conduct of the necessary experiments (CDC 6400) has a large memory, the decision was made to program the information in such a way that storage requirements would be minimal. In this way the concepts could be implemented on smaller computers more easily. This criterion required that computer words be packed to almost the maximum extent allowable by the information content in the material.
2. It was decided that it would not be useful to program in a machinedependent language since that would inhibit the transfer of the programs to other machines. Since Fortran IV is a widely used problem oriented language available on most computers, the decision was made to enbed the CHOOSE procedures in machine-independent Fortran IV language as much as possible. However, since extensive character storage and character manipulation was required, some aspects of the resulting programs make the CHOOSE programs machine-dependent.
3. A method of efficient computation was sought in order to keep computational costs low. Hence integer computation was used wherever possible.
4. It was felt that the implementation of the system for specific alternative uses might require program modifications or extensions. This provided additional motivation to write in a commonly understood host language such as Fortran.

The resulting programs were developed over an extended period of time. The experimentation on the system was conducted for several different candidate sets which would be placed in the CHOOSE framework. The final CHOOSE programs resulting from this iterative design process are given in detail in Appendices $B$ and $C$ whereas the detailed example of use if given in Appendix A.

Although the complete interactive system has been designed and the associated programs written and tested, it would be inadequate to stop at this point. Some additional systems are needed for use by EVALUATORs in developing the appropriate models to be embedded in the CHOOSE system. The auxiliary systems developed for this purpose are covered in the next section.

## Auxiliary Systems for Building CHOOSE models

The CHOOSE system as described above includes the programs necessary for the computer application and a suitably instructed CHOOSER acting as part of the system, and working in such a way so that the "best" candidate for his particular requirements can be found or a small subset at least can be defined. As noted before, CHOOSE procedures do not guarantee that there will be a suitable candidate.

It is recognized, however, that if CHOOSE procedures are to be used easily to update data bases for the problems currently under study or are usually to be applied in completely new systems, then auxiliary systems would be needed. Two such auxiliary systems (programs) were designed to perform these tasks.

## The MAKEUP System

The original assumption was that the EVALUATORs would not necessarily have any strong background in computer science or computer programming. Their expertise would lie in the field of specialty for which the model is being created. Accordingly, the preparation of the model for computer use should be largely oriented in terms of preparation of data rather than of computer programming. Program MAKEUP was established to enable codification of the evaluation criteria (question and answer sets) developed by EVALUATORs in simple data card format. By executing these data cards with Program MAKEUP the two sets of intermediate information needed for further implementation of a CHOOSE model are generated. The two sets of information are as follows: (1) the nucleus of the main
program for the CHOICEMODEL, and (2) a block data subprogram for use with the second auxiliary program, CODEM, presented below.

This program was developed for use on two computers--the CDC 6400 and the Univac 1110. Details of the program and complete listing of the Fortran statements for the Univac 1110 version of MAKEUP are presented in Appendix E .

## The CODEM System

Once the evaluation criteria has been established, EVALUATORs study the various candidates and assign ratings to the various components of the evaluation vectors. These data again should be treatable by a stored computer program so that the EVALUATORs simply need to place the data in the required format.

Program CODEM has been designed for taking data in this form and converting it to the statements needed in the overall CHOICEMODEL program for the subject under study. Details of this program and its use are given in Appendix F.

The example in Chapter 5, and Appendices A and D, illustrates the use of these auxiliary programs in detail.

## Overview of CHOOSE System Model Development

Figure 6 on the following page gives a decision chart for use with a choice problem arises in order to determine whether or not a formal CHOICEMODEL should be implemented.

Note that at the first decision point a choice problem of some nature has already arisen. The questions that need to be asked at this point are as follows:

1. Are there enough viable alternatives to be considered?
2. Can presently available knowledge be organized in some discrete attribute-objective manner which would permit the required value judgments? If not, can additional information be developed?
3. Does the choice problem arise frequently enough so that it is worth the effort to develop a CHOICEMODEL rather than stay with the traditional informal choice techniques?

If the answer to this first question is "yes" then the work proceeds along the flow chart to the left. If not, then alternate methods for making a choice must be used.

Assuming the criteria are met, then the selected team of EVALUATORs will follow through the additional steps required. Note that it would be presumptuous to assume that the first step (creating. the appropriate


Fig. 6. Abbreviated Flow Chart for CHOOSE Methodology
question and answer structure) can be accomplished at the outset in a completely satisfactory way. Hence an iterative loop is provided so that the EVALUATORs will have the opportunity to review and modify, polish, update, etc., their criteria and their value judgments as the program develops. It is only after a sufficient amount of work of this nature has been done that the EVALUATORs should feel that the model is ready for release to potential CHOOSERs.

Applications of the CHOOSE system are presented in the next chapter. Details of the complete problem methodology are given in Appendix D.

## CHAPTER 5

## APPLICATION OF THE CHOOSE METHOD

With the development of the CHOOSE model concept and its implementation through the CHOOSE systems, it now becomes possible to speak of application.

The entire context from which these procedures were developed was that of closing the gap between researchers and potential users so that more of the work of researchers would find its way into practical application. One of the intermediate conclusions of the earlier material was that it was the researchers who were most distressed by the problem but they were also the ones who were best equipped to take the first steps needed to help close the gap. The major step proposed here, of course, is the preparation of the CHOICEMODEL for the entire family of candidates for any broad application they have in mind.

A decision to implement the CHOOSE procedures will reuqire them to make analyses and comparisons that they might not otherwise have been prepared to do. It does not seem that the development of a CHOICEMODEL would be worth the effort unless all of the following conditions were present.

1. There is a reasonably large group of CHOOSERs or potential CHOOSERs who would be willing or could be encouraged to use the CHOOSE system in their choice problem;
2. Sufficient stability exists in the candidate set so that constant re-evaluation of candidates would not become a problem.

When these conditions are met and it has been established that the effort is worthwhile, then the research group who now become the EVALUATORs must take the initial steps to construct the necessary choice framework. They must ask these questions:

1. Is it possible to map the candidates over a finite set of objective statements which will describe all of the important user oriented aspects which will be involved in the choice?
2. Can these objective statements be written in such a way that a CHOOSER's aspiration level to each objective can be effectively decoupled from the aspiration level for the others?
3. Is it possible to make a small discrete set of choices for each of the objectives in such a way that they can be totally ordered on the basis of the ordering relation "is as good as or better than?"

Once the decision has been made that the effort would be worthwhile and that the topic itself is amenable to such treatment, the actual procedure can begin. The researchers then begin to play the role of

EVALUATORs and follow the steps outlined in Chapter 4 beginning on page 64. The steps of this process are given in much more detail and with more of a computer orientation in Appendix $D$.

Rather than discuss the whole procedure in abstract terms, a simple example involving application for selection of home sound reproduction equipment is given in detail. This is followed by another major section describing the various frameworks available for researchers (EVALUATORs) to develop a CHOICEMODEL for watershed models.

## Example for Home Sound Reproduction Equipment

The problem of choosing an 8-track stereo cartridge player for home use was selected for example for instruction purposes only. Here the work of evaluation has been done by the Consumers Union of the United States (Consumer Reports, 1974, pp. 672-675). The additional role of the EVALUATOR can be accomplished very easily by placing value judgments on the various criteria. Although the CHOOSER is presumed to be "intelligent but uninformed" in the general context of model structure choice in hydrology, this example has been chosen because it corresponds to the experience of many CHOOSERs who will be viewing this example. All of them will have some idea--once they go through the choice processes for an 8-track stereo tape unit--as to whether or not the resulting choice is near optimal for them. Therefore, this should be building some confidence that the procedure does lead in the proper direction even in those situations where the CHOOSER is not so well informed.

Although the Consumers Union article in this case contains three pages of useful narrative information, the essential parts of the results are contained in Figure 7.

Note that although the basic work of evaluation has been done by the investigators at the Consumers Union of the United States, they have taken the usual approach of defining measurable attributes with no particular attempt to convert them to objectives in the context of a CHOOSE model. For example, they enumerate the width, length and height of the various units without implying that there is anything good or bad about the various dimensions. The display of such information then leaves it up to the reader to assign his own value judgments with respect to the individual attributes. This of course is not adequate for a CHOOSE model. In the CHOOSE model the EVALUATORs must assign "objective" type evaluations to the various units in order to make the model workable. This is the critical step since it is the point that translates the expert evaluation into a form accessible by users. Note that the term "objective" above refers to the EVALUATOR's perception of the objectives of the potential user. In this context the word "objective" is appropriate even though the judgment of the EVALUATORs may very well be subjective.

Some of the evaluation points of Figure 7 are rated in subjective terminology (fair, poor, excellent, etc.) although most of the attributes to which they are applied can be measured in the laboratory. The six attirbutes beginning with freedom from flutter continuing through ability

Fifteen different models from 15 different manufacturers were evaluated in accordance with the following criteria:

- Retail Price (\$40 to \$120)
- Size (3 linear dimensions; the largest linear dimension of any unit was 13-3/4")
- Weight (4 to 9-1/2 lbs.)
- Freedom from Flutter (poor to very good)
- Frequency Response (good to excellent)
- Signal-to-Noise Ratio (fair to very good)
- Freedom from Crosstalk (fair to excellent)
- Speed Accuracy (poor to very good)
- Speed/Line Voltage (good to excellent)
- Ease of Removing Cartridge (poor to excellent, with various footnotes pertaining to special features of individual units)

Fig. 7. Consumers Union Ratings of 8-Track Tape Decks
to hold constant speed with variation in line voltage are things which can be quantified with laboratory test equipment. It is then up to the EVALUATOR to assign the word description to the numerical values. The next item, however, is of a different type. It relates to a rating on the ease of cartridge removal. Presumably it is possible to perform time experiments using a number of different test subjects to arrive at a rating of this type. However, the ratings finally assigned would probably be based on comparative judgments of several EVALUATORs, without direct application of quantitative data.

The remaining attributes are strictly binary in that there are certain features which the various candidate tape players either possess or do not possess.

These concepts set the stage for the complete evaluation which was done for this case and is given in full detail in Appendix G. The information so developed was then carried through the various steps of the development of the CHOOSE model which were given in Chapter 4 and presented in complete detail in Appendix D. This example was used to illustrate the points in each of the Appendices A through F. With this implementation, the model is ready for any number of CHOOSERs, each of whom seeks to make a selection of an 8-track stereo tape player on the basis of his own individual requirements.

Table 2 shows the sample responses of two individual CHOOSERs to the queries presented at the display terminal by the CHOOSE system. Note particularly that while cost, weight and-size were not particularly important to CHOOSER A, he did insist on having both a fast forward and repeat switch available and also that the unit be capable of playing 4channel tapes. CHOOSER B, on the other hand, had a space limitation in which the maximum linear dimension could not exceed 12 inches and a price target of $\$ 75.00$. Since he already had a 4 -channel recorder, the 4channel option was not required. Also, he did not feel that it was important to have an automatic ejection switch on the uint. With respect to the other attributes/objectives, CHOOSERS A and B selected the same values. Whether the above inputs represent the initial entries by the CHOOSERS for their requirements or represent the end result after a certain amount of interaction, the display is essentially the same. In CHOOSER A's case, he is offered only the following rankings and item identification:


Table 2
Sample Responses in Choosing an 8-Track Tape Deck
$\left.\begin{array}{lccl}\hline \hline & \text { CHOOSER } & \text { Response } & \text { Analysis of Response } \\ \hline \text { Query } & \text { Cost } & \text { A } & 1-1\end{array} \begin{array}{l}\text { Unimportant } \\ \text { Cost under } \$ 65 \text { preferred } \\ \text { Would accept } \$ 65-90 \text { range }\end{array}\right)$

Note that this ranking means that he has failed to achieve his objective on certain items which are not yet identified. However, additional interaction with the CHOOSE system displays the following deficiency information:



```
CAND. MC. - 15
SHOFT N&FF - HFATHKIT GDQ%
    15. CFITFRT& NOT \becauseFT - C
MAFGTNP! FRSPQNCFEFFE 1, % 9%3
GFND. NG- NANF - SCNY'SCOR
    5. CRITFHJ& NOT NFT - iム
NAFGTNAI. FFSFCNOES APE Il:S
CAND.NE. F
GOE.NO.* 1-
```



Matching with the question numbers permits CHOOSER A to select criteria for modification.

CHOOSER B, on the other hand, receives the following choice information from the system:


Note that these choices are quite different from those offered to CHOOSER A. Either A or B can now accept the choices offered after viewing the deficiencies presented by the system or they can interact again,
re-examining their tradeoffs of the various objectives, in order to find an item which may come closer to their overall satisfaction.

Appendix A, The Interactive Instruction Manual, uses the input vector from CHOOSER A to illustrate all of the details of this step by step interaction.

## Application to Watershed Models

The EVALUATORs of the large family of watershed models available will find that a great deal of basic work has been done in attempting to organize the needed information. Such efforts could serve as a valuable starting point for organizing information into the CHOOSE framework. One of the most attractive frameworks for the present purpose is that developed by the World Meteorological Organization in their "Report of the Meeting of Experts on Inter-Comparison of Conceptual Models for Purposes of Hydrological Forecasting" (1972). While the main thrust of the work described was that of developing standard data sets for measuring the relative merit of watershed models, it also contained a questionnaire type summary of conceptual models currently available in several countries. At that time the report contained analyses of 21 models.

A copy of the organized information for a representative model is given in Figure 8 to illustrate the format. Much of the information on the form is clearly amenable to treatment by the CHOOSE method, and, using the additional decomposition techniques described on page qualified EVALUATORs could generate a complete CHOOSE model. In the example cited (the French Model SIMOUN), items 1 through 3 would become part of the model identification vector. Questions 5,21 and 24 would be of limited interest in the objective vectors except to the extent that a component based on confidence in the validity of the model might be developed from these criteria. The balance of the questions would be of the type that lend themselves nicely to use by the EVALUATORs in developing the required question and answer set and in making the appropriate assignment of ratings.

In this particular model note that question 8, APPLICATION RANGE, indicates time simulation can be from seconds to days. Criteria such as these can be directly embedded in the appropriate question and answer set. However, questions such as 17 and 18 which relate to the type of computer language used, and disk and core memory requirements would need additional thought. Some method of stating this in the context of general computer requirements rather than those related to specific computers or computer languages would be useful. It appears in this case that any appropriate computer with 64,000 -word internal memory capacity would be adequate.

Other classification schemes for hydrologic models appear in the literature which could also provide useful information (Ibbitt, 1972). Of particular interest may be the report begun by the late Dr. C.C.Kisiel of the University of Arizona and currently being completed by B. Popkin for the 1975 meeting of the International Union of Geophysics convening


Fig. 8. Sample of Organized Information for Watershed

Contress, Grenoble, France. This report summarizes the status of hydrologic modelling in 23 countries.

A good summary of the hydrologic model choice problem in the context of economics was given by Kisiel and Duckstein (1971). This report organizes a considerable amount of model specification information in a framework which is useful for approaching the cost-effectiveness problem in model choice. Cost-effectiveness questions should also be embedded in a CHOOSE model in addition to the criteria developed by the World Meteorological Organization.

Since the actual implementation for a CHOOSE model in any particular field does depend on the expert judgment of the selected EVALUATOR, no attempt beyond the above was made to implement a CHOOSE model for hydrologic CHOICE questions.

## CHAPTER 6

## CONCLUSIONS, RECOMMENDATIONS, AND EXTENSIONS

The problem which motivated the previous material was that of getting more effective application in the field of hydrologic models that have been developed by various research establishments throughout the world. The investigation led to the conclusion that the models were not presented and marketed in such a way that they would be chosen and implemented by practicing engineers in appropriate situations.

The problem was perceived as occupying a place somewhere in the middle ground between mathematical optimization models which all too often require analysis in continuous $n$-space, and information retrieval schemes which require massive analysis of documents for content without value judgment. In the first method, it is assumed that decisions can be selected from a continuum of available alternatives. In the second method the effort is aimed at supporting researchers who are seeking information as a foundation for future work and from which to synthesize more ideas. Neither of these meet the needs of those who seek an established technology to treat a problem in some sort of satisfactory manner.

By abstracting some of the more useful features of the two techniques it appeared a scheme could be developed which would not only aid in hydrologic model choice but would be applicable in any problem similarly situated. Accordingly, the CHOOSE model (Chapter 3) and the CHOOSE system (Chapter 4 and Appendices A thru F) were developed to provide a general framework for attacking such problems.

The feasibility of the CHOOSE system was demonstrated in a fairly simple case involving choice of an 8-track stereo tape player, and in personal selection. Several suggestions were given (Chapter 5) on how a group of qualified EVALUATORs with the requisite background in hydrologic models could implement such a scheme for hydrologic model choice using and extending attribute structures already developed in the literature.

The two following sections examine some possible extensions to the CHOOSE system, and some additional possible fields for application. The closing section suggests a few additional features that could be embedded in the actual computer programs.

## Possible Extensions to CHOOSE

The CHOOSE procedures as developed above have drawn on mathematical programming and information concepts to provide a methodology suitable for use under the conditions defined on page 70.

At present CHOOSE procedures are limited to completely decoupled questions and to discrete value judgments. Possible extensions of the CHOOSE procedures are as follows.

1. Multi-level Choice -- Basically it seems that there are two ways in which multiple level choices could be made,
(a) Multiple Levels for the Same Choice Problems. It may prove useful to partition the question and answer sets into two or more sets. The first set could contain the characteristics determined by EVALUATORs to be the most important in the CHOICE situation. The second sets could be called upon to supply additional criteria should the CHOICE problem become more difficult. In this way the psychological impact on the CHOOSER could be reduced by giving him fewer questions to respond to in the initial set and only going into the more detailed sets should the circumstances of the CHOICE problem require. For example, in the watershed question given on page 41 , some finer resolution of criteria for size of watershed might be needed in order to make a final choice between certain candidates. The extension of the CHOOSE model here then would be to allow for additional evaluation criteria to be presented when needed.
(b) Taxonomies of Models. Previous reference has been made to making an excessively detailed analysis. A possible extension of CHOOSE is to make it multi-level in the sense that it searches through some sort of taxonomy of models so that those CHOOSERs who are really unacquainted with the field could be led to the appropriate subclass of detailed models. For example, in determining behavior of watersheds some of the available models are strictly probabilistic models (often called streamflow models) and are based on past history, where others are models which examine behavior under varying inputs (rainfall-runoff models). Rather than embedding both streamflow and runoff models into the same CHOOSE model, it may be better to have the CHOOSE model which directs the CHOOSER to one or the other. Then at the next level additional CHOOSE models, each with its own precisely organized question/answer criteria aimed specifically at the class of model selected, would be called into action.
2. Real Value Judgments -- It may be useful to embed some questions in the model structure that permit answers in a continuum rather than at a limited number of discrete levels. This could permit some additional measure in the choiceordering which might be useful.
3. Coupling of Answers -- Mechanism could be included which would permit a few trade-off questions to be embedded. This could be similar to those embedded in Monarchi's SEMOPS (1972), or, more in keeping with the philosophy of CHOOSE, only discrete trade-offs could be considered. A modification of this sort might extend the range of application of CHOOSE procedures to choice problems not otherwise amenable to such analysis.
4. CHOOSER Selected Value Judgments -- It would be possible for the EVALUATORs to establish their evaluations of the decomposed objectives so that ordering could be on the basis "is as good as or better than" or "is as good as or poorer than" then the CHOOSER could be permitted to input his value judgment at the time of interaction. For example, suppose that a CHOOSE model has been implemented for selecting a
synthetic plastic for a particcular application. Different plastics have different ductilities. For one user higher ductility may be better than a lower one. In other cases the opposite may be true. Rather than having the EVALUATORs decide which is best, the selection could be left up to the CHOOSER.
5. Optimization -- As pointed out in Chapter 3, a classical mathematical optimization is not particularly applicable to problems thought to be treatable by the CHOOSE procedures. However, there may be circumstances where, by interaction, the CHOOSER has found several satisfactory candidates and now wishes to go to some form of scheme for attempting to approach the optimum choice. He then may wish to assign linear weights to some selected few of the criteria so that he can obtain some measure of performance which will aid him in his final choice. Such a scheme for "fine tuning" the decision could be built into the CHOOSE system.
6. Choice Combinations -- A major extension to the CHOOSE concept would be to a scheme which allows a combination of choices to meet the needs. For example, in rainfall-runoff models, only two things may be needed: 1) the time of peak flow and 2) the total volume of discharge. There may exist one model which furnishes both of these at the optimum level but at a cost which is unnecessarily high. A better choice, however, might be one inexpensive model which provides one part of the answer, and another such model for the other part at a total cost less than the model which does both. As another example consider the selection of a utility vehicle in which case the CHOOSER finds that he needs a 6-passenger pickup if he is only to buy one vehicle. But he may find that his overall needs are better met with a $1 / 2$ ton pickup and a sedan at less overall cost.

There are combinatiorial problems which would arise in attempting to incorporate such a feature, however, future development in computer speeds could make such extensions to CHOOSE possible.

## Possible Fields of Application for CHOOSE

As noted before the impact of the evaluation effort may very well. restrict CHOOSE procedures to those CHOICE problems for which the candidate set is reasonably static.

For the most part this implies that because a candidate has been chosen by one or a few other users it is not lost to future users. This would be true, for example, on any sort of information oriented choice since the consumption of information by one individual does not remove it as a resource for others. It would also be true in a market place where there is no scarcity of the quantity in question. For example, in implementing a CHOOSE model for selection of tires for passenger automobiles, many, many candidates could be included. The choice of one manufacturer's model tire would not preempt that choice for others, whereas a choice model based on rare Phoenician coins would not have this property since once a particular one is chosen, it has to be removed from the available set as choices for future users.

A number of systems can be visualized where the requisite "infinite" resource condition is met. The following are a few examples:

1. Basic Manufacturing Materials -- In a CHOOSE model for selection of basic materials for manufactured products, choice criteria could include strength, hardness, ductility, colorability, wear resistance, machineability, availability, cost, viscosity, temperature performance (low and high), behavior in high radiation environments, conductivity, density, solubility in air, and solubility in water.
2. Consumer Products -- Practically any high volume consumer product which can be evaluated in an objective way is a candidate for CHOOSE. The TPDEK8 example is a clear case of this.
3. Hydrology -- A broad range of hydrologic models other than streamflow and rainfall-runoff could be structured in the CHOOSE form. These could include snow melt models, ground water models, estuary models, ocean basin models, lake models, etc.
4. Trip Planning -- There would appear to be some possibility of adapting CHOOSE for use in planning travel. For airline travel, for example, some of the objective criteria which might be used to establish a choice might be lowest overall travel time, lowest time in the air, fewest transfers, "optimum" time for transfers, and/or lowest fare.
5. Software Selection -- The great proliferation of computer languages in the last few years has created a selection of languages in a particular application almost as bad as that for hydrologic model choice (U.S. News and World Report, June 24, 1974). It appears that use of a CHOOSE model as a guide in selection of computer languages in particular applications could be very beneficial.

Since the CHOOSE procedures are based on a rather extensive effort by EVALUATORs to establish appropriate values for particular questions and to rate candidates accordingly, any system that has an unstable data base might not be suitable for reduction to CHOOSE procedures. Two brief examples might illustrate this point.

In the selection of personnel to fill a particular position, a data base could be constructed; however, once the candidate becomes unavailable (by being chosen to fill the first opening for such a position, or simply by not being available when the time comes to negotiate an agreement) then he must be removed from the candidate set and the procedure begun again. The problem in personnel selection would be further complicated if a general data base were desired for choosing candidates for several types of positions. Here the value judgment questions would be confused. For example, EVALUATORs would be unable to place value judgment on M.S. in Chemical Engineering vs. M.S. in Electrical Engineering without some knowledge of the position for which they were structuring the data base.

As a second example, consider the problem of real estate marketing. While it may be possible to develop an appropriate question and answer
set and make a reasonable set of value judgments regarding them, the dynamic nature of the market may add and remove candidates from the set at a rate which makes it difficult to keep up the data base.

Potential Program Improvements
Possible extensions and improvements to the CHOOSE programs could be considered along the following lines:

1. Batch Mode -- Additional features could be embedded in the programs so that users could enter end inputs from the terminal but receive the output from the line printer in such a way that they could conduct their preliminary screening "off-line."
2. Readout of Data Base -- Some scheme could be built which would permit anyone dialing in to read out the entire data base for a particular CHOOSE problem into their own local facilities for local interaction. In this way they could avoid lengthy toll calls to link up their terminals with data bases some distance away.
3. Data Base Maintenance -- A scheme could be devised by which the candidate library could be modified directly from the keyboard. This would permit the EVALUATORs to keep the data base up to date more effectively.

## APPENDIX A

## USING THE CHOOSE SYSTEM

The material in this appendix consists of instructions prepared for users of the CHOOSE system (CHOOSERs). Accordingly, most instructions are given in the second person. Occasional parenthetical notes directed to those interested in the entire concept of the CHOOSE system are included. These may be ignored by CHOOSERs.

To follow these procedures requires no knowledge of computer programming or of the method of preparation of the data base. Some minimum acquaintance with an interactive terminal--whether teletype or cathode-ray--is desirable. Beyond that the CHOOSER simply needs:

1. to know that a panel of experts has made appropriate studies and have organized information (under a named program) that can be of use to him in making a choice for some problem with which he is confronted,
2. to have access to an interactive terminal, and
3. to know that choice assistance is available to him through the terminal through telephone connections to the CDC 6400 computer at The University of Arizona.

Throughout the description of the interactive procedures which follow, reference is made to a particular choice problem involving selection of an 8 -track stereo tape player. The data base/interactive program for this problem is called TPDEK8. This is the same example program referred to from time to time in the main text. The steps described here are illustrated by reference to reproductions of actual teletype output obtained during execution of the program with the TPDEK8 data base. For any other previously prepared and available data base, it is only necessary to substitute the appropriate data base name for TPDEK8 wherever it appears.

The data base/interactive program was created in accordance with the instructions contained in Appendix D. At the present time, TPDEK8 and CHOOSE are brought up from magnetic tape storage to CDC mass storage devices each day as needed for test or demonstration. The small CDC "batch" program needed to accomplish this is given at the end of this appendix. Looking ahead to making such information always available (as perhaps a "public-utility" program for worldwide users on a 24 -hour basis) it would be necessary for the programs to be available on mass storage devices (magnetic drum, disk, etc.), or be readily accessible for mass storage on request from interactive users. For the present, funds are not available for the mass storage charges that would arise in an attempt to keep the program in permanent disk storage. But, for purposes of this instructional manual, it is assumed that CHOOSE and TPDEK8 have been brought up on mass storage devices accessible to the interactive user.

Since the CHOOSERs are not required--or expected--to have access to this entire document, the instruction manual portion of this appendix, beginning below, repeats many points that were introduced at other points in the text.

## Interactive Instruction Manual

The CHOOSE procedures have been designed to make expert evaluations of various choices of alternatives accessible to you by a simplified approach that avoids extensive research and analysis on your part. By using these procedures in conjunction with any prepared data base, you can avoid the mental paralysis that arises from having too many alternatives to analyze or the uncertainty that arises from not knowing enough about the alternatives that could be considered.

These procedures begin by obtaining and examining your needs (goals) and resources (constraints). This is done by the presentation of a series of questions which are constructed as carefully as possible to permit you to make direct replies without extensive reflection or computation. Following this, the computer displays those alternatives (candidates) that have come the closest to meeting your stated requirements. If you are satisfied with one or more of the candidates presented, the problem is solved. However, as is often the case, some of the needs will not be met within the constraints given. This leads to the interactive dialog.

The key feature of the CHOOSE procedures is that they not only permit but also help guide a dialog between you and the computer. This feature, when used properly, can illuminate the problems, and, by using information you supply on acceptable modifications to your needs and resources, move you toward the best alternative candidates for you. All of this is done without exposing you to any of the candidates that fail seriously. Hence, it reduces the choice problem to manageable proportions.

## Steps

Discussion of CHOOSE breaks down quite naturally into eight steps. Before going into detail, these procedures are enumerated below. A detailed discussion begins on page 89 .

1. Becoming familiar with interactive terminal operations.
2. Establishing communications between the interactive terminal and the desired procedure.
3. Establishing identification and mode of operation.
4. Obtaining inputs from the CHOOSER. (Section I of CHOOSE.)
5. Reviewing and modifying (if desired) the input. (Section II of CHOOSE.)
6. Summarization and encoding of input. (Sections III and IV of CHOOSE.)
7. Interaction. (Section $V$ of CHOOSE.)
8. Termination.

The process is much easier to use than would appear from the detailed instructions below. So don't become intimidated; just give it a try. The easiest way to become familiar is to dial in and begin using the system. After the first two steps, the process is largely selfexplanatory.

To familiarize you with the method, a special data bank named TPDEK8 has been constructed. It involves a simplified version for selection of an 8-track stereo tape deck for home use. By experimenting with various goals and constraints, you can see how the selection process/interactive dialog works, and how it produces different recommendations under different conditions. The discussion which follows uses TPDEK8 as the example.

Step 1. The Interactive Terminal. Obtain some minimum familiarity with the teletype or cathode-ray terminal you will be using. Customarily, there will be a small operating manual available at the terminal for this purpose. Note that, regardless of the terminal in use, no message is transmitted to the computer unless the RETURN or SEND key is operated. Acknowledgment of receipt of a message by the computer is given by a LINE FEED signal returned by the computer.

Step 2. Establishing Contact with the Computer. Establish initial contact over commercial telephone lines by dialing 602 884-3194. The computer will respond with identification material and end with PLEASE LOGIN followed by a carriage return and a line feed. You must then respond with LOGIN followed by a RETURN' (carriage return). This begins the interaction, and you must now send a series of messages in response to specific inquiries from the computer to establish your right to use the computer, the job size, time limits, and to load and execute the appropriate programs. The entire sequence for this appears in Figure 9, where underlined messages represent inputs to be entered by you at the keyboard and transmitted to the computer by depressing the RETURN key. Those messages shown but not underlined are supplied by the computer. Explanatory notes are given at the right side of the figure.

When the entries given in Figure 9 have been received and accepted by the computer, the connection of the terminal to the computer is completed. The interactive program headings appear next.

Step 3. Preliminaries. Refer to Figure 10. For the record, the computer responds with its address (line 1) and the name of the prepared data base with which interaction is contemplated (line 2). The next two lines give the date that the criteria set was last revised and the date that the candidate information was last updated. (Updating of criteria can take place as new needs or constraints are perceived or old ones found to be of little consequence. Such changes will require an updating of data for each of the candidates at the same time. However, updating of individual candidate information will generally take


0ム/18/75 LRGGFD JN AT 19.39.04. WITH LSEF-ID FL
FEUIP/FMFT 11/1s
CONNAND- ATTACH(TFDFKG,TFDFKG,ID=FFL)
PF CYCLF NM. $=$ NO1
CQNNAND- ATTACH(CHCESF, CHROSF,ID=FFL)
FF CYCLF Ne = 001
CONNAND- CRNNFCT(INFIT, RLTPLT)

NOTE: Entries made from the keyboard are underlined. Others are supplied by the computer.

CRNMAND- FFL, sחOnO.
CENNAND- FTL, $\angle$.
CONNAND- XFG
QPTION= LQAD=TFNFKQ, CFOCSE
RPTIEN=FXFCUTF=THTFKF

Fig. 9. Step 2. Establishing Connection

```
LINIUFRSITY OF ARIZONA, TI'CSQN, AFIZONA 8S7?1
INTFFACTIVF CHOIC.F.FFOGRAN - TPDFKR
    CFITERIA(G AND A SFT) LAST LPPATFE 11-1G-74.
    CANDIDATF INFOFNATIQN LAST IFRATEL 11-1G-74.
FEALYY YES OF NE - YES
QUESTIENS ARF ANSWFFFD EY GIVING A NLMFFICAL FFSPONSF
IN THE RAN'GF FFFMITTFD, FY TYFING
    A (TQ ABORT THE FNTIFF PROCFSS),
    S (TO SKIF AHFAD TO THF NEXT NAJOF SECTION),
    R (TO RETLPN TE AN EAFLIFF NAJQF SFCIION FOF RFVIF!:), OF
    \ (IF Y&| DON'T KNOG HOW TE FESFQND).
YOUR RFSFONSES AFF FNTEFFD IN THF FLACF FFOVINFR AFTFF
THE MESSAGF #GIVE VALIE FFSPQNSF - #. FACH FFSPQNSF NUST
RE FELLQVED RY STPIKING THF #FFTURN# KFY TO TFANSNIT YOUF
PFSPONSE TQ THE COMPLTFF.
READY\ YFSQFNO-NO
HAVF COIFAGF -- YOU WILL PF APLF TO APOFT WITH A ANY TIMF A
CLFSTIEN IS ASKFD. NOK, ONF NOFF CHANCF.
RFADY\ YES EF NQ -
CLFSTIENS ARE ANSWFRFD PY GIVING A NINEFICPL PFSFONSF
IN THE FANGE FFPNITTFD, PY TYPING
    A (TE APOFT THF FNTIFF FFQCESS),
    S (TO SKIF AHFAN TQ THF NFXT NAJQF SECTION),
    R (TQ RFTUFN TQ AN FAFLIFP MAJOP SFCTIQN FQF FFVIFW!), QF
    \ (IF YQU DQN"T KNQU HQW TO FESPQND).
YQUF FESPQNSFS AFF FNTFFFD IN THF PLACF PRQUIDFD AFTFF
THE NESSAGF #GIVF VALID FFSFONSF - #. FACH FFSPQNSF NIIST
BF FRLLQUFD EY STFIKING THE #FETUPN# KFY TO TPANSNIT YOIR
PESPRNSE TE.THF CQNPI:TFF.
FFADY\ YES QF NO - YFS
```

Fig. 10. Step 3. Preliminaries
place whenever additional candidates become available or when experience or judgment jndicates that prior evaluations require adjustment in one or more of the criteria.)

Following the readiness question which next appears, indicate your readiness to proceed by typing YES (or simply Y) or NO (or simply N ). Note that, even though the set of answers offered by the computer seems to be limited, there are actually a number of answers that are always acceptable. For easy reference, these are given in Figure 11. For example, by typing $S$ (for $S k i p$ ) you can cause the interactive program to skip to the next major section. By typing $R$ ( for Return) you can make the program return to an earlier section for review or change of entries. By typing \# you can cause suppression of detailed messages (if they are currently appearing) or cause detailed messages to appear (if they have been previously suppressed). At any time, if you are confused as to how to respond, you may type ? and the program 1) will give you a list of acceptable responses at that point, 2) return to giving out detailed messages, and 3) repeat the previous question for your new response.

After the preliminary information is presented, and you have elected to proceed, you will be led automatically through the major sections of the CHOOSE procedures. Each of the sections represents a phase of the choice process and is identified at the interactive terminal by Roman numerals. The form of response you are to use is as indicated in Figure 11--except in certain cases where special tables are presented for entry of your answers. Note that, if at any time you wish to abort the process, you can do so by simply sending A. Note also that the system has been made largely foolproof. When unacceptable entries are received they are rejected, some conment is then offered on how to make a valid response, and the opportunity is presented to try again. Even though some part of the process may be skipped or deferred to a later time, the system at all times retains "default" values in storage that act in a fail-safe role. About the only mistake that will wipe out previous work is to respond with the abort signal (A) inadvertently.

Rather than explain each of the major sections in detail, each is discussed briefly and reference is made to the appropriate figure to show the behavior for sample answers for the illustrative case. As before, in each figure the underlined entries represent those made by the user from the keyboard. All others are supplied by the computer.

Step 4. Section I of CHOOSE. Eleciting Input Data. Refer to Figure 12. Here you will enter your needs and resources in the criteria set established for the choice problem you have selected.

First you are asked whether you want the complete text with each question. If you have no prior knowledge of the questions and the choice of answers available, you should enter YES. However, if you have access to the questions and the answer set, and perhaps have established the range of answers you intend to supply, you should enter NO. [The latter will, of course, depend on the criteria set being

| Respons <br> Always Accep | Meaning and Comment |
| :---: | :---: |
| numeric | Where numerical responses are expected, the set of integers over which response is to be made is given. Even though numerical response is solicited, the alternative responses given below can be given, if needed. If an invalid integer is entered, the program will reject it with appropriate message and, except in tables, will solicit a new figure. |
| A | Stop the entire procedure. This stops execution of the program. To start over from the beginning it is necessary to respond to the next COMMAND request from the computer with the last three statements of Figure (COMMAND-XEQ, etc.). To terminate the connection, see Step 8 . |
| R | Return to an earlier major section. |
| S | Skip ahead to next major section. |
| \# | Activate or suppress the detailed messages. This entry causes the detailed messages to be switched on or off. "Off" is desired in order to speed up the process. For those unfamiliar with the procedure "On" is the preferred mode. "On" is established initially by the program. This switch can be made as often as desired in the program. |
| ? | Type ? when uncertain what to do. Explanation will be offered, and the question will be repeated. |
| Sometimes Acceptable |  |
| D or blank | No response will be given at this time. Note that a blank is made by space bar followed by carriage return. Carriage return alone is inadequate, since it is interpreted as no message and the computer simply waits for your valid response. |

Fig. 11. Standard Responses and Their Meanings

```
SFCTION I. MODFL CHOICF PFOGFAN RFGINS.
D@ YOL WANT THF COMFLETF TFXT H:ITH FACH GlPSTION
NQTE - YOU NAY FEVFFSE THIS DECISION ANY TINF LATFF
BY TYPING - IN ANSLEF TO ANY GUFSTIRN.
RFPLY -YFS
THERF.ARE 3 W'AYS TO FNTFF YOL'P CHOICF
GRITEFIA. SFLECT ONF.
    1 = FACH GIFSTION INDIVIPLLY PFFSENTFD.
    ? = PFFVIDUSLY PFEFARFD ANSWFFS TQ RF FNTFFFR.
    3 = PFECODFD NLMFFFS AFF T& EF FNTFFFD.
PFPLY -1
    FACH ClFSTIQN IN USF kILL EF CALLFD IN TIFFN. THF l'SIAI.
NUMFPICAL AND SYMPQLIC RESFONSFS AFF TO PF LSED.
RFADY\ - YFS
Q. 1. ANSWFF FANGE - 1 T0 G.
FFCARD YOLIF ANS!IFRS #1# #1#
C. 2. ANSWEF RANGF - 1 TO 6.
PFCORD YOLF FNSWFRS #1# #1#
C. 3. ANSWFR FANGE - 1 TE 7.
RECQFD YOUR ANCHEFS *1## ###
Q. 4. ANSWFR FANCF - I TE 6.
FFCERD YOUF ANSUFPS #1## #1#
G. 5. ANSNEF FANGF - 1 Te 5.
FFCEFD YOLR ANSWFFS #2" #?#
Q. 6. ANSVEF FANGE - 1 T0 5.
RFCORD YRLIR ANSWEFS #B#
    ###
C. 7. ANSWFR FANGF - 1 TO S.
FFCORD YOLIR ANSWEFS #F# # #
```

Fig. 12. Section I. Opening and Method 1 of Entry
generally available to the prospective user, wherever he may be situated. Technical periodicals could be used for this purpose. In this way the user will be able to prepare his criteria in advance (in an "off line" mode) with greater opportunity to reflect on the questions and to discuss some of his proposed responses with his colleagues. The interaction in this case can be reduced to a brief "hook-up." When such advance work can be done, it is the recommended procedure.]

Next you will be asked in which of three forms you wish the questions to be presented. The three alternatives are:

Method 1-- Here each question is presented individually, and its answer range is given. If the full text condition (as previously established) is ON , the question will be stated along with a numbered set of answers followed by a place for two responses. If the full text condition is OFF, only the place for your two responses is provided. This permits you to enter a range of answers if you are uncertain which specific answer to select; that is, you are allowed to equivocate, if you need to, to keep the process moving.

The basic rules for selecting responses (answers is to give, as a low answer, the minimum acceptable condition in the context of your particular situation, and for a high answer, the best you can expect or can offer in resources as the high answer. Both answers may be the same. Figure 12 shows how some responses of this nature are entered. (Note that it is up to those analyzing, developing the coding criteria, and evaluating and coding candidate information to construct meaningful questions and code them in the proper numerical sequence so that the low and high numbers of answers are suitably rank-ordered.)

Method 2--Method 1 is the most tedious but must be relied upon if you have been unable to prepare your answers in advance. Method 2 is the preferred one if you have been able to prepare your responses in advance.

In the latter case, select Method 2 for the form of data and enter the numbers directly into the tabular form. Refer to Figure 13. Note that the questions are displayed 14 at a time along with the maximum allowable answer and the answer currently stored for the low value. Those marked with an asterisk are those for which no response has yet been obtained; those showing no asterisks were answered when Method 1 was in use (Figure 12). At the end of this display the question OK? is asked, to which you may reply YES (or 1 or Y) or NO (or 2 or N). If you enter a NO, a new line is displayed for your entry of the appropriate new or revised numbers. Where no change is needed, blanks can be substituted.

When you have sent the new information, the computer analyzes and reproduces the information and notes any errors you may have made. Again it asks OK? If you respond YES, the procedure advances to an analysis of the high values of your response and continues in this manner until all questions have been displayed and/or accepted. In Figure 13 this occurs when the high answers for the last 2 of the

## 

OLESTIONS WILL APPEAR IN ? TAFLF(S). ANNQTATIQNS AFF -

* LNANSWFRFD GUFSTICN
- FQR ERFGR IN LAST ATTEMPT

| CLIES. NQ. | 1 | $?$ | 3 | 45 | 6 | 7 | 2 | 9 | 10 | 11 | $1 ?$ | 13 | 14 |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| MAX. ANS. | 6 | 6 | 7 | 65 | 5 | 5 | 5 | 5 | 5 | 5 | 2 | ? | ? |  |
| LQW ANS. | 1 | 1 | 1 | 1 ? | 3 | 1* | 1* | 1* | 1* | 1* | 1* | 1* | 1* | aK\ |
| ENTFR NEW |  |  |  |  |  | 3 | 3 | $?$ | ? | ? | 1 | ? | $?$ |  |
| LQW ANS. | 1 | 1 | 1 | 12 | 3 | 3 | 3 | 2 | $?$ | $?$ | 1 | ? | $?$ | aK\} |
| HIGH ANS. | 1 | 1 | 1 | 1 ? | $\triangle$ | 5* | 5* | 5* | 5* | S* | P* | $?$ | $?$ | QKJ |
| ENTFF NEW |  |  |  |  |  | 4 | 4 | 3 | $?$ | < | 1 | $?$ | $?$ |  |
| HIGH ANS. | 1 | 1 | 1 | $1 ?$ | 4 | 4 | 4 | 3 | ? | 4 | 1 | ? | ? | QK\ |
| OLFS. NO. | 15 | 16 |  |  |  |  |  |  |  |  |  |  |  |  |
| MAX. ANS. | $?$ | 6 |  |  |  |  |  |  |  |  |  |  |  |  |
| LQW ANS. | $1 *$ | 1* | OK\} | Na |  |  |  |  |  |  |  |  |  |  |
| FNTFR NFW! | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| LCW ANS. | 1 | 1 | OKI | YFS |  |  |  |  |  |  |  |  |  |  |
| HIGH ANS. | 2* | \&* | OK | NO |  |  |  |  |  |  |  |  |  |  |
| ENTER NFW | $?$ | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| HIGH ANS. | ? | 1 | OK | YFS |  |  |  |  |  |  |  |  |  |  |

Fig. 13. Using the Method 2 Entry Option
total 16 questions have been accepted by answering $Y$ (on the last line).

Method 3--The third option permits direct entry of your criteria in octal coded form. This option is only useful if you have found, during previous interaction, what the coded computer words are for your criteria. This will generally mean that you have worked interactively before and wish to pick up from where you previously left off. When this is the case, you can select Method 3 for input format and then enter those coded words displayed under Section IV (see below) of the earlier run. The form displayed for entry in this is identical with that shown in Figure 15 except that after each label you are expected to enter the appropriate 20 octal characters. For example:

$$
\text { NEED2H }=00000000000000000041 \text { (Carriage return) }
$$

Step 5. Section II of CHOOSE. Review of Input. On arriving at Section II, you will be given the opportunity to review the inputs you established in Section I. This is illustrated in Figure 14. Here, when a REPLY is solicited by the computer, you simply enter the number of any question you wish to review. In the example, question 12 was brought up for review, the maximum available answer indictted, the currently recorded low and high answers displayed and spaces given for the new response. Here, the response $S$ was given since the answers shown for 12 were considered satisfactory, and no other questions were thought to need review. The $S$ causes a skip to the next section.

Step 6. Sections III and IV of CHOOSE. Summary and Encoding of Input. Refer also to Figure 15. Section III gives you the opportunity 1) to obtain a summary of all input information, whether entered from the keyboard or default values entered by the program for unanswered questions, 2) to obtain a listing of unanswered questions only, or 3) to skip over to the next section. This is done by answering 1,2 , or 3 , respectively, to the initial query of the section.

In the example, 1 was selected, and a complete list of recorded responses to all questions was obtained. Note that those low and high values recorded and marked with an asterisk are default values supplied by the program. (In the example, none are so marked, since all questions were answered from the keyboard in Section I.)

Section III closes with a question of whether you are satisfied with the complete list. A response of YES sends you to the next section. A response of NO, returns you to Section II, where you again have the opportunity to call up specific questions and to modify your responses.

In the example, the NO response activated Section IV. Encoding the Answers. Refer to Figure 15. Here, no interaction is required, and the computer encodes your response into octal words to prepare for the search and comparisons which will be needed. For the record, the four coded words are displayed at the terminal. They may be used for direct entry of coded information in Section I (Method 3) on some subsequent run. The program then moves automatically to Section V.

SFCTIONII• RFVIFW OF G.ANDA.
HEFE YQU MAY ASK FQF ANY GLESTIDN TR RF PEFSFNTFD AGAIN. TO LEAVE THIS SECTIQN YOU NLST TYFF S AS YOLFFFSPRNSF. QTHEFUISF GIVF THF DESIRED GU'FSTIRN NLIMPFF AS YOIF FESPQNSF.
REFLY - 12
G. 12. ANSWEF FANGF - 1 TO ?


SECTIEN III• SLIMAPY OF ANSUFFS.
CHQOSF METHOD CF FRESENTATIQN QF FESILTS.
$1=A L L$ ANSWFFS, $?=$ NONE, $3=L N A N S W E R F D$ OLFSTIQNS DNLY. REPLY - 1

THOSE ANSVFPS NAFKED RY AN * AFF ANSWFFS SUPPLIFE PY THF SYSTEM SINCF YQL HAUF NOT YET ANSUFFFD THFN.

| $N O$ | $L Q W$ | $H I G H$ | $M A X$ |
| :---: | :---: | :---: | :---: |
| 1 | 1 | 1 | 6 |
| 2 | 1 | 1 | 6 |
| 3 | 1 | 1 | 7 |
| 4 | 1 | 1 | 6 |
| 5 | 2 | 2 | 5 |
| 6 | 3 | 4 | 5 |
| 7 | 3 | 4 | 5 |
| 8 | 3 | 4 | 5 |
| 9 | 2 | 3 | 5 |
| 10 | 3 | 3 | 5 |
| 11 | 2 | 4 | 5 |
| 12 | 1 | 1 | 2 |
| 13 | 2 | 2 | 2 |
| 14 | 2 | 2 | 2 |
| 15 | 1 | 2 | 2 |
| 16 | 1 | 1 | 6 |

De YOll WISH TO MODIFY ANY FFSFONSFSS -
$N$

Fig. 14. Sections II and III Illustrated


Fig. 15. Section IV. Coded Output Example

Step 7. Section $V$ of CHOOSE. Interaction. A brief explanation of the rating scheme is given here and in Figure 16 prior to discussion of the example. This should be adequate to establish the mechanics of the interactive process for the immediate discussion. The major section of this appendix entitled "Strategy for Interaction" (page 106) gives the detailed meanings of the ratings and, beginning on page 110, outlines some techniques for interaction which can assist you in moving efficiently towards those candidates most suitable for you.

The primary rating places each candidate in a class numbered from 1 through 8, where those rated 1 are those that meet your requirements handily, and those rated 8 are extremely unsatisfactory. Candidates with intermediate ratings are considered poorer candidates. The intermediate ratings--2 through 6--indicate decreasing likelihood (as the numbers increase) that the candidate ought to be selected or considered further.

A secondary rating scheme is used for those candidates with primary ratings of 1 and 6 . For those with a rating of 1 , the secondary rating is a numerical count of the number of criteria for which the candidate exceeds your requirements. Hence, the larger the secondary rating, the better (presumably) the candidate would be. For those with a rating of 6 , the secondary rating is a numerical count of the number of criteria on which the candidate failed to be rated 5 . In this case, those with the larger number are less likely to be selected.

Rather than display each candidate with its rating, the CHOOSE system presents only the top candidates and displays more only on request. Returning now to the example and Figure 17, note that, after an affirmative answer to a readiness question, the top ten candidates are displayed. The information appears in the following order: 1) candidate number, 2) the candidate's primary rating, 3) a short title or name for the candidate, and 4) the secondary rating (zeroes are used when there is no secondary rating).

Typically, the top candidates will not have a high enough rating, and often there will be a number of ties. The first set of ten in the example shows that no candidate fared better than 5 (indicating that none of those candidates met all of the conditions) and that there

```
Primary Rating
Meaning
1 Outstanding. Meets or exceeds ali requirements including the most optimistic statement of your needs and the most severe of your constraints. A secondary rating based on the degree by which the rating is exceeded is given. The higher the secondary rating, the better the candidate.
2
Fully Acceptable. The standard candidate rating is equal to or better than your minimum requirements, and the optimistic candidate ratings are equal to or better than your highest performance requirements.
3
Acceptable A. The most optimistic candidate ratings meet or exceed your highest desired performance.
4
Acceptable B. The standard candidate rating is equal to or better than your minimum requirements at every point.
5 Marginal. The most optimistic candidate ratings exceed your minimum requirements at every point.
6
Bad. There are some instances where you would have to reduce your minimum response by one step in order to achieve a rating of 5 . The number of instances where this would be required is given as a secondary rating criterion.
7 Very Bad. You would have to reduce your minimum response (requirements) by more than one step in at least one instance in order to achieve a rating of 5 .
8
No Hope. There are some points in direct conflict. Basically you have responded to some questions with a requirement that cannot be met. Example: You have asked that the candidate provide peak flow information, and the candidate rated 8 cannot supply it.
```

Fig. 16. Brief Outline of the Rating Scheme

```
SECTION V. INTERACTIVE SFAFCH PROCEDURE.
READY\ YES OR NO - YFS
LIST IN APPARENT DECRFASING DESIRABILITY.
    2. JVC ED1103 1
    16 PANASONIC RSROIUS 2
    7.6 SYLVANIA ETZ750W 3
    10 REALISTIC TF1G6 4
    37 GFN. FLFCTRIC TAS60 0
    4.7. LAFAYETTE RK8? .. . 0
    5.7 SQNY TC2O8 0
    6 7}\mathrm{ ( BSR.TDSS 0
    8 CONCORD CD84 O
    9 7. TOSHIBA KT84 O
5 CHOICES.
REPLY - }
    11 7 WARDS CAT.NO. 6814 0
    12. 7 SUPERSCOPE TD2R 
    13 5FARS 9105 0
    14 HITACHI TPQ14AD O
    15 7 HEATHKIT GD28 0
5 CHOICES.
REPLY -3
CAND. NO. - ? 2
SHORT NAME - JVC ED1103
    2. CRITERIA NOT MET - &
MARGINAL RESPQNSES ARE 2. 5 9 13 14 1S
CAND. NO. - 15
SHORT NAME - HEATHKIT GD28
    15. CRITERIA NOT MET - . 3 4 9
MARGINAL RESPQNSES AFE 1: 2 7 13 14 15 16
CAND. NO. - 1
SHORT NANE - PANASQNIC FSROIUS
    1. CRITERIA NOT MET - 4.9.
MARGINAL PESFCNSES ARE.2 5 6 6:7.131415
```

Fig. 17. Section V. Example of Interaction
were two tied for that position. The balance of the first ten are all rated 6, or poorer.

Following this, you are given five choices for the next step. The choices are:

1. JOB COMPLETED. TERMINATE RUN. This entry is self explanatory. It could be used immediately if a choice can be made from the information already displayed. It would also be useful if, prior to proceeding, you wished to study reference materials on some of the candidates showing promise. Normally it would be used only after a certain amount of interaction to modify criteria has taken place.
2. RETURN TO Q AND A FOR ADJUSTMENTS. A response of 2 would return you to Section II (Step 5) for adjustment of criteria. This entry would normally be used only to correct an error just discovered or to make a number of major changes to input information as a result of serious failure of essentially all candidates.
3. INQUIRE ABOUT SPECIFIC CANDIDATES. This is the real interactive dialog entry. It places you in a special mode where, by candidate number, you can ascertain the precise items on which the candidate failed to meet the stated input conditions. Further, the conditions considered marginal (as determined by the spread given between the high and low values) are displayed. This permits you to reflect on your responses to particular questions and make changes that you feel might be acceptable in your need/resource input criteria. By looking at several candidates which have essentially the same rating, the particular trouble spots can be illuminated and some exploration can be made to see where the greatest opportunity for improvement lies. This gives you the opportunity to compare the trouble spots in some detail with your criteria so that you can select, for further examination, those candidates that seem most likely to be within reach of your goals and constraints, as modified during interaction.

The normal exit from this option is to type $S$ when a candidate number is requested. This option also permits you, by typing $R$, to have specific questions displayed on call. In this way, modifications can be introduced without return to the earlier steps. The return from this question modification option is back to this option 3, so that the dialog can continue until you have made all the progress towards choice that you are prepared to make during the present interactive session.
4. EXTEND LIST. This option permits you to ask for extended lists as illustrated in Figure 17. This is useful if a number of candidates are tied and you wish to display those that may also be tied but could not be displayed within the numeric limitation (10) of earlier lists. It is also useful if you wish to locate a particular candidate with which you have some particular familiarity or interest. While this can be done by calling for that particular candidate under option 3, by doing it here, you can get a better perspective of its relationship with other candidates.
5. RE-DO LIST. With this option you can cause the list of candidates to be re-done and the top ten candidates to be displayed again. This option may be followed by 4 (EXTEND LIST) if desired. This option should be selected if you have made enough changes in your input criteria (goals and constraints) so that you need to re-establish ratings. This is particularly useful in interaction since the modified criteria that the dialog produces may not only make those candidates you are working with appear better, but they may cause other candidates-previously rejected--to come up for reconsideration.

Returning now to discuss the example (Figure 17), note that choice 4 was made initially to extend the list and display the five models not originally displayed. Note also that the highest rating achieved by any unit was No. 6 indicating that the most favorably situated unit did not meet the minimum requirements.

The second choice made was No. 3 indicating that it is desired to examine some of the candidates to see on which specific points they fail. Here the top candidate (No. 2), the bottom candidate (No. 15) and the second candidate (No. 1) were examined. All show that criteria 14 and 15 (among others) constitute trouble spots.

The example is continued in Figure 18 on the next two pages. Criteria have now changed since this represents the presentation after some interaction has taken place. When operating under choice 3 (INQUIRE ABOUT SPECIFIC CANDIDATES), a response of $R$, when a candidate number is solicited, places you in the position shown at the top of page 105 where a question number is solicited. In this way you are permitted to examine the relationship of the candidate currently under consideration with respect to specific questions. When the response to QUES. NO. is R, this places you in a mode in which you can actually change your previous responses. The entries on page 105 illustrate this procedure for criteria 14 and 15 which were previously noted as troublesome.

A reply of $S$ to QUES. NO. returns you to the candidate mode, and a reply of $S$, when in candidate mode, returns you to the mode of five choices. Both of these are illustrated at the bottom of page 105.

Using the five interactive options given above, you can explore your various options and perhaps find a single feasible candidate. In this case you may wish to terminate the program. Alternatively, you may find a small subset of the candidates that meet your goals and constraints at some acceptable level. This information, together with the display of the weak points, may give sufficient information so that further study of the reference material supporting the selected subset of candidates will lead you to a choice without further interaction. Or, you may find that your goals and constraints cannot be met in any satisfactory way by any of the candidates in the data base. In this case (and particularly if the data base is maintained on an up-to-date basis) the conclusion that no candidates meet your requirements may be warranted. If this is the case, then alternative methods for solution of your problem must be investigated. In all of the above cases, the current

```
5 \mp@code { C H O I C E S . }
1. JQB CQMPLFTED. TFRMINATE FLN.
2. FFTLIFN TE O AND A FAF ADJLSTMFNTS.
3. INGUIRE AFQLT SPFCIFIC CANDIDATFS.
4. FXTFND LIST..
5. RE-DC L.IST.
REPLY - 3
GIVE CANDIDATF NQ., S TO FXITTHIS STFP, QF F TE
EXAMINE DETAILS &F LAST CANDIDATE NAMED.
CAND. NO. - 1
SHOFT NAMF - PANASQNIC FSGO1IS
    1. CRITERIA NOT MET - 0
NARGINAL RESPQNSES ARF 6 6 7 9 13
GIVE CANDIDATF NQ\bullet, S TQ FXIT THIS STEF, UR R T@
EXANINE DFTAILS QF LAST CANDIDATE NFMED.
CAND. NO. - - JVC E ED1103
    2. CRITEFIA NRT MET - O
MARGINAL RESPCNSES AFF 9 13
GIVE CANDIDATE NQ., S TO EXIT THIS STFF, QF F TO
EXANINE DFTAILS GF LAST CPNDIDATF NANED.
CAND. NE.- - GFN. - ELECTFIC TAS60
    3. CRITEFIA NCT NET - O
MARGINAL FESFQNSES ARE 7 9 13
GIVE CANLIDATE NQ., S TO EXIT THIS STFP, ØF F TO
EXANINE DETAILS AF LAST CANLIDATE NAMED.
CAND NO. N- SFAPE 13
    13. CRITERIA NOT MET - O
NAFGINAL PESPONSFS AFF 7 9 13
```

Fig. 18. Step 7. Section V. Interaction

```
QUFS. NO. - 15
GUFS, 1S YRUP INFLT 1-2, AND S IS FATFD 1-1, NAX=?
GIVF CLFSTION NO., LSE S TO FXIT, AF F TO MODIFY
RESPONSES TO LAST GLESTION CITED.
GLIFS. NO. - E
Q. 1S. ANSWFR RANGF - 1 T0 2.
NOW FECRRDFD %NGFS LQW #1# HIGH #?#
GIVF GLESTIEN NO., LISF S TO EXIT, OF F TO MODIFY
FESPONSES TO LAST CUFSTIRN CITEL.
GLFS.NO. - 14
OUFS. 14 YOUR INPL'T 2-2, AND 5 IS FATFD 1-1, MAX=?
GIVE GUFSTION NQ., USF S TO FXIT, OR P. TO MODIFY
RFSPQNSFS TE LAST GL'FSTIQN CITFD.
Gl'FS. NO. - E
Q. 14. ANSWFF FPNGE - 1 T0 ?.
NOW RECORDFD (raNSERS LOW #2# HIGH #?#
GIVF GUESTIQN NQ., USE S TE FXIT, OF F TE MOLIFY
FFSPQNSES TG LAST GLESTION CITFD.
GLES. NE. - S
GIVF CANDIDATE NC., S T0 FXIT THIS STFP, OF F TC
EXAMINF DFTAILS gF LAST CANDIDATF NAMFI.
CAND. NE. - S
S CHOICES.
```

Fig. 18. Step 7. Section V. Interaction--Continued
interactive run can be considered complete, and the interaction may be terminated.

Step 8. Termination. Refer to Figure 19. Ending the interaction is easy. You simply need to enter $A$ as a response. However, it will be more customary to enter option 1 (as one of the five choices available during interaction) to reach a normal STOP. This was done in the example and the normal STOP 30 identification was given. This is followed by an indication of how many seconds of computer central processor time was actually used during the execution of your program and the customary request

COMMAND-
At this point you could simply disconnect the terminal by hanging up (terminating the telephone linkage). However, to obtain a log of the cost of the interactive session, the proper response is
COMMAND-LOGOUT.
as illustrated. After the billing information is printed, you can then simply hang up.

Simplified Start-Up for Additional Interactive Sessions. Once an interactive run has been made, certain shortcuts can be used in making additional runs. Steps 1 and 2 must be followed as before in order to establish connection with the computer and data base. The first reply under Step 3 may be \# in lieu of YES or NO. This suppresses the detailed queries and brings you directly to the question on entry option. Here, you simply respond 3, and the system will be ready to accept the entries previously encoded and displayed in the form shown in Figure 15 on page 99. Such entries will have arisen when SECTION IV of the program was executed on an earlier run. Also, updated versions would have been produced during your earlier interaction if you chose option 5. RE-DO LIST (see page 103) during earlier interaction (Step 7).

After the entry of coded information, you are automatically moved to Step 5 to continue modification of inputs, interaction, etc, as before.

## Strategy.for Interactive Negotiations

The material beginning on page 99 described the procedures available for the interactive dialog portion of the CHOOSE procedures. This section describes some of the ways a fruitful exchange of information between you, the CHOOSER, and the computer can be developed. The five choices of interactive steps (page 102) give you a great deal of flexibility in adjusting your requirements and examining your resources to see which choice from the available candidate set will be the most appropriate.

```
CAND.ND. - S
5 CHOICES.
1. JOB COMFLETED. TERMINATE FLN.
2. PETLIFN TQ G AND A FQR ADJLSTMENTS.
3. INQUIFF AEQUT SPECIFIC CANDIDATES.
4. EXTEND LIST.
5. RE-DG LIST.
REPLY -1
    STOP 30
        1.39ム CF SECONDS FXFCUTION TIME
CONMAND- LOGELT
CF TIMF 2.726
FP TIMF 14.725
CONNECT TIME O HRS. }49 MIN
    04/18/75 LथGGFD ØUT AT 20.2&.37.<
```

Fig. 19. Step 8. Termination

Typically, an engineer who is seeking some solution to his problem (whether model choice or any other) is faced with the usual trade-offs: the desire to do the best possible job and the desire to use the minimum of resources. When he is seeking some guidance on model choice, his typical initial position in the interaction ('negotiation" with the data base) will be to state his requirements in the most extravagant terms and his resources in the most parsimonious terms; the idea being that some candidate just might come through and meet all conditions. The usual result will be that none of the candidates will be feasible under these conditions. This is where the interaction must begin.

The criteria set for any CHOOSE data base consists of a series of questions directed to the user (CHOOSER) and a numbered list of response options. The questions are of two types which, as will be described below, are treated somewhat differently in the computer analysis. The two types are:

1. Binary: The binary questions are those for which only two choices for answers are offered. These questions are considered fundamental since they usually correspond to answers that are direct opposites of each other, such as YES or NO, TRUE or FALSE. The candidate EVALUATORs who prepare the data base are not permitted to equivocate on these questions. They must select one answer or the other. (As noted below, however, the CHOOSER can give a range of answers to any of the questions.)
2. Higher Order: The higher order questions are those having more than two response options. From computer memory considerations and to prevent the candidate EVALUATORs from creating too many levels of response for CHOOSERs to work with effectively, the maximum number of responses was limited to eight. Here, the various levels of answer represent, to some extent, a measure of performance or a degree of belief in the level of performance needed. The candidates in the data base are rated on higher order questions by each being assigned a low and a high value. The low value represents a guaranteed level of performance, and the high number, the performance expected under the most favorable conditions.

During the input steps (Sections I and II of CHOOSE: pages 92 through 97) you were asked to give a low and a high response to each question although both responses could be the same. The low answer was to be based on your basic needs or your available resources--using the context of the question as the key to which of these considerations is present. The high answer was to represent the extra performance you would like to have if it was available, or the additional resources you could make available if necessary. It will be convenient to call these responses your low and high aspiration levels.

Allowing two entries for each question has two advantages: 1) it places less of a burden on the CHOOSER to be extremely precise in his judgments until a later time (during interaction) when, for one or two questions, he may have to "tighten" his response, and 2) it provides a basis for a more discriminatory rating scheme for the various candidates.

After your aspiration levels are established and verified, the selection process begins. The analysis for each candidate is based on a question by question comparison of your aspiration levels with the evaluations of the various candidates. A rating is assigned to each candidate based on the scale described below. For convenience of explanation, the ratings are given in the order of decreasing number but increasing desirability.
8. No Hope: The CHOOSER's lowest aspiration level for at least one binary question is 2 , whereas the candidate is rated 1. It can't do the job.
7. Very Bad: The candidate is capable of meeting the minimum aspiration level on all binary questions. However, even at the highest level of performance the candidate can expect to achieve, at least one higher order question fails by two or more units (response levels) to meet the CHOOSER's lower aspiration level. Substantial reduction in requirements and/or increase in resources would be needed to bring this candidate into consideration.
6. Bad: This is the same as 7 except that none of the failures are by more than one unit (or response level). A count of the number of higher order questions that are missed by one unit is given as a secondary rating criterion. This permits some discretion in selecting candidates for review since those candidates with the higher counts will usually stand less chance of eventually being selected. If no better candidates are available, some interaction here may develop one or more possibilities.
5. Marginal: A rating of 5 indicates that the candidate so rated, when viewed in the most favorable conditions, will meet the CHOOSER's minimum aspiration level. This is a marginal rating because only the best levels of candidate performance will meet the minimum CHOOSER requirements.
4. Acceptable B: This is the minimum fully acceptable level. Here, the standard or conservative candidate evaluations meet or exceed each of the CHOOSER's minimum aspiration levels.
3. Acceptable A: Here, the highest aspiration level of the CHOOSER can be met or exceeded for each question by the most favorable candidate rating. This is considered in most cases to be a stronger criterion than a rating of 4 . In some cases, however, a rating of 3 may not be considered by some CHOOSERs as significantly better than 4.
2. Fully Acceptable: Meets the conditions for ratings 3 and 4. A rating of 2 indicates a strong candidate.

1. Outstanding: This most favorable rating is for candidates whose nominal rating equals or exceeds the CHOOSER's highest aspiration levels at every point. The number of points on which the rating is exceeded is given as a secondary criterion. In this case, the higher secondary numbers represent the stronger candidates, and the program orders the list on that basis.

## The Interactive Phase

Assume now that an interactive dialog is required to see whether feasible--and, hopefully, minimal--adjustments of inputs will produce some likely candidates. Guidance in making these adjustments is required. The following strategies can be helpful. The order of their use should be determined by the user; there is nothing sacred about the order in which they appear.

1. Examining the List (Extending, if necessary): All candidates tied for the same rating might not appear on lists arbitrarily truncated at ten. Hence, unless there is a clear and useful break in candidate ratings within the first list of ten presented, it may be profitable to extend the list (Interactive Option 4) until such a break in ratings is noted. For example, if the best rating available for any candidate is 6 , then all candidates rated 6 should be displayed even though there are more than ten.
2. Checking Trouble Spots: Since your aspiration levels may be much more stringent in one question set than in another, it is useful to see where the trouble spots are. For any candidate, the difficult points can be listed by using Interactive Option 3, and then calling for the candidate reports by candidate number. The computer then displays two horizontal lists:

$$
\begin{aligned}
\text { CRITERIA NOT MET }- & (\text { list of question numbers, or } 0 \text { if } \\
& \text { all are okay.) } \\
\text { MARGINAL REPLIES - } & (\text { list of question numbers, or } 0 \text { if } \\
& \text { all are okay.) }
\end{aligned}
$$

Your aspiration levels for a particular question are considered NOT MET by the candidate if the most favorable rating for that candidate is less than your minimum aspiration level for that question. Similarly, it is considered MARGINAL if your high aspiration level cannot be met by the standard candidate rating, but your low aspiration level is met by the most favorable candidate rating.

Consider the following example of various aspiration levels for candidate 47 having a standard rating for question 26 of 4 and a most favorable rating of 7. If your aspiration levels for this question were 3-4 (low response of 3 and high response of 4), the candidate would meet your requirements on this point; at its standard performance level (4), it meets your highest aspiration level (4). If, however, your aspiration levels were 4-5, it would be considered MARGINAL for question 26 since your highest aspiration level (5) cannot be met by the standard candidate rating (4). It is not considered a failure since most favorable rating of 7 does exceed the low aspiration level of 4. In fact, any combination of aspiration levels such that the lowest is less than 7 would not be considered a failure. Any combination with the highest aspiration level equal to or less than 4 would be considered as meeting all criteria.

Note that this definition of MARGINAL is quite severe, but it is included to be sure that nothing is really overlooked. You may decide when examining the comparative ratings that, even though rated MARGINAL here, it is satisfactory. One way of demonstrating your belief in this would be to adjust your aspiration level(s) downward.

The strategy here is to examine as many of the suitably rated candidates in this manner as are needed to perceive a pattern. If it appears taht many of the candidates are in trouble on the same subset of questions, then those questions can be singled out for review of aspiration levels. Of course, any question that is listed as a trouble spot for a likely candidate should be re-analyzed.

The procedure for direct comparison of aspiration levels versus performance ratings for specific candidates and specific questions is as follows:
a. Using Interactive Option 3, enter the number of the candidate of interest.
b. When the terminal responds, asking for the next candidate number, enter $R$ to move to the question comparison level. This produces a request for the question number.
c. Enter the question number of interest. The terminal then displays your aspiration levels and the candidate ratings for that question. (See Figure 18, page 105.) The ratings are displayed in the low-high form (e.g., 3-4). This gives you an opportunity to see the nature and degree of difficulty that the candidate is in for that particular question.
d. Three options are available at this point.

1) If you wish to modify your aspiration levels for that question on the spot, you may do so simply by entering R. This brings up a format similar to that of SECTION I (Method 1) of CHOOSE for your use. After your modifications are accepted, you are brought back to the point where these three options are again open to you.
2) If you wish to see comparisons for other questions for the same candidate, you simply enter the question number as in $c$ above.
3) If you are finished with question comparisons for this candidate, you would enter S. This returns you to a above, and you can now begin to explore the possibilities of different candidates.
e. When you are finished interacting under Interactive Option 3, you simply enter $S$. This returns you to the point where all five interactive options again become available to you.

Note that trouble spots can be checked as often as desired. You may leave this interactive option and return later if it becomes desirable.
3. Returning to the $Q$ and $A$ Section: If the initial display of the most favorably rated candidates is a complete disaster, you may wish to return to the SECTION I of CHOOSE to go through the entire process of entering your criteria (aspiration levels) again. To do this, simply select Interactive Option 2. The return to SECTION V, INTERACTION, is through the steps described earlier. You can speed the process up somewhat, however, by using the $S$ option whenever you desire to skip over a section.
4. Re-Do List: During checking of trouble spots, the influence of any changes in aspiration levels for the particular candidate you are observing at that time is obvious. The influence on other candidates is not as clear. If you have made several changes in aspiration levels, it will often be to your advantage to create a new list of the top ten candidates for study. This is done by selecting Interactive Option 5. Once this is done, the list may be extended, as needed, by selecting Interactive Option 4.
5. Overchoice: If the number of candidates with a suitably high rating is excessive, the problem of overchoice arises; the set of feasible alternatives is too large. Rather than making an arbitrary choice of one from the set, some attempt should be made to make an optimal or near-optimal selection. By using Interactive Option 2, RETURN TO Q AND A FOR ADJUSTMENTS, you can review any or all of your original inputs (aspiration levels) and upgrade them in any suitable way. Then, on return to the interactive part of CHOOSE, you can expect to see the size of the feasible candidate set reduced. This technique can be reused until no further improvement in discrimination between feasible candidates is noted. The final choice, then, is the one remaining candidate, or a selected one of the final subset of candidates. Additional off-line study of the remaining candidates could aid in this choice since there may be other considerations present that were not encoded in the data base but are of some consequence to you. In other cases, the choice may be arbitrary or subjective without being infeasible.

Of course, during all of the above the candidate ratings remain fixed. You, as the CHOOSER, need to work with your aspiration levels in a suitable way in order to find out what is available to meet your requirements. Effective use of the above strategies can speed up the process and educate your discretion in choosing a satisfactory candidate for your needs.

## Bringing Programs Up for Interaction

Since permanent mass storage files have not yet been established for the programs needed for the CHOOSE system, it is necessary to execute a small batch program prior to dialing in for interaction. The following program deck will take care of this for use of PROGRAM TPDEK8.

```
L\emptysetVELLR,BNxxxxxxxx,CM20000,T5,TP1,ST0.
REQUEST (CHØ\emptysetSE,*PF)
REQUEST(TPDEK8,*PF)
LABEL(TAPE,R,L=L\emptysetVELLRE,D=HI,VSN=7058B, PW=REL,R\varnothing)
SKIPF(TAPE,3,17,C)
SKIPF (TAPE, 2,17,B)
SKIPF(TAPE,1,17,C)
C\emptysetPYBF (TAPE, CH\emptyset\emptysetSE)
CATAL\emptysetG (CH\emptyset\emptysetSE,CH\emptyset\emptysetSE, ID=REL)
SKIPF (TAPE, 1,17,B)
SKIPF (TAPE, 1, 17,C)
C\emptysetPYBF(TAPE,TPDEK8)
CATAL\emptysetG (TPDEK8,TPDEK8,ID=REL)
RETURN(TAPE)
```


## APPENDIX B

## THE EXAMPLE MAIN PROGRAM-TPDEK8

The example main program TPDEK8, which was referred to in the main body of the text, is listed in detail here. Note first that this appendix is not for interactive users (CHOOSERs); they need not read past Appendix A. For those EVALUATORs who intend to implement CHOOSE systems, this and the remaining appendices should be of interest.

The form of the main program is strictly for use with the CDC 6400 computer since the 10 -character word length is used extensively. Any modification of the CHOOSE system for use on other computers would require some changes in the CHOOSE subprogram set as well as in the main program.

PROGRAM MAKEUP can be used to prepare the basic framework for any main program. All of the cards except the individual candidate cards (cards 80010 through 89993) for the main program are prepared automatically by MAKEUP. See Appendix E for details. PROGRAM CODEM can be used to prepare individual candidate cards if desired. See Appendix F for details.

The remaining disucssion in this appendix refers to the TPDEK8 program listing by identification number in columns 76 through 80.

Card 10001 gives the program its unique 6-character name. This name is the entry point for the program, and it is necessary that the program be referred to by this name when loading the program for interactive use (see Figure 9 of Appendix A). The first three characters of this name become the identifiers for columns 73 through 75 of each card.

Card 10002 is a comment card which should be updated each time revisions are made to the program.

Card 10013 is standard in form but the dimensions will vary depending on the maximum number of candidates for which the program is currently designed. Also the dimensions of QFORM and CFORM will depend on whether or not question and answer or candidate text information is to be imbedded in detail in this main program.

Card 10016 is a DATA card which identifies the program name and maximum number of candidates to other subprograms through the COMMON block LABEL.

Cards 10017 through 10025 establish the basic data for the question and answer set and remain the same through all changes of candidates until the fundamental question and answer set is modified. Cards 10027 through 10035 establish the beginning location of any question and answer text which is included in the QFORM data beginning at card 20002.

All of these locations are initialized to a value of 1 by PROGRAM MAKEUP. As text is added beginning at card 20002, the text location markers in 10027 through 10035 should be changed.

The candidate library set begins with 30001 , and the preliminary information including a default text for candidates is contained in the subsequent cards through 30005. Card 30003 should be updated whenever any changes are made to any elements of the candidate library. Any additional description of candidates which is to be included in CFORM should be inserted beginning with card 30005 . When such information is inserted the last reference point on the corresponding data card $8 \times x x 3$ should be modified to show the appropriate text starting location. All cards from 80010 through 89993 are candidate cards in the precise form prepared by PROGRAM CODEM (see Appendix F).

The closing statements 90000 through 90003 constitute the portion of the program which calls the CHOOSE subprogram set. Card 90001 (call CHOOSE .....) is the first executable statement in the program. The next card (STPP 30) is the second and last. Return to this statement terminates execution of the program. All of the action takes place in the CHOOSE subprogram set.

There is no requirement that PROGRAM MAKEUP or PROGRAM CODEM be used in preparing the main programs such as TPDEK8. Main programs can be typed directly on cards if desired; however, the coding process for preparing the information that occurs on cards $8 x x x 2$ and $8 x x x 3$ for each candidate is quite complicated and would require a rather extensive knowledge of the packing scheme used in CHOOSE in order to prepare the information properly.



-
$\cdots$
$\cdots$
$\cdots$
$\cdots$
$\rightarrow N$
ने

$\rightarrow \underset{0}{\circ}$
$\cdots \infty_{\infty}^{\infty}$
नべ

$\cdots+\underset{1}{2}+$
नM～
$-{ }^{-1}$
を
ITH EACH CHANGE TO CANDIDATE LİBRARY：


3
0
0
0
0
0
0
0
0
0
3
3
+
110 N

$$
\boldsymbol{U}
$$

$u$

R

HNMOFNMOHNMOFNMOHNMOFNMOFNMOFNMOFNMO




CALL CHOOSE (M,KMDLSOKAND, QFORM, CFORM)
STOP 30
END

## APPENDIX C

## THE CHOOSE SUBPROGRAM SET

The CHOOSE subprogram set consists of a master Fortran program named CHOOSE, and with 10 subordinate subprograms. They constitute a Fortran based computer language for attacking the particular class of choice problems definable in terms of the CHOOSE model given in Chapter 4. The choice of Fortran as a host language was made since many people are familiar with the Fortran language and Fortran compilers are generally available at computer installations. Accordingly modifications desired for local use can be made quite easily.

Many of the concepts, if not the features of the GASP II simulation language (Pritsker and Kiviat, 1969) were used in developing the CHOOSE system. Here, as in GASP, a main program peculiar to the problem under study is prepared. See Appendix B. This program consists largely of storage allocations and certain data initialization statements. It ends with a call to SUBROUTINE CHOOSE (in the present case) or GASP (in the comparative case). The called subprogram takes charge from that point forward and calls other subprograms as program execution requires. Return to the main program, if it ever occurs, is only for stopping.

Because the class of problems that can be approached by CHOOSE is smaller than that for GASP, it has been possible to standardize and automate to a large extent the preparation of the main program to be used with the CHOOSE subprograms. The automated preparation of main programs is described in Appendices D, E, and F. Because of this feature, it is possible for researchers and investigators to develop CHOOSE main programs without learning a new computer language at all. Implementing CHOOSE can be a simple matter of focusing directly on the problem and preparing inputs in the form of data work sheets only. The auxiliary programs MAKEUP and CODEM of Appendices $E$ and $F$ take care of the rest. However, as in GASP, the more sophisticated can modify the subprograms for special purposes at their pleasure simply by applying their knowledge of Fortran.

One note of caution here, however. It was possible for Pritsker and Kiviat to write most of the GASP subprogram in machine independent Fortran. But, in CHOOSE, computer words are "packed" in many cases in order to reduce requirements for internal memory. While this makes possible in the CDC 6400, for example, the storage of 60 independent bits of information in one computer word, it does make the subprograms machine dependent to some extent.

It is not necessary at all to read or understand the remainder of this appendix to become a competent interactive user of CHOOSE (a CHOOSER), or for that matter, to become a generator of programs using CHOOSE (an EVALUATOR). A reasonably detailed knowledge of Fortran IV is presumed for understanding the remainder of this appendix.

The next three major subheadings describe elements important to the understanding of all of the operations of the CHOOSE system. Following these a description and listing of each of the subprograms is given. The complete CDC 6400 Fortran listing is provided in each case. Constant reference should be made to the Fortran listing as each of the subprograms is discussed. To insure that each discussion is reasonably independent, there will be some repetition of material.

## The NEED Array

There are two two-dimensional arrays of fundamental importance in the CHOOSE system: the NEED array and the MODEL array. It will be important to understand how these are organized and used before getting into any discussion of individual subprograms.

The NEED array is dimensioned with 72 rows and 6 columns. Each row represents the status of a particular question from the question set associated with the problem. Hence, one may speak of row number (of the NEED array) and question number interchangeably. The uses of the various columns are as follows:

Column 1 - The number of answers in the answer set for the corresponding question/row. If the question is not in use, column 1 contains a zero. This information is preset by a DATA statement in the main program.

Column 2 - An integer code number indicating whether or not the interactive user has supplied answers to the corresponding question. The codes are as follows:

3 - No part of the question has been answered.
2 - The interactive user has given a lower bound answer to the question but not an upper bound one.
1 - The interactive user has given an upper bound answer to the question but not a lower bound one.
0 - The user has given both upper and lower bound answers to the question.

Column 3 - The current recorded value of the lower bound answer. This is initialized as 1 . Hence, if the user fails to answer a question, the lower bound value remains as a default value.

Column 4 - The current recorded value of the upper bound answer. This is initialized to the value given in column 1. Hence, if the user fails to answer a question, the upper bound value remains as a default value.

Column 5 - An integer representing the number of binary shifts that are to be used when packing the answers into computer words. The values depend on whether the question has two or more answers in its answer set and on the ordinal number of the question in a totally ordered list of numbers of questions of the same type.

Column 6 - An integer representing a location in single subscripted array QFORM. This location begins the text (in variable FORMAT form) of a message associated with the particular question identified with the row number of the NEED array. A DATA statement in the main program initializes this column. If no auxiliary text is available the value of 1 is assigned. This causes a line advance but no text to be produced.

During the interactive development of input information, columns 2,3 , and 4 are constantly updated so that the current status of all questions is always available. However, the information in column 1 remains unchanged during execution and can be used to reinitialize the NEED array for a new start should the need arise. The information in column 5 is generated from column 1 during initialization or reinitialization but reamins unchanged during interaction. Column 6 also remains unchanged.

To modify the CHOOSE set of subprograms to accommodate more than 72 questions would require a change in the row dimension of the NEED array and the addition of variable names to accommodate the additional information developed during coding. Fortran statements would have to be added to manipulate the additional variables.

On the basis of the DATA for colunn 1, the NEED array is initialized (or reinitialized) by the 8th option of SUBROUTINE SPECIAL. (This option is called by statement 50 of CHOOSE through the ENTRY point SPEC8.) During initialization, the values in column 1 are tested to see if the question is in use and, if so, whether it is a binary or higher order question. If the question is nct in use, the other columns of the array are not changed. If the question is in use, columns 2 through 5 are initialized to the values described in the table above.

A double check is made during initialization to see that the number of questions of each type is within bounds ( 50 binary questions, 14 questions of higher order in the present configuration using the CDC 6400 computer). During this process the number of questions actually in use is recorded in $N Q$, and a list of the questions in use (in increasing number order) is stored in the $72-$ vector NQUES. Further, the number of questions of each type (binary and higher order) are stored in the 2-vector KQ, and the corresponding lists of the two types of questions are stored in the $2 \times 59$ array KQUES. All of this information is made available to the other subprograms through COMMON block QA.

## The MODEL Array

The MODEL array is dimensioned with 8 rows and an arbitrary number of columns. The maximum number of candidates expected to be included in the "library" or "data base" of the particular interactive project determines the selection of the number of columns. The upper bound constraint on this choice is the amount of core memory capacity that it is desirable or possible to commit to the operation of a particular interactive project. The variable IMDLS in COMMON block LABEL identifies this column dimension where needed to various subprograms.

There are two special features to note concerning the MODEL array. First, it is not carried in either blank or block COMMON. Second, the normal roles of rows and columns are interchanged. These two features permit the size of the MODEL array to be determined strictly by the main program. No recompilation of subprograms will be required when using the CHOOSE with any properly constructed main program. Further, and perhaps more importantly, any DATA declarations used to place data in the MODEL array need not be modified if the dimension of the MODEL array is changed subsequent to initial preparation of such declarations. (See Appendix $F$ for information of the preparation of such declarations.) The concept involved is similar to that of the NSET array in GASP. For additional details, refer to Pritsker and Kiviat (1969, pp. 23-24).

In a manner comparable with that of the NEED array, the column numbers of the MODEL array correspond to the candidate numbers, so it is possible to speak of candidate model or column numbers interchangeably. The uses of the various rows of the MODEL array are as follows:

Rows 1 and 2 -- The two rows permit a two-computer-word mnenomic label to be stored for each candidate. When using the CDC 6000 series computers this amounts to 20 characters.

Row 3 -- contains the one computer word packed with the candidate rating information for the binary questions. For the CDC 6400 computer, this one word can contain ratings for 59 binary attributes.

Rows 4 and 5 -- Each row contains a packed computer word containing rating information for higher order questions. Four binary locations are reserved for each question. For the CDC 6400, the ratings on 14 higher order attributes are packed into a single word. For the higher order questions, the evaluation team is permitted to assign upper and lower values to candidate ratings. The upper bound answers are contained in row 4 ; the lower bound, in row 5.

Row 6 -- Row 6 has two purposes. Initially, it is used to store information useful in initializing various vectors containing information about the MODEL array. Later, it is used to store the primary candidate ranking information determined by comparison of candidate attributes with the resource and requirements information recorded in the NEED array.

Row 7 -- Row 7 is used to store secondary information on candidate ranking based on the interactive user's inputs as recorded in the NEED array.

Row 8 -- Row 8 is an integer representing a location in the single subscripted array CFORM. This location begins the text (in variable FORMAT form) of a more detailed description of the candidate. The candidate DATA statements in the main program initialize this row. If no auxiliary text is available the value of 1 is assigned causing a line advance but no text to be produced.

Rows 1 through 5 and 8 are initialized by DATA statements in the main program and remain unchanged throughout the computer run. The values in rows 6 and 7 are changed during execution depending on the requirements placed on the various candidates by the interactive user.

## A Pattern of Statements Involving BRANCH

In approximately 20 places in the CHOOSE subprogram set a pattern of Fortran statements similar to the following appears:

PRINT f
CALL BRANCHx (i)

$$
\begin{aligned}
& \mathrm{GO} \mathrm{TO}\left(\mathrm{~s}_{1}, \mathrm{~s}_{2}, \mathrm{~s}_{3}, \mathrm{~s}_{4}, \mathrm{~s}_{5}\right), \text { IGOTO } \\
& \mathrm{s}_{1} \quad \mathrm{GOTO}\left(\mathrm{~s}_{6}, \mathrm{~s}_{7}, \ldots, \mathrm{~s}_{5+i}\right) \text { IANS }
\end{aligned}
$$

where $f$ is the label of a FORMAT statement, $x$ may be $1,2,3,4$ or blank; $i$ is an integer variable or constant; and the $s_{j}$ are labels of various executable statements. In each case the PRINT statement presents a question for which a response of YES, NO or an integer from 1 through $i$ is expected. SUBROUTINE BRANCH (with ENTRY points at BRANCH1, BRANCH2, BRANCH3, or BRANCH4) then takes over the responsibility of reading and analyzing the response supplied at the terminal by the interactive user. On the basis of the answer obtained, various actions are taken as follows:

| Answer | Action |
| :---: | :--- |
| A | Job is terminated with a STOP 27. |
| YES | IGOTO=1, IANS=1. |
| NO | IGOTO=1, IANS=2. |
| $?$ | Clarifying information is supplied, and the <br> original question is repeated. |


| \# | The logical variable FLAG is complemented, and, therefore, the decision on suppressing unwanted text is reversed. The question is repeated. |
| :---: | :---: |
| R | IGOTO=3, IANS is undefined. $R$ usually means return to an earlier part of the program. When it has other meanings, they are explained. |
| S | IGOTO=2, IANS is undefined. S usually means skip to a point further on in the program. Where it has other meanings, they are explained in the text. |
| A blank or D | IGOTO=5 if ENTRY points BRANCH1, BRANCH2, or BRANCH4 are used. Otherwise, treatment is the same as given for "other" below. |
| An integer from 1 through i | IGOTO $=1, \quad$ IANS $=$ the integer answer. |
| other | An error message is issued, and the question is repeated. |

The variables IGOTO and IANS are passed through subprograms in unlabeled COMMON. Analysis of the patterns as they appear will show that the computed GO TO using IGOTO as the controlling variable takes care of all special answers which arise during execution, while the one using IANS as the controlling variable is the one which controls normal responses to the questions. In each case, when IGOTO=1 the program branches to a statement which makes use of IANS.

The use of this pattern of statements standardizes both the preparation of the subprograms and the pattern of response during execution. It permits the interactive user to break out of the normal pattern at will, to change modes of presentation, to clarify questions, and to branch easily back and forth to various parts of the procedure. Note that 1 and YES are interchangeable as answers, as well as 2 and NO.

With these three items of general background the discussion of the subprograms begins.

## Subroutine CHOOSE

SUBROUTINE CHOOSE controls all of the operations during interaction. It is called by a main program developed specifically for a particular project. The calling program need contain only two executable statements: CALL CHOOSE . . ., and STOP. Once CHOOSE is called, return to the main program is solely for the purpose of stopping the execution.

Fortran Listing of SUBROUTINE CHOOSE:
SURROUTINE CHOOSE(MODEL.KMDLS.KAND.OFORM,CFORM)
C SECTION O. INTRODUCTORY MATFRIAL. PRELIMINARIES. C BY R. E. LOVELL 2-10-73. LAST PROGRAM REVISION 11-17-74. C LAST DECI.ARATIVE CHANGES 9-2-73.

LOGICAL FLAG.MARK.ECHO
INTEGER START(8),FIN(8),SORT(3)
DIMENSION IFMT(3).NO(4)
DIMENSION MODEL (8,1),KMDLS(1),KAND(1)
COMMON IGOTO. IANS,FLAG,DIGITI,ECHO
COMMON/GA/NQ,NQUES(72),NEED (72,6),KO(2),KQUES $(2,59)$
COMMON/LABEL/IDATE,KDATE,NAME,NMDLS, IMDLS
DATA IBIAS,IADD/10HBH7H7H7H7 -9RQDQDQDQDQ/
DATA IFMT2/1UH1H .I5.1H/

+ •IFMT(1),IFMT(3)/10H(1X.2IG .4H.I5)/
DATA SORT/1.6.8/
30 FORMAT(20HOREADYE YES OR NO *.)
C FIRST EXFCUTABLE STATEMENT.
CALL SPEC7 (.TRUE. MODEL'KMDLS:KAND,QFORV, CFORM)
$E C H O=. F A L S E$.
C RE-ENTER HERE TO START FROM SCRATCH.
50 CALL SPEC8 (.TRUE. MODELMKMDLS,KAND,QFORY,CFORM)
51 FLAG=.TRUE.
PRINT 54, NAME.IDATE,KDATE
54 FORMAT(40H1UNIVERSITY OF ARIZONA. TUCSON: ARIZONA
+ 5H85721/30H INTERACTIVE CHOICE PROGRAY - AG/3X.
+ 35HCRITERIA(Q AND A SET) LAST UPDATED •R9, /3X,
+ 35HCANDIDATE INFORMATION LAST UPDATED - RO)
PRINT 30 \$ CALL BRANCH4(3)
GO Tn (55.100,50,52,70), IGOTO
40 CALL SPECIAL(.FALSE., MODEL,KMDLS.KAND,QFORM,CFORM)
GO Tn 51
$55 \quad$ GO TO( 70.56 .40$)$ IANS
56 PRINT 57 玉 PRINT 30 \$ CALL BRANCH4 (3)
57 FORMAT(*OHAVE COURAGE -- YOU WILL BE ABLE TO ABORT*
+     * WTTH A ANY TIME A*/* QUFSTION IS ASKED. NOW. *
+ *ONF MORE CHANCE.*)
GO TO(60.100.50.55.70). IGOTO
GO Tn( 70.62.40), TAiNS
PRINT 61 E STOP 6i
FORMAT(*OWE ARE ALL DISAPOOInTED;*)
$\operatorname{IF}(F \mid A G) 71: 100$
PRINT 72
FORMAT(//* RUESTIONS ARE ANSWERI:D SY GIVING A NUMER* + *iCal Response*/* in thf Range penmittedr by typing* * /* a (TO ABORT THE ENTIR PROCESS):*

```
    + /* S (TO SKIP AHEAD TO T..E NEXT MAJOR SECTION).*
    + /* R (TO RETURN TO AN EARLIER MAJOR SECTION FOR*
    + * RFVIEW), OR*/* [ (IF YOU DON'T KNOW HOW TO RE*
    * *SPOND).* /* YOUR RESPONSES ARE ENTERED IN THE *
#e:こ=-+.*PLICE PROVIDED AFTER*/* THE MESSAGE \GIVE VALID RE*
    + *SPONSE - \. EACH RESPONSE MUST*/* RE FOLLONED BY *
    + *STRIKING THE \RETURN\ KEY ro TRANSMIT YOUR */* RES*
    + *PONSE TO THE COUPUTER.*)
.73 PRINT 30 क CALL BRANCH4(3)
    GO TO (80,100.50,73.100).IGO O
80 .GO Tn(100,56.40),IANS
C SECTION T. QUESTION AND ANSNER GECTION.
C BY R. E. LOVELL 2-10-73. LAST REVISED 9-7-73.
100 PRINT 101 $ IF(.NOT.FLAG)GO TO 113
101 FORMAT(*-SECTION I. MODEL CHOICE PROGRAM BEGINS.*)
110 PRINT 111 $ CALL RRANCH(2)
111 FORMAT(*ODO YOU WANT THE COMPLETE TEXT WITH *
    + *EACH QUESTIONL*/* NOTE - YOU MAY REVERSE THIS *
    + *DERISIOV ANY TIME LATER*/* BY TYPING % IN *
    + 23HANSWER TO ANY QUESTION.)
        GO Tn(115,200.51,110),IGOTO
115 IF(I\DeltaNS.EQ.2)FLAG=.F.
113 PRINT 118
118 FORMAT(31HOTHERE ARE 3 WAYS TO ENTER YOUR
    + * CHOICE*/* CRITERIA. SELECT ONE.*)
        IF(F|AG)PRINT 114 $ CALL BRA CH(3)
114 FORMAT(* 1 = EACH QUESTION INDIVUALLY PRESENTEO.*/
    * * }>= PREVIOUSLY PREPARED iNWERS TO BE ENTERED.*/'
    * * 3 = PRECODEO NUVimERS ARE TO BE ENTERED.*)
        GO Tn(112,200.51,113),IGOTO
112.GO TO(120.130,180),IANS
C NORK WIITH EACH QUESTION INDIVIDUU LLY.
120 CALL QUEST(QFOR.H)
121 GO Tn(200,200,113),IGOTO
C ENTRY FROM PREPARED LIST.
130 CALL TABLE $ GO TO 121
C ENTRY OF PRECODFD NUMRERS.
180 CALL CODEIN(NO)
C SECTION II. REVIEW DF QUESTIONS AND RESPONSES.
C BY R. E. LOVELL 2-19-73. LAST RE ISED 9-2-73.
200 PRINT 201
201 FORMAT(*-SECTION II. REVIEW OF Q. ANN A.*)
        IF(F|AG)PRINT 202
202 FORM\triangleT(/5X**HERE YOU MAY ASK FOR ANY OUESTION TO BE *
    t *PRFSENTED AGAIN.*/* TO LEA E THIS SFCTION YOU MUST*
    + * TYPE S AS YOUR RESPONSE */* OTHERWISE GIVE THE*
    + * DFSIRED QUESTION NUVBER A YOUR*/* RESPONSE.*)
203 CALL RRANCH(72) $ GO TO(204.300.100.200).IGOTO
204 I=JANS
    CALL OUERY(I,QFORV)
    GO TO(203.3nO.I13),IGOTO
```

C SECTION TII. SUMMARY OF ANS:NERS
C BY R. E. LOVELL 2-19-73. LAST REVISED 8-17-73.
300 PRINT 301
301 FORMAT(*-SECTION TII. SUMMARY OF ANSWERS.*) IF(FIAG)PRINT 302
302 FORMAT(/5X.*CHOOSE METHON OF PRESENTATION OF RESUL'* + *TS.*/ * $1=$ ALL ANSWERS, $2=$ NONE, 3 = UNANSWER* + *ED NUESTIONS ONLY.*) CALL RRANCH(3) I VARK=.T. GO $\operatorname{Tn}(304,400,200,30 \mathrm{n})$, IGOTO
304 GO Tn(306.400,307),IANS
306 MARK=.F.
307 IF(FIAG)PRINT308 5 PRINT 309
308 FORMAT $* O T H O S E$ ANSWERS MARKE BY AN *iH** ARE ANSWERS* + * SIIPPLIED by THE*/* SYStem Since you have not yet * + *ANGWERED THEM.*)
309 FORMAT(*O NO. LOW HIGH MAX.*) DO $320 K=1$, NQ $\$ \mathrm{I}=\operatorname{NQUES}(\mathrm{K})$ $J 4=\operatorname{NFED}(1,2)+1$ GO $\operatorname{Tn}(310,311,312,313), \mathrm{J} 4$
310 IF (MARK)GO TO 320 \$ IFMT(2) IFMTZ \$ GO TO 319
311 IFMT(2)=IFMT2-600NOOOOOODNOO B \$ GO TO 319
312 IFMT(2)=IFMT2-63 \$ GO TO 319
313 IFMT(2)=IFMT2-6000000000000063
319 PRINT IFMT,I,NEED(I,3),NEED(14),NEED(I,1)
320 CONTINUE
324 PRINT 325
CALL BRANCH3(2)
325 FORMAT(40HOOO YOU WISH TO MO IFY ANY RESPONSES[ - ) GO Tn(326.400.200,324), IGOTO
326 IF (IANS.EQ.1)GO TO 200
C SECTION TV. ENCODING THE ANSWER -
C BY R. E. LOVELL 2-19-73. LAST R VISED 6-14-73.
400 PRINT 401
401 FORMAT(*-SECTION IV. ENCODI G THE ANSWERS.*) CALL RECODE(IND)
C SECTION $V$. INTERACTIVE SEARCH PROCEDURE.
C BY R. E. LOVELL 2-19-73. LAST RE ISED 11-17-74. 500 PRINT 501
501 FORMAT(*-5ECTION V. INTERACTIVE SEARCH PROCEDURE**)
502 PRINT 30 末 CALL BRANCH3(?) GO Tn(504.950.300.502).IGOTO
504 IF(IANS.EQ. 2) GO TO 300
C RATE THE VARIOUS CANDIDDATES.
510 DO $520 \mathrm{~K}=1$, NMDLS
$\mathrm{I}=\mathrm{KMnL}(\mathrm{S}(\mathrm{K})$
MODEI (7.I) $=0$
C TRY EASY RINARY TEST. J2L2=.NDT.MODEL(3.I).AND.NDi )
IF(JンL?.EQ.0)GO TO 505
C ASSIGN RATING OF is AND COUNT NO. OF FAILURES.

```
    MODEI (6,I)=8
    KOUNT=0
    JSTOP=KQ(1)
    DO 5n3 J=1.JSTOP
    IF((.12L2.AND.1).EN.0)GO TO 5 3
    KOUNT=KOUNT+i
503 J2L2=SHIFT(J2L2,-1)
    MODEI (7,I)=KOUNT
    GO T\cap 520
C SET UP RFMAINING EVALUATION CRITERIA.
505 J2H2=.NOT.MODEL(3.I).AND.ND(1)
    NX=M\cap\)EL(4,I).OR.TBIAS
    J8HBI =NX-NJ(4)
    J8H8H=NX-VJ(2)
    NX=MODEL(5,I).OR.IBIAS
    J8L8H=NX-VD(2)
    J8L8I =NX-ND(4)
C TRY EASY HIGHER ORDER TEST.
    ITEST=.NUT J8HBL.AND.IBIAS
    IF(ITEST.EQ.O)GO TO 508
C ASSIGN RATING OF }7\mathrm{ OR }6\mathrm{ AND EVALUATE FAILURES.
    ITEMЮ=(.NOT.IBIAS.AND.JBHBL) IADD
    ITEST=ITEMP.OR.J&HBL
    ITEST=.NOT.ITEST.AND.IBIAS
    IF(ITEST.EQ.0)GO TO 506
    MODEI (GOI)=7
    GO T\cap 520
505 MOOEI (G:I)=6
    KOUNT=0
    JSTOP=KQ(2)
    ITEMP=.NOT.JBHBL.AND.ITEMP
    DO 5n7 J=1.JSTOP
    IF((TTEMP.AND.8).EQ.0)GO TO 07
    KOUNT=KOUNT+1
507 ITEMP=SHIFT(ITEMP,-4)
    MODEI (7,I) =KOUNT
    GO TO 520
C TRY MORE DIFFICULT TESTS.
508 MODEI (ǴI)=5
    ITEST=.NOT.J&LRBL.AND.IBIAS
    IF(ITEST.VE.O)GO TO 509
    MODEI (G,I)=4
509 ITEST=.NOT.JBHBH.AND.IBIAS
    IF(ITFST.VE.0.OR.J2H2.NE.0)G TO 520
    MODEI (G,I)=MODEL (G,I)-2
    ITEST=.NOT.JतL&H.AND.IBIAS
    IF(ITEST.VE.0)GO TO 52.0
    MODEI (G,I)=1
    KOUNT=0
    JSTOP=KQ(2)
    DO 51F J=1.JSTOP
```

```
    IF((.18L8H.AND.7).EQ.0)GO ro 15
    KOUNT=KOUVT+1
515 J8L8H=SHIFT(J8L8H, -4)
    MODEI (7,I)=KOUNT
520 CONTTNUE
C MAKE ORDFRED ROSTER OF CANDIDATE.
    KOUNT=0
    DO 540 J=1.8
    START(J)=KOUNT+1
    DO 5.30 K=1,NMDLS
    I=KMnLS(K)
    IF(J.NE.MODEL(G.I))GO TO 530
    KOUNT=KOUNT+1
    KAND(KOUNT)=I
530 CONTTNUE
540 FIN(,1)=KOUNT
C MAKE SECONDARY SORT ON MODEL(T,I) WHERE IT EXISTS.
    INDEX=1
    DO 5&0 Jx=1,3
    J=SORT(JX)
    IGO1=START(J)
    ISTOP2=FIV(J)
    ISTOP1=1STOP2-1
    IF(IGTOP1.LT.IGOI)GO TO 556
    DO 550 K=IGO1.ISTOP1
    I=KAND(K)
    ITEST=MODEL(7,I)
    KP1=k+1
    DO 550 L=KP1,ISTOP2
    M=KAND(L)
    IF((TTEST-MODEL(7,M))*INNEX.&E.0)GO TO 550
    KAND(K)=M
    KAND(L)=I
    ITEST=MODEL(7.M)
    I=M
    550 CONTTNUE
    556 INDEX=-1
    560 CONTTNUE
    C LIST SOMF RESULTS.
    561 L=1
        LIST=MINO(FIN(2)+10,FIN(7))
        PRINT }56
565 FORMAT(42HOLIST IN APPARENT ECREASING DESIRABILITY./)
568 DO 570 K=L.LIST
            I=KANO(K)
570 PRINT 571,I,MODEL(G,I),MODEL,1,I),MODEL(2,I)
    +.MONEL(7.I)
571 FORMAT(2I4,3X,2A1D,\5)
C INTERACTIVE CONTROL.
576 PRINT 573
573 FORMAT(11H05 CHOICES.)
```

```
    IF(FIAG)PRINT 574
    G0 rn 579
5 9 1
    IANS=MANS
    CALL DISPQ(ND,IBIAS,VODEL,KMDLS,KAND,OFORM,CFORM)
    GO Tn 579
Close out ctatevents.
950 PRINT }95
    CALL RRANCH3(3)
952 FORMAT(*OENO COMING UP. MAK CHOICE -*/* 1=STOP. ₹*
    + *=Gn.PART WAY BACK. 3=START OVER. CHOOSE - *)
        GO Tn(953,950,500,950),IGOTO
953 GO Tn(954,300.50),IANS
954 RETUFN
    END
```

A detailed listing of CHOOSE begins on the following page. Reference to it during the following discussion may be helpful.

Subprogram CHOOSE is separated into 6 major sections by comment cards. Also, for each reference, the Fortran statement numbers in each section uniformly use the section number as the "hundreds" digit. Hence, all numbered statements in section III are numbered in the range 300-399. All statements in section 0 (zero) are numbered between 1 and 99. A few closing statements are numbered in the 900 series.

Section 0 of CHOOSE contains all type, DIMENSION, COMMON, and DATA statements, and several initialization statements. At statement 30+2 the first executable statement, CALL SPEC7 (an ENTRY in SUBROUTINE SPECIAL), appears. It causes certain initialization work to be done using the MODEL array. In particular, the number of candidates, NMDLS, is established, and a list of active candidate numbers is prepared and stored in the IMDLS-vector KMDLS.

The logical variable ECHO controls the reprinting of information read by the program. Since CHOOSE is designed for normal use in the interactive mode, (the input information being displayed directly at the terminal), the normal setting for ECHO is FALSE. When operating. in batch mode, it is desirable to have each input printed out. This is accomplished by entering 3, 9 and 1 as the first three data entries. This causes a branch to the 9th set of statements in SUBROUTINE SPECIAL and a return to CHOOSE with ECHO changed to TRUE.

Statement 50 calls SPEC8, another ENTRY in SUBROUTINE SPECIAL, to cause initialization of the NEED array and the associated variables described earlier.

The balance of the statements in Section 0 are short interactive statements to establish readiness for operation and to give instructions, if needed, to the first time user.

Section I begins the question and answer section by asking in which of three formats the user wishes to operate. The options are:

1. Choice 1 causes a branch to SUBROUTINE QUEST. The option should normally be selected by inexperienced people or by those who do not have reference material at hand concerning the content of the various question and answer sets. Each question number is presented together with the range of answers permitted. If the main program has been prepared with the full text, then questions can be presented in full text form, and the texts of the corresponding answer set can be listed.
2. Choice 2 causes a branch to SUBROUTINE TABLE. If the interactive user has been able to do some "homework" in the study of questions and alternative answers, he will be in a pcsition to enter his criteria directly in a tabular form. This option presents question numbers in a series of tables and provides a convenient format for entry of answers to 14 questions at a time. This is the recommended
choice for most cases, since it requires some thought be given to responses before sitting down at the terminal. Those who try to learn the entire process and generate all of their responses while sitting at the terminal will probably find the entire process too frustrating and costly for regular use.
3. Choìce 3 causes a branch to SUBROUTINE CODEIN. Here criteria can be entered directly in coded (packed word) form. To do this requires either an excessive amount of advance preparation and an uncommon ability and interest in binary arithmetic, or access to precoded information. Precoded information will most easily be obtained from printouts from earlier runs. As will be seen later Section IV of CHOOSE prints out the coded results so that they may be used as starting entries of subsequent runs if desired.

A brief description of the content of each of these three subprograms is given with the corresponding Fortran listing towards the end of this appendix. A normal return from each of these input options leads to Section II of CHOOSE.

Section II gives the interactive user a chance to review his response to any question and to modify it if he desires. Specific numerical requests for review of particular questions are routed to SUBROUTINE QUERY. When review is completed, the interactive user enters $S$ (for skip) to cause a normal branch to Section III.

Section III offers the user a chance to have his responses represented. He is given three options: (1) to receive an annotated list of all responses recorded (or standing in the NEED array by default), (2) to receive a list of only those questions for which responses were not given, and (3) to skip the summary. The lists are prepared sc that the default answers are annotated with an asterisk. The Fortran statements in this section are fairly straightforward except for those (310-312) involving character manipulation to modify variable FORMATs to incorporate the asterisks where needed.

Section IV offers no options to the user but simply packs the information standing in columns 3 and 4 of the NEED array into 4 words of the vector ND. Results are printed at the terminal. All of this is done by reference to the ENTRY point RECODE of SUBROUTINE CODEIN.

Section V is the section in which comparisons are made between needs and capabilities (NEED vs. MODEL). It also makes evaluations which highlight those candidates which meet all requirements and establish rating information useful in moving towards feasibility for those candidates which fail in some fashion.

The statements from 500 through 520 constitute the section where evaluations are performed in accordance with the criteria established in Appendix A, page 106. Each candidate is reviewed and compared with the criteria established by the CHOOSER in sections I and II. Once this is complete, the following statements through statement 560 perform the ranking required for the various models.

Statement 561 begins printout of the top 10 candidates and then turns the program back to the CHOOSER at statement 576 for interactive control.

Here the CHOOSER is presented with five choices:

1. JOB COMPLETED. TERMINATE RUN.

This option returns the CHOOSER to the main program for normal stop with a message of STOP 30. This option would be selected as the initial choice only if a clear cut decision was reached on the basis of the initial list furnished. On subsequent interaction return to statement 576 does occur and option 1 is again offered for choice.
2. RETURN TO Q AND A FOR ADJUSTMENTS.

If the initial list proves so disastrous that the input set requires a complete review, this option would be selected. It returns the control to section I. Although the array of answers previously established are not erased, the CHOOSER can elect any of the original three entry options to review and readjust his earlier aspirations. Return to section $V$ after this choice is through the normal steps of sections II, III and IV.
3. INQUIRE ABOUT SPECIFIC CANDIDATES.

Choice of option 3 causes a branch to statement 581 so that the CHOOSER can examine specific candidates with respect to the criteria he has established. With its calls to subprogram DISPL it allows for the detailed interaction described in Appendix A beginning on page 99. Return from this option (by entering an $S$ at the keyboard when CAND.NO. is called for) is back to statement 576 so that the five choices are again displayed.
4. EXTEND LIST.

By this option the CHOOSER can cause the list of top candidates to be extended by 10. Return from this option again is to statement 576 where the five choices are again available.
5. RE-DO LIST.

Since option 3, on some of its calls to DISPL or its alternate entry point DISPQ, has modes which permit adjustment of input criteria it will sometimes be useful to see how the adjustment of the input criteria changed the relative positions of the candidates other than those immediately in question. Option 5 gives this opportunity by causing the return to statement 510 so that each of the candidates is compared with the revised aspirations of the CHOOSER.

## Fortran Listing of SUBROUTINE BRANCH：

SUBROUTINE BRANCH（MAXANS）
C BY R．E．LOVELL 2－4－73．LAST PR GRAM REVISION 12－19－74．
C．LAST DECI ARATIVE CHANGES 7－13－73
C TO ELICIT AND EVALUATE A RESPDNS $\quad$ TO A GENERAL QUESTION．
LOGICAL DIGITI．BLANK，FLAG．ECHO
COMMON IGOTO．IANS，FLAG，DIGIT1，ECHO
BLANK＝．F．
1 PRINT 50
5 READ 51．KOK1 \＄MAXANT＝MAXANS
7 DIGIT1＝K1．EQ．1R \＆IF（ECHO）P INT 53．K．K1
IF（K．NE．1R ）GO TO 2
DIGIT1ニ．T．玉 K＝K1
CALL CHAR（K，9，MAXANT，BLANK）
GO Tn（3．6．6．8．6），IGOTO
IF（FiAG） $6 \cdot 1$
IF（DTGITI）GO TO 4
IF（K1．LT．1R0．0．K1．GT．1R9）GO． 010
IANS $=K 1-33 B+I A N S+I A N S+$ SHIFT IANS．3）
IF（IANS．LT．I．OR．IANS．GT．MAXANS）GO TO 10
6 RETURN
10 PRINT 52
GO TO 5
ENTRY BRANCH1
BLANK＝．T．\＄GO TO 1
ENTRY BRANCH2
BLANK＝．T．\＄GO TO 5
ENTRY RRANCH3
BLANK＝．F．S GO TO 5
ENTRY BRANCH 4 READ 51，K，KI
BLANK＝．T．末 MAXANT＝MAXANS－1 \＄GO TO 7
C I／O FORMATS．
50 FORMAT（＊REPLY－＊）
51 FORWAT（2R1）
52 FORMAT（22H INVALIn，TRY AGAI－）
53 FORMAT（＊＋＊，57X，2R1，22X＊＊AS R AD IN．＊）
END

The closing statements beginning with statement 950 are simply for warning the CHOOSER that he must make appropriate choices, or for causing return to main program for termination.

## Supporting Subprograms

The 10 subordinate or supporting subprograms used in connection with the controlling subprogram CHOOSE are described briefly and listed in detail, each under its own subheading.

Subprogram BRANCH
Subprogram BRANCH is a utility subroutine subprogram for reading and examining interactive user's response to any question posed by the program. It detects any errors in response and asks for corrections when needed. For valid responses it controls branching within other parts of the CHOOSE program. For invalid responses it solicits corrected answers.

BRANCH is called from many places in the CHOOSE subprogram set, either through its own name or one of its four alternate entry points BRANCH1, BRANCH2, BRANCH3, and BRANCH4. Its only subprogram reference is to CHAR.

Subprogram CHAR
Subprogram CHAR is a SUBROUTINE for analyzing response characters entered by the user at interactive terminals.

It contains its own error messages which are used when illegal characters are encountered, but always returns to the calling program for solicitation of corrections from the user.

CHAR is called by both BRANCH and QUERY. It does not call other subprograms.

Subprogram INEED
INEED is a FUNCTION subprogram used to encode and pack data in suitable format. Since it packs information into 10 -character words it is only useful with a CDC 6400 10-character word system.

It is referenced by SUBROUTINE CODEIN both through its normal entry point and through ENTRY RECODE. It does not call other CHOOSE subprograms.

Fortran Listing of SUBROUTINE CHAR:
SUBROUTINE CHAR(K,MAXANS, MX, LANK)
C BY R. E. LOVELL 2-11-73. LAST PR GRAM REVISION 7-13-73.
C. LAST DECI ARATIVE CHANGES 7-1.3-73

C DETAILED ANALYSIS OF RESPONSE CH RACTERS.
LOGIrAL BLANK.FLAGDDIGITI
COMMON IGOTO.IANS.FLAG,DIGIT ,ECHO
IGOT $n=1$
IF(K.GE.1R1.A.K.LE. (MAXANS $+3 \times B)$ ) 1.2
IANG=K-33B \$ RETURN
$\begin{array}{ll}1 & \text { IANS }=K-33 B \text { S RET } \\ 2 & \text { IF (K.EQ. } 1 R Y) 3.4\end{array}$
3 IANS $=1$
15 DIGITI=.T. s RETURN
4 IF(K.EQ.IRN)5,
5 IANS=2 5 GO TO 15
6 IF( $K \cdot E Q \cdot 1 R$. O.K.FQ. 1RD).A.B, ANK) 7, B
7 IGOTO=5 \$ RETURN
8 IF(K.EG.IRS) 9,10
9 IGOTO=2 E RETURN
10 IF(K.EQ.IRR)11.12
11 IGOTO=3 \$ RETURN
12 IGOTn=4 s IF(K.NE.1Rx)GO TO 22
FLAG=.N.FLAG 5 IF(FLAG)GO TO 20
PRINT 19 \$ RETURN
PRINT 21 \$ RETURN
IF(K.EQ.1RE)23.27
PRINT 24,MX $\operatorname{si}$ IF(BLANK)PRINT 26
IF(.N.FLAG)PRINT21 G FLAG = .T. $\$$ RETURN
IF (K.EQ.1RA)STOP 27
PRINT 25
IF(F|AG)PRINT 70 玉 RETURN
C I/O FORMATS.
19 FORMAT(* DETAILET TEXTS WILL BE OMITTED.*)
21 FORजAT(* DETAILEO TEXTS WIL BE PRINTED.*)


+ *: THRCUGH*I3,*, OR -*/* " (YES=1) N(NO=2)*
+     * R(REVIEN) S(SKIP) [ UNCERTAIN)*/* A*
+ *(ARORT) *(CHAVGE PRINTIN INSTRUCTION).*)
26 FORMAT(* ALSO IN THIS REGION VALID RESPONSES *
+ *include a blank*/* (SPaCe bar). or a d(DEfer).*
+     * these prfservf the most qECENT*/* ANSWER*
+     * gtven, or, in default, Th original answer *
+ *SUPPLIED*/* BY THE PROGRAM.*)
25 FORMI:T (* INVALID RESPONSF.*)
70 FORMat (* Illegal character 0 integer too large.*) END

Fortran Listing of FUNCTION INEED:
FUNCTION INEED(ID.ICOL)
C BY R. E. LOVELL 2-14-73. LAST PROGRAM REVISION 7-13-73. C LAST DECI ARATIVE CHANGES 9-2-73.
C TO ENCODF AN ARRAY OF ANSWERS INTO A SIMPLE VARIABLE. C ID=1 FOR LOW(BINARY) VALUES. ID= FOR HIGH(OCTAL) VALUES. C ICOL = COLUMN OF NEED ARRAY TO Be ENCODED.

COMMON/QA/NN,NQUES(72),NEED(2,6),KQ(?)PKQUES(2.59)
INEEN $=0$ \$ KSHIFT=ID+ID+ID-2 \$ ISTOP=KQ(ID)
DO in $N=1, I S T O P$ i $I=K Q U E S(I D N)$
INEEN=INEED $+\operatorname{SHIFT}($ NEEO $(I, I C O L)-1,(N-1) * K S H I F T)$
10 CONTINUE क RETURN
END

```
Fortran Listing of SUBROUTINE QUEST:
    SUBRNUTINE NUEST(NFORM)
C BY R. E. LOVELL 2-10-73. LAST PR GRAM REVISION 10-10-73.
C LAST DECI ARATIVE CHANGES 9-2-73.
C TO POSE ALL QUESTIONS AND ELICIT ANSWERS.
        LOGITAL FLAG
        DIMENSION QFORM(1)
        COMMON IGOTO, IANS.FLAG,DIGIT EECHO
        COMMON/QA/NQ,NQUES(72),NEED (2.6),KQ(2):KQUES(2.59)
        IF(FIAG)PRINT 5
        PRINT 8 S CALL BZANCH3(2) \(\$ 0\) TO(6.26.26.4),IGOTO
        IF(IANS.EQ.1)GO TO 1
        IGOTN=3 \$ RETURN
        \(K=1\)
        I=NQUES(K)
        CALL QUERT(I,QFORU)
        GO Tn(3.11.10), IGOTO
        \(K=K-11\) \$ IF \((K) 26,3,3\)
\(11 \quad K=K+a\)
\(3 K=K+1\) s IF(K.LE.NO)GO TO ?
26 RETURN
C I/O FORMATS.
        FORMAT'/ \(5 \mathrm{X}, * E A C H\) QUESTION I USE WILL' BE CALLED IN *
    + *TURN. THE USUAL*/* NUMERIrAL AND SYMBOLIC RESPON*
    + *SEG ARE TO BE USED.*)
8 FORMAT(10HOREADYE - )
        END
```

Fortran Listing of SUBROJTINE QUERY:

SUBROUTINE NUERY(I, QFORM)
C BY R. E. LOVELL , 2-10-73. LAST PR GRAM REVISION 12-17-74.
C LAST DECI ARATIVE CHANGES 9-2-73.
C TO ASK AND REGUEST EVALUATION OF A SPECIFIC QUESTION. LOGICAL FLAG.ECHO DIMENSION QFORM(1)
CONMON IGOTO, IANS,FLAG,DIGIT EECHO
COMMON/QA/NQ, NQUES(72), NEED $(72,6), \operatorname{KQ}(2), \operatorname{KQUES}(2,59)$
JI=NFED (I, 1)
IF(ل) 101.101.104
101 PRINT 102.I $\$$ RETURN
102 FORMAT(*UQUESTION*I3* IS NOT IN USE.*)
104 PRINT 1.I.J1
1 FORMAT(*OQ**I3*. ANSNER RA GE - 1 TO*I2***)
IF (FI AG)CALL QFILE (I,QFORU,Q ORM)
2 J5=NFFD(I,3)+333
J6=NFED $(I \cdot 4)+33 B$
J4 =NFED (I, 2) + 1B
GO Tn(10,3,4,5),J4
J5=55B \$ GO TO 10
$\begin{array}{ll}3 & J 5=55 B \\ 4 & J 6=55 B\end{array}$
10 PRINT 11. J5.J6
11 FORMAT (* NOW RECORDED LOW $\backslash * R 1 * * \backslash$ HIGH \*

- +R1** *)

5. PRINT 6

6 FORMAT(* RECORD YOUR ANSWERS* $6 \times * \backslash \backslash * 8 X * \backslash * / 1 H+, 21 X)$
READ 12,K•L $\ddagger$ IF (ECHO)PRINT . 3OKOL
12 FORMAT(R1.10X,R1)
CALL CHAR(K,J1,J1..TRUE.)
GO TO(20.103.103.5.25).IGOTO
20 NEED $(5,2)=N E E D(1,2) . A . \operatorname{N.1B}$
NEED (I, 3) =IANS
IF(IANS.EN. ل1)NEED (I, 2) =NEED I, 2).A...N.2B
25 CALI CHAR(L.J1.JI..TRUE.)
GO TO(30.103.103.2,35).IGOTO
30 NEEO (I•2) =NEED (I.2).A..N.2B
NEED(I;4) IIANS
IF (IaNS.EQ.1)NEED(I.2)=NEED(,2).A..N.1B
35 IGOTn=1
103 IF(NFED(I,4).GE.NEED(I,3))RE URN
PRINT 40 § GO TO 2
40 FORMAT (* YOUR LOW ANSWER EX EEDS THE HIGH ONE: *)
ENTRY QUERT
JI=NFFD(I.1) \$ GO TO 104
13 FORMAT(*+*,21X.R1.10X:R1.42X *AS READ IN.*)
END

Fortran Listing of SUBROUTINE TABLE：
subrnutine táble
C BY R．E．LOVELL 6－3－73．LAST PR GRAM REVISION 11－17－74．
C LAST DECLARATIVE CHANGES 9－2－73．
C to Provine tables for direct entry of answers．
LOGICAL FLAG．ECHO
DIMENSION NQLOC（14）．NMAX（14）LL（28），LH（14）．LANS（42）
$+\quad$ MF（4）M（2）
COMMON IGOTD，IANS，FLAG，DIGIT ，ECHO

DATA M（1）／10H（＊LOW AN／．M（2）／10H（＊HIGH AN／

+ ．KF2／10HS．＊．7X．00／
+ ，KF（3），KF（4）／10H（1X，I1，R1），9H．＊OK［＊）／
NT＝（N（N－1）／14＋1
PRINT 1。NT \＆IF（FLAG）PRINT 2
ITBL $=1$
C WORK WITH SECCTION OF TABLE IDENTIFIED BY ITBL．
99 KSTART＝14＊（ITBL－1）\＄MX＝MINO（NQ－KSTART，14） $M X 2=M X+M X$ \＄$M X 3=M X 2+M X$
DO 5 IM＝1，MX
$C K=$ OROINAL NO．OF QUFSTION．I＝QUESTION NUMBER．
$K=K S T A R T+I M E$ I $=$ NQUES K）
$\operatorname{NQLOR}(I M)=I$ \＆ $\operatorname{NMAX}(I M)=\operatorname{NEED}(, 1)$
5 CONTTNUE
PRINT 6．（NQLOC（IM）！IM＝1．MX）
PRINT 7，（NMAX（IM），IM＝1，MX）
C begin regponse area．J＝1 For low values．j＝2 FOR high． $3 \quad \mathrm{~J}=1$
8 N＝Jt？\＄JBAR＝3－J \＆NBAR＝JBAR 2 ゅ IF（J．EQ．2）PRINT 63
DO 9 IM＝1，MX
$9 \quad L H(I M)=0$
10 DO 15 INi＝1；MX $\$ K=K S T A R T+T M \$ I=N Q U E S(K)$
$\operatorname{LL}(2 * I M-1)=N E E D(I, N) \$ L L(2 * M)=1 R$
IF（NFFD（I，2）．GE．3．0．NEED（I，2 •EQ．J）LL（2＊IM）＝1R＊
IF（LH（INi）．NE．0）LL（2＊IM）＝LL（2＊IM）－508
15 CONTTINUE
KF（1）＝M（J）\＆IF（MX．LE．9）GD TO 16
MXTEN $=$ MX／1．0 $\ddagger$ ；MXONE $=M X-, 0 * M X T E N$
KF（2）$=$ KF $2+\operatorname{SHIFT}(W X T E N, 6)+M X O N E ~ \$ ~ G O ~ T O ~ 20 ~$
$16 \mathrm{KF}(2)=K F 2+22003+M X$
20 PRINT KF．（LL（IM2），IM2＝1，MX2）
CALL RRANCH2（3）
21 GO Tn（30，61，62，23．29），IGOTO
22 RETURN
23 CALL RRANCHI（3）玉 GO TO 21
24 ITRL $=1$ TBL +1 I IF（ITGL．LE NT） 99.2 ．

```
26 ITBL = ITBL - 1 末 IF(ITBL.LE 0)22.99
29 IANS = 1
30 GO TO(60.40,35),IANS
35 PRINT 36 क CALL SRANCHZ(NT)
    GO Tn(37,24,26,35),IGOTO
37 ITBL=IANS $ GO ro 99
40 PRINT 41 क READ 42.LANS $ IF ECHOJPRINT 64,LANS
    DO 49 IM=1,MX
    K=KSTART+IM $ I =NQUES(K) $ NUM=0 $ IBLANK=0
    DO 4.3 I3=1.3
    IC=3*IM-3+I3
    IF(LANS(IC).EQ.1R )IRLANK=IB ANK+1
    IF(LANS(IC).LT.1RI.D.LANS(IC.GT.NMAX(IM)+33B)GO TO 43
    INUM=INUM+1 क ILOC=IC
43 CONTINUE
    IF(INUM.EQ.1.A.IRLANK.EQ.2)G TO 47
    IF(IRLANK.EQ.3)GO TO 49 $LH(IM)=1 $ GO TO 49
47 NEEO(I,N)=LANS(ILOC)-33B & NEED(I,2)=NEED(I,2).A..N.J
    ITEST=1 & IF(J.EQ.1)ITEST=NM X(IM) क LH(IM)=0
    IF(NFED(I,3).GT.NEED(I,4))NE D(I,NBAR)=NEED(I,N)
    IF(NFED(I,N).EQ.ITEST)NEED(I 2)=NEED(I,2).A..N.JAAR
49 CONTTNUE क GO TO 10
6 1 ~ I F ( J . E Q . 1 ) 6 0 , 2 4 ~
60 J=J+1 $ IF(J.LE.2)8.24
62 IF(J.FQ.2)3.26
C I/O FORMATS.
    FORMAT(*OQUESTIONS WILL APPE R IN*I2* TABLE(S).*)
    FORMAT(* ANNOTATIONS ARE -*/ H ** UNANSWERED QUEST*
    + *ION*/* E FOR ERROR IN LAST ATTEMPT*/* < FOR *
    + *BOTH.*)
6 FORMAT(*OQUES. NO. *15I )
7 FORIMAT(* YAX. ANS. *,15I3)
36 FORMAT(* WHAT TABLE NEXT[ - )
41 FORMAT(* ENTER NEW*)
42 FORMAT(4221)
63 FORMAT()
64 FORMAT(*+*,9X,42R1,24X;*AS R AD IN.*)
    END
```

Fortran Listing of SUBROUTINE CODEIN：
SUBROUTINE CODEIN（ND）
C BY R．E．LOVELL 6－10－73．LAST PROGRAM REVISION 7－13－73． C LAST DECI ARATIVE CHANGES 9－2－73．

LOGIr．AL FLAG，MARK．ECHO
DIMENSION IFORM（4），IFMT（3），N（4），IREAD（20）
COMinON IGOTO．IANS，FLAG，DIGIT EECHO
COMMON／QA／ND，NQUES（72），NEFD（2．6），KB（2），KQUES（2．59）
DATA IFORA／7H2H＝＊）．7H8H＝），7H2L $=*), 7 \mathrm{HBL}=*) /$ ．
＋IFMT（1）／10H（＊NEED／日IFMT 3）／4HO20）／
PRINT 1 \＄IF（FLAG）PRINT $\&$
C BRING IN VALUES AND REJECT INVALID CHARACTERS． 004 J＝1．4
IFMT（2）＝IFORM（J）\＄PRINT IFMT \＄READ 2，IREAO
IF（EC．HO）PRINT 105．IREAD
ND（J）＝0 玉 MARK＝．F．$\$$ ITEST＝1R3
DO $3 \mathrm{~K}=1.20$ 玉 $\mathrm{KR}=60-3$＊K
IF（IREAD（K）．EQ．IRO）GO TO 3
IF（IREAD（K）．LY．1RO．O．IREAD（K）．GT．ITEST）GO TO 5
6 ND（J）＝ND（J）．O．SHIFT（IREAD（K）－33B，KR）\＄GO TO 3
5 MARK＝．T．
3 ．ITEST＝1R7 $\$$ IF（MARK）PRINT 7
4 CONTINUE
C Put in rinary values．
MASK＝1R \＄NMASK＝．N．MASK \＄NS KQ（1）
DO $5 \mathrm{n} \mathrm{K}=1, \mathrm{NS}$ \＆I＝KQUES $(1, k)$
$\operatorname{NEED}(\mathrm{I}, 3)=\mathrm{NEED}(1,4)=1 \mathrm{~B}$ \＄NDT $T=\mathrm{ND}(1)$ ．A．MASK
IF（NNTST．NE．0）NEEO（1．4）$=2 R$
NDIST＝ND（3）．A．MASK $\$$ IF（NOTST•NE．0）NEED（I，3）$=2 B$
ND（1）＝NO（1）．A．NMASK $\$$ ND（3）$=\mathrm{D}(3) . A . N M A S K$
MASK＝SHIFT（MASK．1）$\ddagger$ NMASK $=$ N．MASK
50 NEED（I，2）$=0$
C PUT IN ORTAL VALUES．
MASK＝ $7 B$ \＄NVIASK＝．N．MASK \＆KOUNT＝0 $\$$ NSEKQ（2）
DO Sn K＝1，NS \＄I＝KQUES（2，k）
NEED（1，4）$=\operatorname{SHIFT}(N D(2) . A \cdot V A S$ ，KOUNT）$+1 B$
NEED（I，3）$=\operatorname{SHIFT}(N D(4), A \cdot$ MAS,$K O U N T)+1 B$
ND（2）$=$ ND（2）．A．NMASK $\$$ ND（4）$=D(4)$ ．A．NMASK
NASK＝SHIFT（MASK．4）玉 NNASK＝N．MASK § KOUNT＝KOUNT－4．
$60 \quad \operatorname{NEED}(\mathrm{I}, 2)=0$ \＄MÃRK＝．F．
C TEST CORRECTNESS OF VALIJES，CORR CT，PRTNT ERROR MESSAGES．
DO 8n K＝1．4 § KOUNT＝0
DO 7n L＝1．20 \＄I＝20＊K－ $\mathrm{CO} 0+\mathrm{L}$ § IF（I．GT．NQ）GO TO 71
I＝NQUFS（I）\＆ITEST＝NEED（I，1）\＄IREAJ（L）$=0$
IF（NFFD（I．4）．GT．ITEST．DeNEED IG3）．GT゙．ITEST）GO TO 65
IF（NFED（I，3）．LE．NEED（I，4）：60 TO 70

```
65 KOUNT=KOUNT+1 $ IREAD(KOUNT) I
    NEFD(I,3)=MINO(iNEFD(I,3),NEE (I,4),ITEST)
    NEED(I,4)=MINO(ITEST,MAXO(NE D(I,4),NEED(I,3)))
70 CONTINUE 
75 IF(MARK)GO TÓ 76
    MARK=.T. I PRINT 61
76 PRINT 77,(IREAD(M),M=1,KOUNT
80 CONTINUE क MARK=.F.
C TEST FOR EXCESS INPUT DATA.
    DO 9n J=1.4
    IF(N\cap(J).EQ.O)GO TO 90
    IF(M\triangleRK)GO TO }8
    MARK=.T. $ PRINT 87
89 IFMT(2)=IFORM(J).A.771400000 0000000000B.0.4HAH*)
    PRINT IFMT
    CONTTNUE $ FRINT g9 $ GO TO 2
C ENCODE DATA.
    ENTRY RECODE
    PRINT 91
92 ND(1)=INEED(1,4) & ND(2)=INEED(2,4)
    ND(3)=INEED(1,3) $ ND(4)=INE D(2,3)
    DO 9R J=1.4
    IFMT(2)=IFORM(J).0.4000000B
98 PRINT IFMT,ND(J) & RETURN
C I/O FORMATS.
1 FORMAT(*OENTER DIRECTLY IN C DED FORM -*)
2 FORMAT(20R1)
7 FORMAI(* CONTAINS INVALID HARACTER(S).*)
FORMAT(* USE 2O OCTAL CHAR CTERS EACH.*)
61 FORMAT(*OERRORS NOTED AT QUE TION(S) -*)
77 FORMAT(1X.15I4)
87 FORMAT(* EXCESS INPUT DATA I -*)
91 FORMAT(*OENCODED RESULTS-*)
99 FORMAT(*OYOUR INPUTS WERE AC EPTED AS -*)
105 FORMAT(*+*,9X,20R1,46X,*AS R AD IN.*)
    END
```


## Subprogram QUEST

Subprogram QUEST is a subroutine for organizing the solicitation of answers to the entire battery of questions during the preparatory phase for interaction. Specifically it is called at Statement 120 in Section I of CHOOSE when the option of viewing and answering each question individually has been selected by the interactive CHOOSER. It calls QUERT, and ENTRY point in subprogram QUERY, to present the individual questions. In addition to calling QUERT it also enters BRANCH through BRANCH3.

Subprogram QUERY
Subprogram QUERY is a subroutine designed to pose individual aspiration level questions to the CHOOSER at the interactive terminal prior to calling CHAR to receive the response. It is entered directly by CHOOSE statement 204+1 during review of questions and answers, and through its ENTRY point QUERT by QUEST and DISPL.

## Subprogram TABLE

Subprogram TABLE is a subroutine which is used only for organizing the second optional input form for the question and answer set. It is entered by statement 130 of CHOOSE. It prepares the table presentation, up to 14 questions at a time, for interactive option 2 of CHOOSE. It accepts and processes the responses obtained, calling BRANCH (through BRANCH1, BRANCH2, and BRANCH3). It offers an alternate quick method to enter aspirations.

Subprogram CODEIN
Subprogram CODEIN is a subroutine assigned specifically to implement option 3 of input format which is selected by the interactive operator in section I of-SUBROUTINE CHOOSE. It also contains an ENTRY point RECODE which is sued by statement $401+1$ of CHOOSE for encoding responses. CODEIN is only used for entry of answers on coded forms; this presumes previous knowledge of the code as would be obtained, for example, from previous interactive sessions. It refers to FUNCTION INEED.

Subprogram DISPL
Subprogram DISPL is a SUBROUTINE for performing certain display operations required during interaction. It is called from section 5 of SUBROUTINE CHOOSE. DISPL is called at statement $590+1$ of CHOOSE. It is a key subprogram since it controls the interaction level, and presents information on individual candidates and how they fared with respect to the stated aspiration levels.

Fortran Listing of SUBROUTINE DISPL:

```
    SURROUTINE DISPL(ND,IBIAS,MO EL,KMDLS,KAND,OFORM
    + -CFORM)
C BY R. E. LOVELL 7-20-73. LAST PR GRAM REVISION 11-17-74.
C LAST DECI ARATIVE CHANGES 9-2-73.
    LOGICAL FLAG,MARK
    DIMENSION ND(4),MOUFS(72),JL 2),JH(2),LQUES(72)
    DINENSION MODEL(8.1),KMDLS(1 ,KAND(1)
    + - QFOFM(1).CFORM(1)
        COMMON/LABEL/IDATE,KDATE,NAME,NVDLS,IMDLS
    COMMON IGOTD,IANS,FI_AG,DIGIT, ECHO
    COMMON/QA/NS.NQUES(7?),NEED(72,6),KN(2),KQUES(2,59)
    PRINT 93,VODEL(1,IANS),MODEL(2,IANS)
    IF(FLAG)CALL CFILE(IANS:MODEL,CFORM)
    I1=.NOT.MJDEL(3.IANS)
    JL(1)=I1.AND.NO(3)
    JH(1)=I1.ANO.ND(1)
    JL(2)=.NOT.((MODEL(4,IANS).OR.IRIAS)-ND(4))
    JH(2)=.NOT.((MODEL(5,IAN5).O.IBIAS)-ND(2))
    LQ=MO=LQUES(1)=viQUES(1)=0
    DO 5n K=1.NQ & I=VQUES(K)
    II=Iつ=1 $ I3=-1
    IF(NFED(I,1).LE.2)GO TO 10
    II=2 $ l2=8 $ I3=-4
10 IF((,L(I1).ANU.I2).EN.0)GO T 30
    LQ=LQ+1 $ LQUES(LQ)=I $ GO T 40
30. IF((,JH(II).AND.I2).EQ.0)GOT }4
    MQ=MN+1 S MQUES(MO)=I
40 JL(I1)=SHIFT(JL(I1),I3)
    JH(II)=SHIFT(JH(I1),I3)
50 CONTINUE
    PRINT 95,IANS,(LQUES(K),K=1, Q)
    PRINT 96, (MQUES(K),K=1, Q)
55 RETURN
C ENTRY FOR CANJIDATE-CRITERIA COM ARISON BY QUESTION NO.
    ENTRY DISPQ
    N=IANS & IANT=KOUES(1)
61 IF(F|AG)PRINT 98 $ PRINT 97
    CALL RRANNCH3(72) з GO TO(83. 5.72.61),IGOTO
    63 IF(NFED(IANS,1)-2)64.55.66
    64 PRINT 99,IANS $ GO TO 61
    65 KL=KH=(SHIFT(MODEL(3.N),NEED IANS:5)).AND.1)+1
    MARK=.FALSE. $ GO TO 67
    66 KL =(SHIFT(MODEL(5,N).NEED IANS;5)).AND.7)+1
    KH=(SHIFT(MODEL(4,N)ANEED IANS,5)).AND.7)+1
    MARK#:TRUE.
```

    CALL QUERT(IANT, QFORM)
    GO Tn(73,55,61,72), IGOTO
    KS=-NFED (IANT,5)
    IF (MARK)GO TO 75
    IS=.NOT.SHIFT(1.KS)
    ND(1) \(=\) IS.AND.ND(1).OR.SHIFT(NEED(IANT, 4)-1,KS)
    ND(3) \(=15\).AND.NO(3).OR.SHIFT( EED(IANT,3)-1,KS)
    GO Tn 61
    IS=.NOT.SHIFT(7.KS)
    ND(2) \(=\) IS.AND.ND(2).OR.SHIFT( EED(IANT,4)-1.KS)
    ND(4) \(=15\).AND.ND(4).OR.SHIFT( EED(IANT,3)-1,KS)
    GO Tn 61
    C I/O FORMATS.
93 FORMAT(14H SHORT NAME - .2A10)
94 FORMAT(GH QUES.,I.3.11H YOUR INPUT,I2,1H-II1,
+5 H . AND.I4.9H IS RATED.I?.1 -.II. 6 H . MAX $=$.II)
FORMAT(IX.I3,20H. CRITERIA N T MET -112I3/(20I3))
FORMAT(24-1 MARGINAL RESPONSE ARE 12I3/(20I3))
FORMAT(13HONUES. NO. - )
FORMAT(44HOGIVE QUESTION NO. USE $S$ TO EXIT, OR R
+ GHTO MODIFY/34it RESPONSES TO LAST QUESTION CITED.)
99 FORMAT(9H QUESTION.I3,12H NO IN USE.)
END

Fortran Listing of SUBROUTINE SPECIAL:
SURROUTINE SPECIAL (MARK,MODEL,KMDLS,KAND,QFORMOCFORM)
C BY R. E. LOVELL 6-12-73. LAST PROGRAM REVISION 12-6-74. C LAST DECIARATIVE CHANGES 11-17-74.

LOGICAL FLAG, MARK, ECHO,KTRL
DIMENSION YODEL(8,1),KMDLS(1),KAND(1)

+     - QFORM(1). CFORM(1)
COMIMON/LABEL/IDATE,KDATE,NAM ,NMDLS.IMDLS
COMMON IGOTO, IANS,FLÁG,DIGIT1,ECHO
COMMON/QA/NO,NQUES(72),NEED(72.6),KQ(2),KQUES (2.59)
7 PRINT 6 \& KTRL=.T.
1 PRINT 2 £ IF(FLAG.A.KTRL)PRI T 3 g KTRL=.F.
6 FORMAT(*-SECTION $X$. ALGORITH SERVICE AREA.*)
2 FORMAT (* SPECIAL CHOICES A E 1 THRU 9.*)
3 FORNATI* 1 - RETURN TO CALLI G PROGRAM.*/
+25 H 2 - LIST THF NEED ARRA ./
+ 24H 3-ADD A LIBRARY ITEM /
+     * 4 - MODIFY A LIBRARY ENTR .*/
+     * 5 - LIST SELECTED LIRRARY ENTRIES.*/
+     * 6 - LIST aLL LIBRARY ENTR ES.*/
+     * 7 - INITIALIZE LIRRARY RE ISTER.*/
+     * 8 - INITJALIZE THE Q. AND A. SET.*/
+     * 9 - CHANGE ECHO CHECK.*)
CALL BRANCH1(9) \$ GO TO(5,4, ,7,4), IGOTO

5. GO Tn(4,200.300.400.500.600,700.800,900), IANS

C 1. RETURN TO CALLING PROGRAM.
RETURN
C 2. TO LIST THE NEED ARRAY.
200 DO $2 \pi 1 K=1$, NQ
I=NQUES(K)
201 PRINT 202.I,(NEED(I,J):J=1.6
202 FORMAT(5I4,I5,I7)
GO Tn 1
C 3. TO AND A LIBRARY ITEM.
300 GO Tn 999
C 4. TO MODIFY A LIBRARY ENTRY.
400 GO Tn 999
C 5. List selected library entrie -
500 GO Tn 999
C 6. LIST ALL LIRRARY ENTRIES.
600 PRINT GU1. VAME,KDATE
601 FORMAT(1H1,A10.19H LIBRARY L ST AS OF.R10/)
00 Gn $3 \mathrm{~K}=1$. NWDLS
$\mathrm{I}=$ Kinn! $5(\mathrm{k})$
603 PRINT 604,I:MODEL(1,I), MOTEEL 2, X), MODEL(4,I). $+\operatorname{MDDFL}(3, I) \cdot(M O D E L(J, I) \cdot J=5$,

```
604 FORMAT(1X,I3,2X,2A10,3X,020/3X,2(3X,020),2I3,I6)
    GO Tn 1
C 7. INITtalIZE the LIfRARY REGIStER.
    ENTRY SPEC7
700 NMDL¢=0
    DO 7ñ I=1.1MOLS
    IF(MNDEL(8,I) .LE. 0)GO TO 7n2
    NMDL\subseteq=NMDLS+1
    KMDL\subset(NMDLS)=I
702 CONTTNUE
    IF(MARK)RETURN
    GO TO 1
C 8. INITIALIZE THE O. AND A. SET
    ENTRY SPEC8
800 II=J,IニNQ=0
    DO 810 I=1,72 $ IF(NEEO(I,1)-2)801,803,804
801 NEEO(I,1)=0 $ GO TO 810
803 II=IT+1 $ KQUES(1,II)=I $ NEED(I,5)=1-II $ G0 T0 805
B04 JJ=J.1+1 $ KQUES(2.JJ)=I $ NEED(I!5)=SHIFT(1-JJ,2)
805 NEED(1,2)=3 $ NEED(I,3)=1 $ NQ=NQ+1 $ NQUES(NQ)=1
    NEED(I,4)=NEED(I,1)=MINO(8,NEED(I,1))
B10 comTtNUE
    KQ(1)=II & KQ(2)=,JJ
    IF(IT.GT.59)STOP 60 & IF(JJ.GT.14)STOP 15
    IF(MARK)RETURN
    PRJNT 812 $ GO ro 1
812. FORMAT(32H NEED ARRAY WAS RECONSTRUCTED.)
C 9. CHANGE ECHO CHECK.
900 ECHO=.NOT.ECHO $ IF(ECHO)PRI T 901 $ GO TO 1
901 FORMAT(* ECHO CHECKS OF IN UT WILL APPEAR.*)
999 PRINT 950,IANS $ GO TO 1
950 FORMAT(*0 PROGRAG FOR OPTIO *I3* IS NOT READY YET.*:)
    END
```

It has an ENTRY point DISPQ for causing display and review of specified questions. It also updates the coded aspiration information when the CHOOSER modifies his responses. For this purpose it calls QUERT. DISPQ is called at statement $591+1$ of CHOOSE.

DISPL calls QFILE and CFILE when needed to display detailed textual information on the question and answer set and candidate set, respectively. It also calls BRANCH through ENTRY point BRANCH3.

Subprogram SPECIAL
Subprogram SPECIAL is a subroutine which permits a more knowledgeable user to call for special performance during certain parts of the operation. It contains 9 optional functions as follows:

1. Return to calling program.
2. To list the NEED array.
3. 
4. $\}$ Reserved for later use
5. 
6. List all library entrịes.
7. Initialize the library register.
8. Initialize the Q and A set.
9. Change echo check.

Option 1 is an option to exit back to the calling point. Option 2 provides a list of the NEED array. Option 6 outputs a complete list of the candidates. Options 2 and 6 are sometimes of interest to EVALUATORs when reviewing the makeup of the main program.

Option 7 (initialize library register) and Option 8 (initialize question and answer set) are available as direct ENTRY points to the controlling CHOOSE subprogram. They are called by statements $30+2$ and 50, respectively.

Option 9 (change echo check) is an option which can be entered from the keyboard if necessary but is usually activated when only operating in batch mode. In this mode it causes a printout of the answers to individual questions taken from data cards. If it were not for this feature, operating the procedure in batch mode would be difficult since the input data would otherwise not be visible.

Subprogram SPECIAL is not normally accessible to the CHOOSER since its function, when needed, is obtained automatically. Persons

Fortran Listing of SUBROUTINE QFILE:

```
            SURROUTINE QFILE(IMMODEL,FORM)
C BY R. E. LOVELL 11-17-74. LAST R VISED 12-6-74.
            DIMENSION FORV(1),MODEL(R,1)
    COMMON/QA/N.,NQUES(7?),NEED( 2.6ioKA(2)OKQUES(2.59)
    K=NEFD(1,6)
    GO TO 1
    ENTRY CFILE
    K=MONEL(8.I)
    1 PRINT FORM(K)
    RETURN
    END
```


developing CHOOSE main programs, however, can get access to SPECIAL by entering a " 3 " when the YES or NO response to first READY? question is solicited after the "University of Arizona" label is printed out. Since this method of access is for restricted use, it is not identified to the interactive user.

For batch, rather than interactive, the appropriate first two data cards are 3 (to get access to SPECIAL) and 9 (to cause ECHO of inputs to be printed). This can be followed by 2 (to list the NEED array) and 6 (to list library entries), but must finally be followed by a 1 to return to CHOOSE for continuation.

Subprogram QFILE
Subroutine QFILE is a small subroutine for controlling access to the textual information stored in variable formats in the main program. Entry through the name QFILE gives access to the messages related to the question and answer set. Access through the ENTRY point CFILE gives access to the text associated with specific candidates.

Subroutine Calling Structure
For convenience in understanding the CHOOSE calling structure, Figure 20 shows the various program linkages.

## APPENDIX D

## COMPLETE PROBLEM METHODOLOGY

In the main body of the text mention was made at several points that the procedures developed during these investigations would be applicable over a broad range of activity. This appendix contains the step by step procedure for generating such interactive programs.

To assist research application teams in establishing new classes of candidates and building corresponding "libraries" of performance and resource requirements data for other problems similarly situated, two auxiliary computer programs were developed. The first one, named MAKEUP, actually prepares two card decks. It uses as inputs the numbers of the questions assigned and the upper bounds on the number of answers for each question. One of the two decks prepared by MAKEUP is used in the second auxiliary program to aid in preparing the library. The other becomes the nucleus of a new Fortran main program for use interactively with the CHOOSE subprogram set described in Appendix C. Details of PROGRAM MAKEUP are given in Appendix E.

The second auxiliary program, CODEM, accepts as inputs identification and rating information on particular candidates prepared by EVALUATORs--those responsible for establishing the attributes of each candidate in the selected question and answer format. It produces in coded form the additional punched card records needed to complete the new main program prepared by MAKEUP. Details of CODEM are given in Appendix F .

The use of the two auxiliary programs requires practically no knowledge of computer programming either in machine language or in the higher level languages. The entire system is made so that only data cards need be prepared and processed. Additionally, the formats for the data cards have been made as simple as possible. Hence, evaluation teams which may be formed to make "Consumer Reports" on various elements of a candidate set need not include any significant capability for computer programming.

Given that a project is found which is suitable for treatment by the procedures described in Chapters 3 and 4, the step-by-step procedures outlined below can be followed to create the necessary program and data base (library of information) needed for interactive use with the CHOOSE system. Note, however, that it may not be possible to determine whether the project or problem is treatable by these methods until most of the second step below is completed. Figure 21 may be used to follow the steps of the procedure.

1. Label and Identification. To establish project or program identity on a formal basis, the following steps are required in preparation of the first (or header) data card for PROGRAM MAKEUP:


Fig. 21. Step by Step Process for Project TPDEK8
a. Select a Fortran mnemonic name for the project. The name must begin with a letter and can be followed by up to five alphabetic or numeric characteris. The name should be chosen so that there can be no confusion with other programs in the total set of such packages. Preferably, the first three characters should be unique since they become the identifiers in columns 73-75 of the cards produced by programs MAKEUP and CODEM. The name selected is entered in columns 73 through 78 of the first data card for PROGRAM MAKEUP.
b. Pick the date you wish to associate with the question and answer set. Enter it in columns 43-50 of the first data card for MAKEUP. The preferred form is 10-21-73 for the 8 columns allotted.
c̄. Select a maximum or ceiling for the number of candidates you expect to introduce to the program. This number will determine the amount of computer memory storage which will be allotted when the interactive program is executed. Therefore, the number should be selected somewhat conservatively to reduce operating costs and to insure that computer memory capacity is not exceeded. The number selected here should be entered in columns 68-70 of the first MAKEUP data card. The number should be right justified and written without decimal point or other punctuation. If you fail to make an entry in this space, the program will assume a ceiling of 50 .
2. Prepare the Question and Answer Set. Select a set of decoupled or independent questions or queries and an answer set for each question. Two kinds of questions are permitted: binary--questions which admit of only two answers, usually yes or no, and higher order questions ( 3 answers minimum, 8 answers maximum). Each answer set must be capable of being totally ordered on the basis of some preference relation. Each answer set must be ordered on the basis of this preference relationship so that the answer with the smaller number indicates a less stringent performance requirement or capability, or a smaller availability of resources, than those with higher numbers.

It is important to recognize that this is the most important step in the process. It is here that it may be realized that preparation of questions in suitable form may not be possible, or may require some transformations not ordinarily done. For example, neither conductivity nor density may be of importance in selection of material for an electric wire for a particular application; a more important criteria might be a quantity represented by conductivity divided by density. Hence, the additional question for this attribute should be included.

For the present implementation of this interactive procedure on the CDC 6400 computer the number of binary questions must be limited to 59 , the number of higher order questions to 14 , and the total number of questions to 72 . The CHOOSE system could be modified to extend this range.
3. Numbering the Questions. Assign a number to each question from the set of integers 1 through 72. It is not necessary that all numbers be used or that they be assigned consecutively. Question
numbers may be grouped or spaced as convenient for the analysis team. Spaces may be left for possible insertion of additional questions later.
4. Preparing the Second Data Card. Prepare the second data card for PROGRAM MAKEUP. Here, the column numbers of the card represent the question numbers selected in step 3. For each of the questions in use, punch in the corresponding column the number of distinct answers that have been developed for that particular question. Leave blank any of the first 72 columns which are not associated with questions. Columns $73-80$ may be used for card identification.
5. Execution of MAKEUP. Execute PROGRAM MAKEUP with the two data cards prepared in steps 2 and 4. Examine the printed output to see that no error messages were printed and that the output list reflects the input intended. The program provides error messages when the maximum permissible number of questions in any category is exceeded, or if any improper symbols are inadvertently punched on the input data cards.
6. Interpretation. Run the punched cards produced by MAKEUP through the interpreter to make them easier to identify and handle. Each of the cards produced will have the first three characters of the name you selected in step 1 punched in columns 73-75. This will help in keeping various decks separated when work is in progress on more than one package at the same time. The interpreted deck must now be divided into two parts. The first part consists of cards numbered from 10001 through 90003 and constitutes the nucleus of the CDC main Fortran program which is to be combined (see step 9) with the CHOOSE subprograms to produce the complete interactive program for the project. The second part consists of cards numbered in columns 79-80 from 1 through 18, and constitutes a complete CDC Fortran BLOCK DATA subprogram for use with program CODEM as described in step 8.
7. Analysis of Candidates. Step 7 is the second part where the expertise, objective analysis, educated discretion and/or experienceconditioned judgment enter into the process. Here, each of the candidates must be evaluated for each of the attributes represented in the question set. Answers must be selected for each of the questions for each of the candidates. For each of the binary questions the evaluation team must select unequivocally one of the answers. For the higher order questions it is permissible--and useful--to evaluate candidates by selecting a lower bound and an upper bound answer so that variations in candidate evaluation can be accommodated. This feature helps give the added leverage needed during interaction to develop and display information helpful to the interactive decision maker as he attempts to modify his resources and requirements criteria to move towards a feasible solution.
8. Preparation of Data for CODEM. Enter the evaluation information from step 7 on $80-$ column EDP cards, 3 cards per candidate. These cards then become the data cards for use with PROGRAM CODEM, with the BLOCK DATA subprogram prepared by MAKEUP appended. The format for preparation of the data cards is very simple and is given in Appendix F.
9. Execution of CODEM. Execute PROGRAM CODEM to obtain both a line printer output and additional punched cards. Review the line printer output to see that any errors detected, corrected and listed by CODEM are acceptable. If acceptable, insert the cards produced by CODEM by serial numbers (columns 76-80) into the nucleus Fortran Main program which was prepared by MAKEUP in step 6 and which were set aside for this purpose.

Note that steps 7, 8 and 9 can be repeated completely or in part as many times as needed to produce updated BLOCK DATA cards for particular candidates, to add candidates, or to redo the evaluation package completely. To delete obsolete or incorrect evaluations, it is only necessary to delete the corresponding cards from the main program. If, however, additional questions are to be posed, or the question and answer set is to be modified in any way, it is necessary to return to step 1.
10. Compile and Store the New MAIN Program. Affix the appropriate control cards for compilation and storage of the new MAIN program. The following deck structure is suitable for the CDC 6400 installation at the University of Arizona using the main program TPDEK8 as the example.

```
L\emptysetVELLR,BNxxxxxxxx,T5,CM50000,ST0
REQUEST (TPDEK8,*PF)
C\emptysetPYCR (INPUT,TEMP)
REWIND (TEMP)
FTN(I=TEMP , B=TPDEK8)
CATALøG(TPDEK8,TPDEK8,ID=REL)
7 89
        PR\emptysetGRAM TPDEK8(INPUT,\emptysetUTPUT) . . . TPD10001
        •
        •
        END
        TPD90003
6}\mp@subsup{7}{89}{
```

With the completion of step 10, the new program is ready for testing by the evaluation team and for use by clients as outlined in Appendix A.

Note, however, that the procedures here do not provide for the texts of questions and answers to be made available in the computer storage. Inclusion of text materials for interaction presentation, requires substantial additional effort which may be counterproductive. See the discussion in Appendix $B$ for additional information.

## APPENDIX E

## PROGRAM MAKEUP

The purposes of the PROGRAM MAKEUP are as follows:

1. To accept basic information from any decision problem that can be formulated in the context of the total ordering of responses to individual questions as formulated in Chapter 3. This information consists of a problem name, date, and the number of answers associated with each of the question numbers to be used.
2. To check the values proposed to see that they meet the specifications for entry into the interactive PROGRAM CHOOSE. (Limits: 59 binary questions, 14 questions permitted 3 to 8 answers, 72 questions total).
3. To print error messages if any of the criteria are not met.
4. To produce nucleus main programs which can be completed by cards produced by CODEM.
5. To prepare in punched card form the complete BLOCK DATA subprogram needed for PROGRAM CODEM. (PROGRAM CODEM is used in turn to generate additional cards for use in main programs designed for CHOOSE execution.)

In those problems amenable to interactive choice by the proceedures outlined in the text and Appendix A, use of PROGRAM MAKEUP is the first step in the set up of the required information. Refer to Appendix $D$ to see its position in the complete procedure.

As with the other programs prepared for this research, the language of MAKEUP is not machine independent. Since the Univac 1110 was used in preparation of the examples in this study, the program was written to accommodate the character sets, word lengths and Fortran character manipulation statements of the Univac 1100 series Fortran V compiler. Note, however, that the materials prepared by MAKEUP are in the form needed for execution on the CDC 6400 computer. If this program is to be used on any computer other than the Univac 1110, it must be rewritten to the extent needed to insure compatibility with the computer being used.

## Input

Apart from the Fortran program itself and the control cards needed to enter the job into the computer, two data dards must be prepared as follows:

1. Card 1 -- Label and Identification

Columns 43-50 The date of this version of the question and answer set. The preferred form is month, day, and year separated by hyphens as follows: 10-12-72.

Columns 68-70 The maximum number of candidates which are expected to be entered into the program. The number must be right justified in the available space. The allocation of only 3 columns to this parameter automatically limits to 999 the number of candidates which may be considered. If entry here is omitted or is less than 50, MAKEUP will supply a value of 50.

Column 72. For normal operation, this column is left blank. To suppress the production of punched cards on output, punch any alphabetic or numeric character in column 72. This option is convenient if the program is to be executed for test purposes, and the punched card output is not desired.

Columns 73-78 The program label. The first character must be alphabetic. It may be followed by up to 5 alphabetic or numeric characters.

All other columns of the first card may be used for any purpose by the user. Typically, they are used to describe the entries on the card and identify the project.
2. Card 2 -- Raw Question and Answer Set Data. The first 72 columns of the second data card represent a question number. For each question in use the number of answers in the answer set is punched in the column number corresponding to the question number. Columns 73-80 may be used for identification. If a particular question is not in use, the corresponding column must be left blank.

PROGRAM MAKEUP allows the preparation of more than one set of outputs during a single execution. This is done by appending as many pairs of data cards to the program as outputs are desired. They should not be separated either by blank cards or in any other way.

## Oritput

The output of PROGRAM MAKEUP appears in two forms. The first is a line printer output which gives an "echo" of the input data cards. A presentation of any detectable errors in the input is made. The errors which are capable of detection are

1. The number of binary questions exceeds 59.
2. The number of higher order questions exceeds 14 .
3. Improper characters appear in columns 1-72 in the second data card. Only blanks and the numbers 2 through 8 are allowed to appear.
4. An improper choice of program label. Errors which cannot be detected are: improper data or attempts to use columns 73-80 of the second data card to establish the question and answer set.

The error listing is followed by a list of images of each of the punched cards produced.

The second form of output is the deck of output cards. The images on these cards are identical to the corresponding output lists mentioned above. Columns 73-75 of these cards are used for project identification and contain the first three characters of the selected program label. The first series of cards constitute the basic statements for a main interactive program of the type illustrated in Appendix B. Only the Fortran DATA statements to be prepared by CODEM are needed to complete the program. The remaining cards constitute a complete BLOCK DATA subprogram for PROGRAM CODEM (Appendix F).

Figure 22 on the following page shows the data cards used by MAKEUP in preparing the initial materials for the sample case used in the main body of this dissertation. Figures 23 and 24 show the computer output for the data set of Figure 22. This appendix concludes with a complete listing of the Fortran statements of PROGRAM MAKEUP.
Fig. 22. Input Cards for MAKEUP using TPDEK8 Example

Fig. 23. Computer Output for MAKEUP using TPDEK8 Example t000ICd


|  |
| :---: |
|  |  |
|  |  |

- 

$\because$. 1.al.ivn

Fortran Listing of PROGRAM MAKEUP:

 H H H




| U00 | $\cup$ | uvu |
| :---: | :---: | :---: |
| ¢ < | 4 | ¢ ¢ |
| $\ggg$ | $>$ | $\ggg$ |
| HMH | H | HッM |
| $2 \geq 2$ | 2 | z 2 |




## $U$ 3 3 3

IF(FLAG)PUNCH $94,(M(I X), I X=I S T A R T, I S T O P), ~ N A M E, L$
CONTINUE
$L=L+1$
PRINT 95, (M(IX),IX=61,72), NAME',L
IF(FLAG)PUNCH 96. (M(IX), IX $=61,72)$, NAME,L

CONTINUE
GOTO 1

0

## APPENDIX F

## PROGRAM CODEM

The purpose of program CODEM is to accept candidate data from data cards which are prepared by EVALUATORs. The data is processed by CODEM. The output is both a listing of the resulting information with any errors identified and a set of punch cards sequentially numbered for insertion in the associated main program. PROGRAM CODEM also prepares revised date cards for insertion in the main program so that the continuity of changes by date is maintained. Use of CODEM permits the EVALUATORs to add or modify information for main programs previously prepared, and to create the completely new data sets (cards 8001089993) needed when the question and answer sets prepared for a particular CHOOSE problem are updated.

PROGRAM CODEM has been prepared and executed in two forms: one, for use with the CDC 6400 series Fortran and one for the Univac 1110 Fortran V. Only that prepared for the CDC 6400 is listed here. Again since the program is quite dependent on the character sets, word lengths and Fortran character manipulation statements of the computer being used, the programs cannot be made machine independent.

Since PROGRAM CODEM prepares data in the context in a particular question and answer set, to function properly with a candidate set it is necessary to compile and link with PROGRAM CODEM the BLOCK DATA subprogram developed by PROGRAM MAKEUP for the same CHOOSE problem. See Figure 21 in Appendix D for the required deck structure. Safeguards are built into the PROGRAM CODEM so if any attempt is made to use data cards which are not coded (in columns 73-75) properly for use with the associated BLOCK DATA subprogram error messages are printed out.

Data requirements for CODEM are as follows: One header card for the data must be prepared in the following form:

Columns 73-78 The full 6 character identifier for the CHOICE program under study, for example, in the sample program used here the 6 character word is TPDEK8.

Column 51-58 The date entered in 8 characters in exactly the format used in PROGRAM MAKEUP for identifying the question and answer set.

Column 72
For normal oepration, this column is left blank. To suppress the production of punched cards on output, punch any alphabetic or numeric character in column 72. This option is convenient if the program is to be executed for text purposes, and the punched card output is not desired.

Following this, three cards for each candidate must be prepared in specialized format. The form for the first one is as follows:

Column 4-6 The candidate number right justified in the field. This number must be equal to or less than the maximum number of candidates for which the main program is designed. The maximum number of candidates is shown on card 10013 of the main program and on the DATA card (variable name IMDLS) in the BLOCK data subprogram associated with CODEM for the particular project.

Columns 22-41 Insert a short unique 20 character name or identifier for the candidate.

Columns 45-52 Insert the date of the evaluation for the candidate. The following form is preferred: 10-19-74.

Columns 58-59 These columns are available for a version number if desired. Typically the number 1 would be right justified in these two columns for the first version and as later evaluations of the same candidate took place the number would be increased by 1. However, and Fortran characters that the user desires can be used in these two columns.

Columns 73-75 The first 3 characters of the name of the main program must be inserted in these columns. Unless these 3 characters are the same as the first 3 characters of the main program mnenomic name the data point will be rejected by CODEM.

Cards 2 and 3 have identical format. Each is to contain an integer in each of the columns ( 1 thru 72) which is identified with any question in the desired question and answer set. Card No. 2 contains the EVALUATOR's highest evaluation for the candidate. Card No. 3 contains the lowest. Columns 73 through 75 must be labeled the same as card No. 1.

A computer listing of a typical header card and several candidate cards for input to CODEM are given in Figure 25.

## Output

The line printer output of PROGRAM CODEM contains a heading with some identification followed by information for each candidate in two forms: (1) an echo of the input values is printed for each comparison with the EVALUATOR's handwritten notes, and (2) the images of the four cards punched for each candidate.

If column 72 of the first card of the data (the header card) has been left blank, cards will be punched exactly as they apear on the
line printer output. Each of the cards contains in columns 73 through 80 the 3-character project identifier, bollowed by a serial number based on the candidate number so that it may be inserted automatically in proper sequence into the associated main program by a card sorter if desired.

Error Messages: In addition to printed messages for those candidates which appear to be in the wrong deck for the BLOCK DATA program associated with CODEM, the following error messages are used.

* An asterisk indicates that the numerical value proposed by the EVALUATOR is outside the range allowed for that particular question.
\$ The dollar sign indicates that there is an error in the low value and that the low value has been adjusted accordingly. This can occur for two reasons. In a binary question the EVALUATOR is expected to make an unequivocal evaluation. If he assigns with different values for the high and low input for a binary question, adjustment is made and the error message given. Further if the EVALUATOR should indicate a low answer higher than the high one for a higher order question a correction is made and an error message printed.
\# The number symbol indicates an attempt was made to answer a question which is vacant. PROGRAM CODEM ignores such inputs except for printing out the error message. To the EVALUATOR this may indicate that one of his valid answers has been punched in the wrong column.

A segment of the computer output from an execution of CODEM with TPDEK8 candidate cards is given in Figure 26.

PROGRAM CODEM may be executed many different times with the same BLOCK DATA program to prepare new or revised cards for the associated main program. If, however, the question and answer set for a particular CHOICE program is modified then the BLOCK DATA subprogram for CODEM should be modified correspondingly. Then each of the candidates must be updated in accordance with the revised question and answer set. The Fortran source code for PROGRAM CODEM completes this appendix.

## OOनNO～NONN <br> 음́ㅁ́acococ <br> トトトトにトにトにに



| 0 0 | 100 | 00 |
| :---: | :---: | :---: |
| －1 | $\cdots$ | －r |
| $\cdots \mathrm{H}$ | －r | － |
| －1－1 | $\cdots-1$ | －1 |
| N | －r | － |
| ก | $\pm \pm$ |  |
| in $n$ | 15 | in 10 |
| N | M N | No |
| in in | ก | ¢ |
| M M | $\pm \pm$ | $\cdots \mathrm{m}$ |
| －MMN | さ | $\bigcirc$ |
| $\pm$ ¢ | M N | $\sim \sim$ |
| $\cdots \mathrm{m}$ | $\cdots \mathrm{M}$ | －r |
| $\pm \pm$ | $\pm \pm$ | $m \mathrm{~m}$ |
| $\pm M$ | $\cdots \mathrm{m}$ | $\cdots \mathrm{m}$ |
| $\pm \pm$ | $\pm \pm$ | $\pm$ |

$$
\begin{aligned}
& \text { UVIVAC PREPARATION OF BLOCK DATA CARDS USING QUESTION AND } \\
& \text { AND ANSWER SET NAMED PRNDSET' AND DATED 7-11-73. } \\
& \text { ERROR CODES } \\
& *=\text { VALUE OUT OF RANGE. MINIMUM SUBSTITUTED. } \\
& \cdot \$=\text { ERROR IN LOW VALUE. LOW VALUE ADJUSTED. } \\
& H=A T T E Y P T ~ T O ~ A N S W E R ~ V A C A N T ~ Q U E S T I O N . ~ I G N O R E D . ~
\end{aligned}
$$

Fig. 26. Segment of Line Printer Output from CODEM

Fortran Listing of PROGRAM CODEM:
PROGRAM CODĖMIINPUT, OUTPUT, PUNCH.TAPE5=INPUT,TAPEG=OUT CDC PREPARATION OF MAIN PROGRAM DATA STATEMENTS FOR USE WITH C THE SUBPROGRAM PACKAGE NAMED CHOOSE. C BY R. E. LOVELL 6-20-73. LAST REVISED 11-16-74. C CARDS PREPARED ARE FOR USE IN CDC 6400 FORTRAN. C DATA deck structure -
C A. HEADER CARJ AS FOLLOWS C UPDATE DATA DECK FOR PROGRAM ' ',DATE ' '. C AAAAAA RRRRRRRR C ANY NON-BLANK CHARACTER IN COLUMN 72 ABOVE SUPPRESSES C THE PUNCHING OF OUTPUT DECKS. (USE FOR EARLY ANALYSIS) C B. THREE ADDITIONAL CARDS PER CANDIDATE C 1. LABEL AVD ID AS FOLLOWS CO. - SHORT NAME - VER= • C III AAAAAAAAAAAAAAAAAAAA RRDATERR AA C 2. HIGH VALUE INPUT CRITERIA AS FOLLOWS CRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR C 3. LOW VALUE INPUT CRITERIA IN THE SAME FORM AS 82. LOGICAL MARK(2),TAG
INTEGER TNAME,TIDATE
DIMENSION JAME(2),N(72,2)
COMMON/LABEL/IDATE,KDATE,NAME,NMDLS.IMDLS
COMIMON/QA/NQ,NQUES(72),NEED(72.6),KQ(2),KQUES (2.59)
C TAGE.FALSE. ALLOWS PUNCHING 10 TAKE PLACE.
TAG $=$. TRUE.
C INITIALIZE THE Q AND A ARRAY -- NEED(72.6).
800 II $=J J=N Q=0$
DO 810 I=1,72 $\operatorname{IF} \operatorname{IF}(N E E D(I, 1)-2) 801,803,804$
801 IF(NEED(I,1) .EQ. 0) GO TO 810
$K L=0$ \$ $\operatorname{WRITE}(6,85) Y$ INEED(I,1),KL $\$ \operatorname{NEED}(I, 1)=0$
60 TO 810
803 II =II + 1 \$ KQUES(1,II) $=I$ \$ GO TO 805
804 IF(NEED(I,1) .LE. 8) GO TO 806

$806 J J=J J+1$ \$ KQUES(2,JJ) = I
$805 N Q=N Q+1$ S $\operatorname{NQUES}(N Q)=I$
810 CONTINUE
$K Q(1)=I I \quad \$ \quad K Q(2)=J J$
IF(II •LE. 59 .AND. JJ •LE. 14) GO TO 811
WRITE(6,82) II,JJ, (NEED(I,1),I=1,72)
77 STOP 77
C READ HEADER CARD, TEST, AND PRINT HEADINGS.
811 READ(5.8B)TNAVE,TIDATE,KAG
IF(ENDFILE 5)60061
60 WRITE (6.84)NAME

STOP 60
61 IF(TNAME •EQ. NAME .AND. TIDATE •EQ. IDATE)GO TO 62
WRITE (6,83) TNAME,TIDATE,NAME, IDATE
STOP 61
62 IF (KAG •EQ. $1 R$ ) TAG =.FALSE.
WRITE (6,89). NAME,IDATE
NAMH $=$ SHIFT(-42.NAME)
C. INDIVIDUAL EVALUATION OF EACH OF THE CANDIDATES OFFERED.

10 READ (5,90)NR, JAME, MDATE, IVER,NAMF
IF(ENDFILE 5) 1,2
1 STOP 30
2 IF(NAMH •EQ. SHIFT(-42,NAMF))GO TO 6
KARD $=1$ \$ READ $(5,86)$
$7 \operatorname{READ}(5,86)$
8 WRITE (6,80)NR,JAME,MDATE,IVER,NAMF,NAME,KARD
GO TO 10
$6 \operatorname{READ}(5,91)(N(I, 2), I=1,72)$, NAMF
IF (NAMH . EQ. SHIFT(-42,NAMF))GO TO 3
KARD $=2$ S GO TO 7
$3 \operatorname{READ}(5,91)(N(I, 1), I=1,72), \operatorname{NAMF}$
IF(NAMH •EQ. SHIFT(-42,NAMF))GO TO 4
KARD $=3$ \$ GO TO 8
4 PRINT 92,NR,JAME,IVER,MDATE, (I,I=1,7), ((J,j=1,9),K= $+1,7),(N E E O(I, 1), I=1,72),(N(I, 2), I=1,72)$
$M R=N R+8000$
MSTOP $=$ NR*B $\$$ MSTART $=$ MSTOP -7
DO $15 \mathrm{I}=1,72$
$15 \operatorname{NEED}(1,3)=\operatorname{NEE}(1,4)=0$
C BEGIN ANALYSIS AND INDIVIDUAL COMPUTATIONS.
DO $30 \mathrm{~K}=1.2$ \$ $J=3-K$ \$ MARK $(J)=$.FALSE.
DO $28 \mathrm{I}=1,72 \$ \operatorname{NEED}(\mathrm{I}, 2)=1 \mathrm{R}$
IF(NEED(I,1).EQ.O.AND.(N(I,J).EQ.1RO.OR•N(I,J).EQ.1R

+ )) GO TO 28
IF(N(I,J).GE.1R1.AND.N(I,J).LE.NEED(I,1)+1R0)GO TO 27
$\operatorname{NEED}(I, 2)=1 R * \$ \operatorname{IF}(\operatorname{NEED}(I, 1) \cdot \operatorname{NE}, 0) \operatorname{NEED}(I, J+2)=1$
$\operatorname{MARK}(J)=. \operatorname{TRUE}$. \$ GO TO 28
$27 \operatorname{NEED}(I, J+2)=N(I, J)-1 R 0$
28 CONTINUE
IF(J.EQ.2)GO TO 29
DO $35 \mathrm{I}=1,72$
$\operatorname{IF}(\operatorname{NEED}(\mathrm{I}, 4)$.GE.NEED (I,3)) GD TO 35
$\operatorname{NEED}(I, 3)=\operatorname{NEED}(I, 4)$ \$ MARK(1)=.TRUE.
$\operatorname{NEED}(1,2)=1 R \pm$
35 CONTINUE
PRINT 95: (N(I,1), I=1,72)
29 PRINT 93.(NEED (I.J+2):I=1,72)
IF(MARK(え))PRINT 94;(NEED(I,2):I=1,72)
30 CONTINUE
NDI=INEED $(1,4)$ \& ND2=INEED $(2,4)$
ND3=INEED $(1,3) \mp N D 4=\operatorname{INEED}(2,3)$
$C$ PROUUCE OUTPUT FOR ONE CANDIDATE.

```
        PRINT 101
    PRINT 102 ,NR,JAME,IVER,MDATE,NAME,MR
    PRINT 103 ,NR,MSTART,MSTOP, JAME,NAME, MR
    PRINT 104 ,NDI,ND2,NAME,MR
    PRINT 105 *ND4.NAME,MR
    IF(TAG)GO TO 10
    PUNCH 1021,NR,JAME,IVER,MDATE,NAME,MR
    PUNCH 1031,NR,MSTART, MSTOP, JAME, NAME,MR
    PUNCH 1041,NO1,ND2,NAME,MR
    PUNCH 1051 ,ND4.NAME,MR
    GO TO 10
C I/O FORMATS.
80 OFORMAT(*OCANDIDATE*,I4,IX,2A10;* OF **R8;*/**A2,
    1 * FROM DECK **A3;* IS NOT FOR PROGRAM *;AG****/
    2 * CARD*,I2** HAS IDENTIFICATION ERROR.*)
82 OFORMAT (*OVEED VALUES EXCEED PERMISSIBLE RANGE.*//
    1 I5:* EXCEEDS BINARY QUESTION LIMIT OF 59 OR*/
    2 I5:* EXCEEDS OCTAL QUESTION LIMIT OF 14.*/
    3 *OCARD VALUES FOLLOW -*/13X,72I1)
83 OFORMAT(*ONAME AND DATE IN DATA *:AG,5XrR8,/,
    1 * NAME AND DATE IN BLOCK *,A6,5X:R8)
    FORMAT(*O.VO DATA GIVEN FOR *,A6)
85 OFORMAT(*OWARNING -- LOCATION NEED(*,I2,*,1) CONTAINS*
    1 !I2,* -- CHANGED TO*,I2)
        FORMAT()
88 FORMAT (32X,A6,8X,R8,17XPR1)
89 OFORIAT(*ICDC PREPARATION OF MAIN PROGRAM CARDS *
    1 *USING QUESTION AND*/* AND ANSWER SET NAMED *,AG.
    2 * AND DATED **R8.//.10X**ERROR CODES*,15X*1H**
    3 * = VALUE OUT OF RANGE. MINIMUM VALUE SUBSTITUTED.*
    4/36X,*3 = LOW VALUE EXCEEDED HIGH ONE. LOW VALUE *
    5 *REDUCED TO MATCH HIGH ONE.*/)
90 FORMAT(3X,I3,15X,2A10,3X,R8,5X,A2,13X,A3)
91 FORMAT(72R1,A3)
92 OFORMAT(4HONO.,I3.*. SHORT NAVE - *,2AIO** VERSION *AZ
    1 * OATED*,R10/13K,7I10/13X:7(9I1,1H0),2H12/
    2 13HOMAX. VALUES ,72I1/13H HIGH INPUTS ,72RI)
93 FORMAT(13H VALUES USED .72II)
94 FORMAT(13H ERROR COOES .72R1)
95 FORMAT(13HOLON INPUTS P72R1)
101 FORMAT(*OIVAGES OF PUNCHED OUTPUT CARDS -*)
102 OFORMAT(13X,IHCPI4P3X.2A10,10H. VERSION ,A2:GH DATED.1X
    1,R8,17X,A3,I4,1H0)
1021 OFORNAT (IHC.I4,3X.2AIO.1OH. VERSION PA2,GH DATED.IX
    1 (R8.17X:A3.I4.1HG)
103 OFORMAT(19X,12HDATA(M(I)PI=,I3,1H.,I4,1H,.I3,5H)/10H.
    1 A10,4H:10H:A10,13X,A3,I4,1H1)
1031 OFGRIMAT (6X,12HDATA(M(I),I=,I3&1HrpI4pIHP,I3,5H)/10H.
    1 A1O,4H,10:1,A10,13X,A3,I4,1H1)
104 FOFMAT(18X.5Ht ,.020.4HB̈, 020.SHFS.17X,A3.I4.1H2)
104. FORMAT(5%:5:1t :000,4H3, 020:1H5 ,17X:A3,I4&1H2)
```

```
105 FORMAT(18X,5H+ ,020,11HB,0,0, 1/ .31X,A3,I4,1H3)
1051 FORMAT( 5X,5H+ ,0020,11H3,0,0, 1/ ,31X,A3,I4,1H3)
    END
    FUNCTION INEES(ID,ICOL)
C FOR USE WITH CDC CODEM.
C GENERATED 2-14-73. LAST REVISED 10-13-74.
C TO ENCODE AN ARRAY OF ANSWERS INTO A SIMPLE VARIABLE.
C ID=1 FOR LOW(BINARY) VALUES. ID=2 FOR HIGH(OCTAL) VALUES.
C ICOL = COLUMN OF NEED ARRAY TO BE ENCODED.
    COMMON/QA/NQ,NQUES(72),NEED(72,6),KQ(2),KQUES(2,59)
    INEED = 0
    KSHIFT = ID + ID + ID - 2
    ISTOP = KQ(ID)
    DO 10 N=1,ISTOP
    I = KQUES(ID,N)
10 INEED=INEED+ SHIFT(NEED(I,ICOL)-1,(N-1)*KSHIFT)
    RETURN
    END
```


## APPENDIX G

TAPE DECK CHOICE MODEL

A problem in choice of 8-track tape decks'was selected as a simple illustration of the CHOOSE system. The basic data for this problem was extracted from Consumer Reports (1974, pp. 672-675) in which the test results were formulated in a manner in which they could be easily converted to an objective question and answer structure. The report evaluated 8 -track stereo cartridge players for home use. The sequence of steps here follows the program methodology given in detail in Appendix D. The results are the main program PROGRAM TPDEK8 presented in Appendix B.

## CHOOSER's Viewpoint

From the information in this article a question and answer set oriented strictly in terms of a CHOOSER's objectives or aspirations was developed. While it is true that each CHOOSER would, if left to his own resources, develop his own criteria, the situation here is that the EVALUATOR is able to create an appropriate question and answer set in terms of objectives which will meet the needs of most CHOOSERs. Note that the EVALUATOR must pose the answer set in a form which reflects his evaluation of users' value judgments for the item. The question and answer set developed for this CHOOSE problem is given below. The value judgments are noted.

1. I expect to pay (implied value judgment: CHOOSER prefers to pay less):
2. Greater than $\$ 130$
3. $\$ 90-\$ 130$
4. $\$ 65-\$ 90$
5. $\$ 45-\$ 65$
6. $\$ 30-\$ 45$
7. Under $\$ 30$
8. The weight of the unit should be (implied value judgment: lighter is better):
9. Greater than 11 lbs .
10. 8-11 lbs.
11. 6-8 lbs.
12. 4-6 lbs.
13. 2-4 lbs.
14. Less than 2 lbs .
15. The maximum permissible measure in the largest linear dimension is (value judgment: smaller is better):
16. Greater than 14 in.
17. 12-14 in.
18. $10-12$ in.
19. $8-10$ in.
20. 6-8 in.
21. 4-6 in.
22. Less than 4 in.
23. The total volume should not exceed (value judgment: smaller is better) :
24. Greater than 400 cu . in.
25. 300-400 cu. in.
26. 200-300 cu. in.
27. Less than 200 cu . in.

The following seven questions all have the same answers set, which is as follows: (1) poor, (2) fair, (3) good, (4) very good, or (5) excellent. The value judgment here is, of course, that the better choices (higher numbered answers) represent higher aspirations.
5. Freedom from flutter rating:
6. Frequency response evaluation:
7. Signal-to-noise ratio evaluation:
8. Freedom from cross-talk rating:
9. Tape speed accuracy rating:
10. A rating for ability to hold constant speed with variation in line voltage:
11. A desired rating for ease of cartridge removal:

The next four questions are "binary" questions and relate to the presence or absence of certain special features:
12. 1. Switch to disengage cartridge not needed.
2. Must have ejection switch to disengage cartridge.
13. 1. No fast forward or repeat switch desired.
2. Must have fast forward and repeat switch.
14. 1. Option to play 4-channels not required.
2. 4-channel option needed.
15. 1. Automatic ejection switch for end of tape not required. 2. Ejection switch required.

The last question distinguishes between those units which require assembly by the CHOOSER and those which are ready off-the-shelf.
16. For assembly and preparation of the unit, I would be willing to devote (the implied value judgment here is that the shorter assembly preparation time is better) :

1. Greater than 10 hrs .
2. 5-10 hrs.
3. 3-5 hrs.
4. 1-3 hrs.
5. $0-1 \mathrm{hr}$.
6. 0 hrs .

Once the evaluation team has completed the basic question and answer set (or statement and response set), preparation of the basic main PROGRAM TPDEK8, as well as a BLOCK DATA TPDEK8 can proceed. This is done initially by preparing the data for execution with PROGRAM MAKEUP as described in Appendix E. A listing of the cards produced by MAKEUP for both the skeleton PROGRAM TPDEK8 and BLOCK DATA TPDEK8 are given in Figures 23 and 24.

The next step is to evaluate the various candidates within the framework just established.

## The EVALUATOR's Viewpoint

The 16 questions or statements given above are presented from a CHOOSER's viewpoint. As explained in Chapter 5, the evaluation team must, during the first step of the process, prepare questions and answers attempting as much as possible to approach all aspects of the investigation in terms of a potential CHOOSER's objectives. Once these questions have been formulated, as they were above for the TPDEK8 problem, the EVALUATORs must now change their role and pose the questions to the various candidates in a slightly different format. The answer set remains the same. The questions which the EVALUATORs must now address themselves in evaluating the particular units in the above framework would be as follows:

1. What is the delivered price?
2. The weight of the unit is:
3. Of the three normal dimensions (width, length and height), the maximum dimension is:
4. The overall volume of the unit is:
5. Freedom from flutter is rated:
6. Frequency response is rated:
7. Signal-to-noise ratio is rated:
8. Freedom from cross-talk is rated:
9. Tape speed accuracy is rated:
10. The ability to hold constant speed with variation in line voltage is rated:
11. The ease of removing the cartridge is rated:
12. 13. Does not have ejection switch for disengaging cartridge. 2. Has such a switch.
1. 2. Does not have a fast forward and repeat switch. 2. Has such a switch.
1. 2. Does not have ability to play 4-channel cartridges. 2. Has the ability of playing 4-channel cartridges.
1. 2. No automatic ejection switch for end of tape.
1. Has such a switch.
2. Assembly and/or preparation time for the unit is evaluated as:

Once all of the candidates which the EVALUATORs decide should be embedded in the CHOICE problem have been evaluated, the information is entered very simply in computer format for processing of the data, by PROGRAM CODEM. Figure 27 shows a typical set of evaluation cards prepared for the 15 candidates covered in the referenced Consumer Reports article. This data when executed by PROGRAM CODEM (using BLOCK DATA TPDEK8), produced the cards shown in Figure 28. Note that these cards are identified by the 3 -letter code TPD and a 5 -digit serial number in columns 73 through 80. This permits their easy insertion into the skeleton PROGRAM TPDEK8 prepared by MAKEUP described earlier. They can either be inserted by hand or by sorting on columns 75 through 80. Note that columns 73 through 75 are used as identifiers so that cards placed in the wrong deck can easily be identified. Every card in the main PROGRAM TPDEK8: should be punched with TPD in those columns.

## The Complete Main PROGRAM TPDEK8

Appendix B gives the complete main PROGRAM TPDEK8 which is now ready for insertion into the system for call and interaction with the CHOOSE subprograms in the manner described in Appendix A.

Note that the PROGRAM TPDEK8 can usually be updated by (1) removing the four cards associated with any candidate that is no longer available, (2) inserting new candidate cards in the appropriate place, or (3) replacing candidate cards with new evaluations when required.


| BSR TOBS | $10-20-74$ |
| :--- | :--- |
| Fig. 27. Typical Input Set for CODEM with TPDEK8 |  |

Fig. 27. Typical Input Set for CODEM with TPdeks

Any change of the candidate set should also cause card number TPD30003 to be updated. Note that PROGRAM CODEM is designed to produce an updated card for this use.

Changes to the question and answer set pose a more difficult problem in that each of the candidates should be reviewed against the revised question and answer set before the main program can be properly updated. This requires going back to PROGRAM MAKEUP and essentially preparing new information from the beginning.

## REFERENCES

al-Ghazali. "The Deliverer from Error," ca. 1100, cited by Wilson B. Bishai, Humanities in the Arabic-Islamic World, Dubuque: Wm. C. Brown Company, 1973, p. 57.

Åström, K. J. and P. Eykhoff. "System Identification, A Survey," Identification and Process Parameter Estimation, preprints of the Second Prague IFAC Symposium, Czechoslovakia, International Federation for Automatic Control, June 15-20, 1970, pp. 1-38.

Bartlett, John. Familiar Quotations, 11th ed. Boston: Little, Brown \& Co., 1941, p. 1044.

Beakley, George C. and Ernest G. Chilton. Introduction to Engineering Design and Graphics. New York: Macmillan, 1973.

Benayoun, R., Oleg I. Larichev, J. de Montgolfier and J. Tergny. "Linear Programming with Multiple Objective Functions: Step Method (Stem)," 7th Mathematical Programming Symposium, The Hague - The Netherlands, September 14-18, 1970.
. "Linear Programming with Multiple Objective Function: The Method of Constraints," translated from Russian, Avtomatika i Telemekhanika, No. 8, August 1971. New York: Consultants Bureau, January 1972.

Booth, Taylor L. Sequential Machines and Automata Theory. New York: John Wiley \& Sons, Inc., 1967.

Boyd, Dean Weldon. "A Methodology for Analyzing Decision Problems Involving Complex Preference Assessments," Ph.D. dissertation, Stanford University, May 1970.

Churchman, C. West, Russell L. Ackoff and E. Leonard Arnoff. Introduction to Operations Research. New York: John Wiley \& Sons, Inc., 1957.

Cochrane, James L. and Milan Zeleny, ed. Multiple Criteria Decision Making. Columbia: The University of South Carolina Press, 1973.

Consumer Reports, Vol. 39, No. 9 (September 1974).

Dracup, John A., Fereidoun Mobasheri and Miguel A. Cardenas. "An Assessment of Optimization Techniques as Applied to Water Resources Systems," a report, Environment Dynamics, Inc., Los Angeles, August 1970.

Dyer, James S. "A Time-Sharing Computer Program for the Solution of the Multiple Criteria Problem," Management Science, Vol. 19, No. 12 (August 1973), pp. 1379-1383.

Eilon, Samue1. "Goals and Constraints in Decision-Making," Operational Research Quarterly, Vo1. 23, No. 1 (March 1972), pp. 3-15.

Emsellem, Y. and G. de Marsily. "Présentation d'une méthode nouvelle de resolution du 'problème inverse'," Ecole Nationale Supérieure des Mines de Paris, Centre d'Informatique Géologique, May 1969.

Feinberg, A. "An Experimental Investigation of an Interactive Approach for Multi-Criterion Optimization with an Application to Academic Resource Allocation," Working Paper No. 164, Western Management Science Institute, University of California, Los Angeles, March 1972.

Fiering, Myron B. Streamflow Synthesis. Cambridge: Harvard University Press, 1967.

Fitts, Paul M. "Functions of Man in Complex Systems," Aerospace Engineering, Vol. 21, No. 1 (January 1962), pp. 33.1-33.6.

Fogel, Martin M., Lucien Duckstein and Jerry L. Sanders. "An EventBased Stochastic Model of Areal Rainfall and Runoff," presented at the Symposium on Statistical Hydrology, Tucson, Arizona, August 31, September 1-2, 1971.

Food and Agriculture Organization of the United Nations. Progress Report on the Use of Hydrologic Models in FAO Field Projects, 1972.

Gcoffrion, A. "Solving Bicriterion Mathematical Programs," Operations Research, Vol. 15, No. 1 (1967), pp. 39-54.

Hall, John R., Jr. "Against a Universal Model Methodology," presented at 40th National Meeting of the Operations Research Society of America, Anaheim, California, October 25-29, 1971.

Hammer, Preston C. "Approximation Spaces," in Advances in Mathematical Systems Theory, Preston C. Hammer, ed. University Park: Pennsylvania State University Press, 1969, pp. 121-133.

Highet, Gilbert. "The Trouble with Half-Truths," This Week Magazine, February 17, 1963.

Ibbitt, R. P. "Representative Data Sets for Comparative Testing of Conceptual Catchment Models," internal correspondence, UNESCO Working Group on Representative and Experimental Basins, September 1972:

Johnsen, Erik. Studies in Multiobjective Decision Models. Lund, Sweden: Studentlitteratur, 1968.

Johnston, W. B. "Human Behaviour in Resource Use and Management," Part 2 of the Proceedings of a Symposium on Watershed Management in Water Resources Development, a Research Publication of the New Zealand Agricultural Engineering Institute, Lincoln College, Canterbury, New Zealand, No. 9, December 1969.

Kashyap, R. L. "The Determination and Validation of Stochastic Difference Equation Models," presented at IEEE Decision \& Control Conference, Miami Beach, Florida, December 1971.

Kates, R. W. "Hazard and Choice Perception in Flood Plain Management," University of Chicago, Dept. of Geography Research Paper No. 78, Chicago, 1962.

Kemeny, John G. "Mathematical Models and the Computer," Pi Mu Epsilon Journal, Vol. 5, No. 8 (Spring 1973), pp. 373-386.

Kisiel, Chester C. and Lucien Duckstein. "Economics of Hydrologic Modelling," presented at International Symposium on Mathematical Modelling Techniques in Water Resource Systems, Ottawa, Canada, December 1971.
—_ Hasan K. Qashu and Lucien Duckstein. "Decision-Theoretic Aspects of Model Building, Use and Validation in Hydrology and Water Resources," a research proposal to Office of Water Resources Research, U. S. Department of the Interior; Hydrology and Water Resources Office, University of Arizona, Tucson, Arizona, September 25, 1970.

K1ir, George J., ed. Trends in General Systems Theory. New York: Wiley-Interscience, 1972.

MacCrimmon, Kenneth R. "An Overview of Multiple Objective Decision Making," in James L. Cochrane and Milan Zeleny, ed., Multiple Criteria Decision Making. Columbia: The University of South Carolina Press, 1973.

Monarchi, David Edward. "An Interactive Algorithm for Multipleobjective Decision Making," Ph.D. dissertation, University of Arizona, Tucson, 1972.

Mylks, Herbert William. "A Planning Methodology for Comprehensive Health Planning," Master's thesis, University of Arizona, Tucson, 1971.

Popkin, Barney Pau1. "The State of Research and Application of Stochastic Methods in Hydrology," for presentation at the International Union of Geophysics, Contress, Grenoble, France,
$\therefore$ :...:- August 1975.
Pritsker, A. Alan B. and Philip K. Kiviat. Simulation with GASP II: A Fortran Based Simulation Language. Englewood Cliffs: Prentice-Hall, Inc., 1969.

Rosenblueth, Arturo and Norbert Wiener. "The Role of Models in Science," Philosophy of Science, Vol. XII, No. 4 (October 1945), pp. 316-321.

Roy, Bernard. "Problems and Methods with Multiple Objective Functions," Mathematical Programming I. The Netherlands: NorthHolland Publishing Co., 1971, pp. 239-266. and P. Bertier. "La méthode Électre II: Une application au média-planning," a report by Metra International, 35, boulevard Brune, Paris 143, June 1972.

Sagar, Budhi. "Calibration and Validation of Aquifer Models," Hydrology and Water Resources Office Report No. 17, University of Arizona, Tucson, 1973.

Salton, G. "Automatic Text Analysis," Science, Vol. 168 (April 17, 1970), pp. 335-343.

Saucedo, Roberto and Earl E. Schiring. Introduction to Continuous. and Digital Control Systems. New York: Macmillan, 1968.

Shakespeare, William. The Tragedy of Hamlet, Prince of Denmark, Act Second, Scene II, lines 255-257, 1603, appearing in Tucker Brooke, John William Cunliffe and Henry Noble MacCracken, Shakespeare's Principal Plays, Third Ed., New York: D. Apple-ton-Century Co., 1940, p. 535.

Sheridan, Thomas B. and William R. Ferrell. Man-Machine Systems: Information, Control, and Decision Models of Human Performance. Cambridge: The MIT Press, 1974.

Slevin, Dennis P. and Randall L. Schultz. "The Implementation of OR/MS Models: Theory Research and Application," proposal for a Conference at the Graduate School of Business, University of Pittsburgh, September 1972.

Smailwood, Richard D. "A Decision Analysis of Model Selection," IEEE Transactions on Systems Science and Cybernetics, Vol. SSC-4, No. 3 (Sep'tember 1968), pp. 333-341.

Time. "What Went Wrong," Vol. 102, No. 24 (December 10, 1973), pp. 49-50.

Toffler, Alvin. Future Shock. New York: Random House, Inc., 1970.
Torgerson, W. S. Theory and Methods of Scaling. New York: John Wiley \& Sons, Inc., 1958.
U. S. News and World Report, Vol. 76, No. 25 (June 24, 1974), pp. 86-87.

Viessman, Warren Jr. "Economic and Technical Aspects of the Use of Mathematical Models in State Water Resources Planning Programs," Publication No. 3, Water Resources Research Institute, University of Nebraska, Lincoln, September 1970.

Wiener, Norbert. Cybernetics. New York: John Wiley \& Sons, Inc., 1948.

Wilde, Douglass J. and Charles S. Beightler. Foundations of Optimization: Englewood Cliffs: Prentice-Hall, Inc., 1967.

Woolsey, R. E. D. "Operations Research and Management Science Today," Operations Research, Vo1. 20, No. 3 (June 1972), pp. 729-737.

World Meteorological Organization, "Report of the Meeting of Experts on Intercomparison of Conceptual Models for Purposes of Hydrological Forecasting," Geneva, Switzerland, September 23-26, 1972.

Wymore, A. Wayne. A Mathematical Theory of Systems Engineering: The Elements. New York: John Wiley and Sons, 1967.
. A Notebook of Systems Engineering Methodology. Tucson:
A. Wayne Wymore, The University of Arizona, 1970.
. "The Role of Models in Research," unpublished paper, Department of Systems and Industrial Engineering, The University of Arizona, Tucson, 1971.
. "A Wattled Theory of Systems," in George J. Klir, Trends in General Systems Theory. New York: Wiley-Interscience, 1972. - Systems Engineering Methodology for Interdisciplinary Teams. Tucson: A. Wayne Wymore, The University of Arizona, 1973.

Xerox Corporation advertisement appearing in Time, Vol. 98, No. 23 (December 6, 1971).

Yakowitz, Sidney and John L. Denny, Jr. "On the Statistics of Hydrologic Time Series," unpublished report, University of Arizona,
$\therefore \approx=\therefore . .$. Tucson, 1973. $\therefore \because$
Young, Dennis R. "Choosing Among Alternative Complex Systems when Input Characteristics are Uncertain," IEEE Transactions on
$\therefore \therefore \quad$ Systems, Man and Cybernetics, Vo1. SMC-1, No. 1 (January 1971), pp. 77-82.

Zadeh, Lofti A. "Fuzzy Sets and Systems," Proceedings of the Symposium on System Theory, Microwave Research Institute Symposia Series,
$\because こ \because$. Vol. XV. Brooklyn: Polytechnic Press of the Polytechnic Institute of Brooklyn, 1965.

Zadeh, Lofti A. "Similarity Relations and Fuzzy Orderings," Information Sciences, Vol. 3 (1971), pp. 177-200.

Zeitler, Eddie Lorenz. "A Method for Developing Information Retrieval. Systems for Small Libraries," Master's thesis, University of Arizona; Tucson, 1970.

