WORKING PAPER

DISCRET: AN INTERACTIVE DECISION SUPPORT SYSTEM FOR DISCRETE ALTERNATIVES MULTICRITERIA PROBLEMS

J. Majchrazak

December 1988
WP-88-113

IIASA
International Institute
for Applied Systems Analysis

DISCRET: AN INTERACTIVE DECISION SUPPORT SYSTEM FOR DISCRETE ALTERNATIVES MULTICRITERIA PROBLEMS

J. Majchrazak

December 1988
WP-88-113

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Foreword

This paper is one of the series of 11 Working Papers presenting the software for interactive decision support and software tools for developing decision support systems. These products constitute the outcome of the contracted study agreement between the System and Decision Sciences Program at IIASA and several Polish scientific institutions. The theoretical part of these results is presented in the IIASA Working Paper WP-88-071 entitled *Theory, Software and Testing Examples in Decision Support Systems* which contains the theoretical and methodological bacgrounds of the software systems developed within the project.

This paper presents the DISCRET system. This system has been designed to solve basic multicriteria choice problems in which a finite set of feasible alternatives is explicitly given and for each alternative the value of all criteria are known. The decision maker is assumed to be rational in the sense that he can accept a Pareto-optimal solution as his final solution of the problem.

Such a decision problem is rather simple as long as the number of criteria and alternatives is small. However, if the number of alternatives and/or criteria grows, the human information processing capabilities may reach their limits and therefore decision support facilities need to be utilized to guarantee efficient decision making.

Alexander B. Kurzhanski
Chairman
System and Decision Sciences Program

DISCRET
An Interactive Decision Support System for Discrete Alternatives Multicriteria Problems

Janusz Majchrzak

Systems Research Institute, Polish Academy of Sciences, Warsaw.

Contents

Part I
Executive summary

1 General description

DISCRET is a system created to solve basic multicriteria choice problems in which a finite set of feasible alternatives (and decisions) is explicitly given and for each alternative the value of all criteria describing its attributes interesting to the decision maker (DM) were evaluated and listed. The DM is assumed to be rational in the sense that he is looking for a Pareto-optimal solution as his final solution of the problem.

Such a decision problem is rather trivial for any human being as long as the number of criteria and alternatives is small (say, 3–5 criteria, 7–15 alternatives) However, if the number of alternatives and/or criteria grows, the limits of human information processing capabilities are reached and some decision support facilities have to be utilized to guarantee a proper and efficient decision making.

In many real–life problems the decision variables take their values from a discrete set rather than from a continuous interval. Usually there is a finite number of available facility location sites, the facility size or production capability may be chosen from a discrete set of values, during a design process the components are chosen from a set of typical elements available on the market, etc. Such problems form the "natural" field of applications for the DISCRET system.

Another field of possible applications of the DISCRET system consists of cases in which the original problem is actually continuous (rather than discrete) but the analysis restricted just to a finite number of alternatives appearing in this problem may be interesting and useful for the DM, since it may result in an enlightening and a more precise definition of his preferences, region of interest or aspiration levels.

Situations falling under the latter category may occur for at least two following reasons. Firstly, a sample of alternatives together with the corresponding criteria values may be readily available (from simulation model runs for example). Secondly, for the purpose of an initial analysis the DM may take into considerations just a few values for each decision variable or generate a random sample of alternatives.

The DISCRET system makes no restrictions on the forms of the criteria. Therefore, attributes as complicated as required may be considered.

To start the session with DISCRET the DM has to supply the file containing set of the criteria values for all feasible alternatives, the problem specification file and (optionally) the file containing the set of feasible decisions. These files, called the data, specification and the additional data file respectively, describe the problem under consideration.

After loading the problem the DM may obtain the information about the criteria values ranges and he may put the lower and/or upper bounds on the values of some/all criteria. The bounds setting may be utilized to eliminate irrelevant alternatives from further considerations or to specify the current region of interest in the objective space.

In the next step the DM may eliminate the dominated alternatives by an explicit enumeration technique. The tolerances for criteria values play an important role here. If they are all equal zero or have small positive values that correspond to indifference limits of the DM's for criteria values, the whole set of the nondominated solutions will be obtained. If the values of tolerances are equal to some significant fractions of the corresponding criteria ranges, then a representation of the set of nondominated solutions will be obtained. The representation is a subset of the set of nondominated solutions preserving its shape and containing the smaller number of elements, the larger were chosen tolerance coefficients.

The biggest advantage of the implemented enumeration method is its ability to select a representation of the nondominated set instead of the entire set. Unlike other known approaches which find the entire nondominated set first and then select a representation (differently defined for each of those methods), the presented method selects a representation at once. This fact profits in efficiency. Observe that because the representation contains less elements than the nondominated set, it will be obtained with a smaller computational effort.

The powerful tool of the reference point approach is also available for the DM. By determining a reference point, he exhibits his aspiration levels for criteria values, confronts them with the obtained solution and modifies them and the reference point. To learn more about the problem the graphic display of two–dimensional subproblems on the terminal screen can be utilized. The DM chooses two criteria for the vertical and horizontal axes, while the other criteria are restrictively bounded — a two–dimensional "slice" is cut out of the original m–dimensional problem. The graphical displays are very useful on this stage of the decision making process, since the DM can see clusters (groups) of alternatives.

DISCRET is an interactive system. The DM may execute its commands in any order. The variety of paths the DM may follow guarantees flexibility in meeting his demands. The implemented approach seems to be easy to understand and approve even for a user who is not very familiar with multicriteria optimization techniques.

2 Short program description

The interactive decision support system DISCRET has been designed to solve medium–size discrete multicriteria choice problems with the number of alternatives ranging from few hundreds to few thousands. The number of criteria is in the current version restricted to 20 (mainly due to the limitations of display facilities).

The program is recorded on a diskette(s) and should be installed on an IBM–PC–XT/AT or a compatible computer with a Color Graphic Adapter, Enhanced Graphic Adapter or Hercules Graphic Card and a hard disk. The compiled code is distributed together with a number of files it requires and with two test problems generators providing demonstrative examples.

The system supports the following menu–controlled general functions:

- Loading user's problems in an easy to prepare standard of an ASCII file form.
- Criteria values ranges (utopia and nadir points) display, and new criteria values bounds setting to define the user's current region of interest.
- Solving the discrete multicriteria optimization problem with explicit alternatives (implicit constraints), i.e. finding the set of nondominated or weakly nondominated elements or it's representation, keeping or rejecting duplicate elements.
- The reference point approach, i.e. selection of nondominated alternatives that correspond to user-supplied aspiration levels for criteria.
- Graphic display of the two–dimensional "slices" of the problem showing the user alternatives clusters (groups).

Part II

Methodological Guide

3 Introduction

3.1 Scope of the report

This report aims to:

- provide the information necessary to use the DISCRET package and to understand its structure as well as the capabilities of the implemented approach,
- discuss such methodological issues associated with the implemented approach, which might be interesting for the user and which justify the chosen approach,
- attract and encourage the reader to take the advantage of the package utilization,

It is assumed that the reader and the package user possess just the very basic information about multicriteria optimization and discrete choice problems.

3.2 Purpose of the DISCRET package

DISCRET is a package created to solve basic multicriteria choice problems in which a finite set of feasible alternatives (and decisions) is explicitly given and, for each alternative, the values of all criteria describing its attributes interesting to the decision maker (DM) were evaluated and listed. The DM is assumed to be rational in the sense that he is looking for an efficient (Pareto-optimal) solution as his final solution of the problem.

Such a discrete multicriteria optimization problem is rather a problem of choice than optimization, since all the information necessary to make a decision is readily available. Such a problem is rather trivial for any human being as long as the number of alternatives is small (say, less than ten or twenty). However, if the number of alternatives and/or criteria grows, the limits of human information processing capabilities are reached and some decision support facilities have to be utilized to guarantee a proper and efficient decision making.

The purpose of the DISCRET package is to support the DM in his search for final decision in an interactive and user-friendly manner. It is assumed that the DM has only a limited knowledge of the problem he wants to solve at the beginning of the session with DISCRET. Therefore, during the session no difficult questions are asked (for example, about criteria trade-offs, DM' utility function or pairwise comparisons of alternatives). The package-provided information enables the DM to gather the experience related to his problem's specific features as well as his own preferences.

The implemented approach seems to be easy to understand and approve even for a user who is not very familiar with multicriteria optimization techniques.

The DISCRET package has been designed to solve medium-size discrete multicriteria problems with the number of alternatives ranging from few hundreds to few thousands. The number of criteria is in the current version restricted to 20 (mainly due to the limitations of display facilities).

During the session the user controls the decision-making process by choosing suitable options from the displayed "menu". Therefore, he does not have to learn and remember any command pseudo-language. This feature, together with special procedures for handling user's mistakes and with self-explanatory package messages, makes the package user-friendly and allows for an unexperienced user.

3.3 Fields of the package applications

In many real-life problems, decision variables take their values from a discrete set rather than from a continuous interval. Usually, there is a finite number of available facility location sites, the facility size or production capability may be chosen from a discrete set of values, during a design process the components are chosen from a set of typical elements available on the market, etc. Such problems form the "natural" field of applications for the DISCRET package.

Another field of possible applications of the DISCRET package consists of cases in which the original problem is actually continuous (rather than discrete) but the analysis restricted just to a finite number of alternatives appearing in this problem may be interesting and useful for the DM, since it may result in an enlightening and a more precise definition of his preferences, region of interest or aspiration levels.

Situations falling under the latter category may occur for at least two following reasons. Firstly, if a sample of alternatives together with the corresponding criteria values is readily available, the utilization of the DISCRET package may enable the DM to gain an insight into the original multicriteria problem. The analysis of an assembly of runs of a simulation model is an example of this case. Secondly, for the purpose of an initial analysis of a problem in which the decision variables actually take their values from continuous intervals, the DM may take into consideration just a few values for each decision variable or to generate a random sample of alternatives.

An encouraging factor that may attract the DM is the fact that the DISCRET package makes no restrictions on the forms of the criteria. Therefore, attributes as complicated as required may be considered.

4 Background

4.1 The discrete multicriteria optimization problem

Package DISCRET has been created to support—in an interactive manner—multicriteria optimization and decision making for problems with a finite number of discrete alternatives. Such problems are frequently referred to as implicit constraints or explicit alternatives problems.

Let us consider the following discrete multicriteria optimization problem (DMOP). It is assumed that a set X^0 of feasible discrete alternatives is explicitly given and for each of its elements all criteria under consideration have been evaluated. The criteria values for all feasible discrete alternatives form the set Q of feasible outcomes or evaluation.

$$\min_{x \in X^0} f(x)$$

$$X^0 = \{x_1, x_2, \ldots, x_n\} \subset X = R^s$$

$$f(x) = (f^1(x), f^2(x), \ldots, f^m(x))$$

$$f : X^0 \to Q$$

$$Q = \{f_1, f_2, \ldots, f_n\} \subset F = R^m$$

$$f_j = f(x_j), \quad j = 1, 2, \ldots, n$$

Furthermore, it is assumed that a domination cone Λ is defined in the objective space F. As in most applications the positive orthant is considered, $\Lambda = R_+^m$ and $\tilde{\Lambda} = R_+^m \setminus \{0\}$. The domination cone introduces the partial pre-order relation "$\prec$" into the objective space:

$$\forall f_1, f_2 \in F, \quad f_1 \prec f_2 \iff f_1 \in f_2 - \tilde{\Lambda}$$

The element f_1 dominates f_2 in the sense of the partial pre-order induced by the domination cone Λ.

Element $\bar{f} \in Q$ is nondominated in the set of feasible elements Q, if it is not dominated by any other feasible element. Let $N = N(Q) \subset Q$ denote the set of all nondominated outcomes in the objective space and let $N_X = N(X^0) \subset X^0$ denote the set of the corresponding nondominated alternatives (decisions) in the decision space. To solve the DMOP it means to find the set N of nondominated outcomes and the corresponding set N_X of nondominated decisions.

Notice that DMOP is described by the two sets Q and X^0 defined above (together with m, n and s). Therefore the package input files supplied by the user must contain these two sets.

Observe also that no assumptions were made about the nature of the criteria functions f_i. In fact, the only requirement for them is that they should assign numerical values to the alternatives, indicating their attractiveness with respect to the attribute under consideration. In particular, the criteria functions may be of the qualitative type. The single restriction is that values assigned to alternatives by criteria should be expressed by numbers and that the user is able to indicate whether he wishes to increase or decrease these numbers. In doing so, he defines or changes the domination cone Λ.

Observe also that the above abstract definition of a solution to DMOP is not very practical: the set N of all nondominated outcomes might be very large and difficult to compute, and its full computation might be useless if the user decides to change the domination cone Λ. Therefore, an important issue is to find some representation of the set N, not the entire set.

4.2 Overview of existing approaches

The discrete multicriteria optimization problem (DMOP) is a combinatorial problem involving sorting and one could expect a large number of papers in the bibliography devoted to this subject. However, the problem did not focus much attention of the researchers—except in its utility theoretical variant that actually transforms the problem to a single–criteria one—and the bibliography we are able to point at consists only of (Kung et al., 1975, Polak and Payne, 1976, Stahn and Petersohn, 1978), plus some reports of the earlier research summarized there.

The insignificant interest in methods for solving DMOP could be explained by the fact that the solution of the DMOP, the whole set of nondominated alternatives is not the solution of the multicriteria decision making problem (MCDMP), a selected preferred alternative. However, since the efficiency of methods dealing with MCDMP usually depend on the number of alternatives, it is wise to reject the dominated alternatives.

A rather large number of approaches have been suggested for the solution of the MCDMP involving discrete alternatives. They differ both in the problem formulation and the assumptions about the decision maker (DM). Let us mention here just some most interesting ones.

The method suggested in (Keeney and Raiffa, 1976) is based on utility functions constructed first for each criterion and then combined into a global utility function. In (Zionts, 1981) a linear, while in (Köksalan et al., 1984) a quasiconvex underlying utility function of the DM is assumed and the best alternative according to an approximation of this utility function is found by asking for answers to a number of comparisons between pairs of alternatives. Other methods, e.g. (Roy, 1971) or (Siskos, 1982), are based on outranking relations. In (Rivett, 1977), multidimensional scaling techniques are used to obtain a graph pointing from least to most preferred alternatives.

Other group of approaches (some of them were proposed originally for some different problems) is based on an observation that if the number of the alternatives is small, then the DM is able to make a decision intuitively, without any formalism of expressing his preferences. If the number of alternatives is larger, then one has to reduce it for the DM by selecting a small but representative sample. Several methods for obtaining such a representation were proposed. They utilize cluster analysis (Törn, 1980, Morse, 1980), filtering (Steuer and Harris, 1980) or random sampling (Baum et al., 1981).

Approaches from the first of the two above-mentioned groups place the burden on the DM. He is asked to supply the information about his preferences by the evaluation of the alternatives —by pairwise comparisons or rankings for example. These evaluations are substantial for the methods. Each of these methods is based on certain implicit or explicit assumptions about the DM, such that, for example, he has an utility function expressing his preferences. The size of the problems that can be solved is limited by the DM's ability to provide the required amount of information by ranking or comparing pairwise the alternatives.

In the approaches from the second group, the burden is placed rather on the computer. The crucial point here is whether the obtained representation of the nondominated set will be illustrative for the DM. No special assumptions about the DM are made. He is only expected to prefer the nondominated alternatives rather than the dominated ones.

Our approach presented in this report may be classified as one of the second group. It is based on a new efficient method for DMOP, which also can efficiently produce a representation of the nondominated set.

4.3 The method of dominated approximations

The implemented method is of the explicit enumeration type. It is called the method of dominated approximations and is based on the following concept.

Def. 1 Let Q be the set of all feasible alternative outcomes, N the set of corresponding nondominated alternative outcomes and Λ the domination cone. Set A is called a dominated approximation of N iff

$$N \subset A - \Lambda$$

In other words, A is a dominated approximation of N iff for each $f_i \in N$ there exists $f_j \in A$ such, that $f_i \prec f_j$ in the sense of the partial pre-order induced by Λ.

We will say that the approximation A_2 dominates the approximation A_1 of the nondominated set N iff

$$A_1 \subset A_2 + \Lambda$$

Hence, as the worst approximation of N we can consider the entire set Q, while the best approximation is the set N itself. The method of dominated approximations generates a sequence of approximations A_k, $k = 0, 1, 2, \ldots, l$ such that

$$Q = A_0 \supset A_1 \supset A_2 \supset \cdots \supset A_k \supset \cdots \supset A_l = N$$

Thus, given Q and Λ we are supposed to determine $N = N(Q)$. Assume that all criteria are to be minimized.

Step_0 Let $A_0 = Q,\ N_0 = \emptyset,\ k = 0$.

Step_1 If $A_k \setminus N_k = \emptyset$ then STOP with $N_k = N$, else choose any index $i \in I = \{1, 2, \ldots, m\}$ and find $\bar{f} \in Q$ such that the i–th component of it is minimal in $A_k \setminus N_k$:

$$\bar{f}^i = \min_{A_k \setminus N_k} f^i$$

(See Remark 2).
Set $N_{k+1} = N_k \cup \{\bar{f}\}$.

Step_2 Create the new approximation A_{k+1} by rejecting from $A_k \setminus N_{k+1}$ all elements dominated by $\bar{f}$ (see Remark 1)

$$A_{k+1} = \Big(\{A_k \setminus N_{k+1}\} \setminus \{(\bar{f} + \tilde{\Lambda}) \cap (A_k \setminus N_{k+1})\}\Big) \cup N_{k+1}$$

Set $k = k + 1$ and go to Step_1.

Remark 1. While rejecting the elements dominated by $\bar{f}$ it is sufficient to compare elements of the set $A_k \setminus N_{k+1}$ with $\bar{f}$ according to all but i-th criterion, since $\bar{f}^i$ is minimal among all f^i in $A_k \setminus N$.

Remark 2. The minimum may happen to be non-unique. Let B be the set of those elements $f_j \in A_k \setminus N_k$ for which f^i_j appear to be minimal in $A_k \setminus N_k$. Actually not all elements of B are nondominated. One has to solve the following problem. Given B and Λ select $N(B)$. The above presented method may be used for this task with $A_0 = B$ and $I = I \setminus \{i\}$. This recurrence is applied until an unique minimum is found in the Step_1 of the algorithm. Then, after the execution of Step_2 one has to return to the lower level of recurrence. On each level *Remark 1* holds.

Note that if the recursion described in *Remark 2* would not be applied, then the set of weakly nondominated alternatives would be determined by the above algorithm.

In order to measure the efficiency of the method, let us consider the number of scalar comparisons $S(m, n, p)$ required by the method to solve the DMOP with m criteria, n feasible alternatives and p nondominated alternatives. From the analysis of the method one can easily obtain

$$S(m, n, p) \leq \frac{mp}{2}(2n - p - 1)$$

As one can see, the method solves easily problems with small p. In practical problems p is usually a small fraction of n; the worst case is for $p = n$, i.e. when all alternatives are nondominated. Note that the performance of the method does not depend on the permutation of the alternatives.

4.4 Selection of a representation of the nondominated set

The biggest advantage of the method of dominated approximations is its ability to select a representation of the nondominated set N instead of the entire set N. Unlike other known approaches which find the entire nondominated set first and then select a representation (differently defined for each of those methods), the presented method selects a representation at once. This fact provides much gain in algorithmic efficiency.

Let t_i, $i = 1, \ldots, m$ be some given tolerance coefficient for the m criteria under consideration, $t_i \geq 0$, and $T_i = (t_1, t_2, \ldots, t_{i-1}, 0, t_{i+1}, \ldots, t_m)$ be a vector in the objective space. For the sake of simplicity let us assume that all criteria are to be minimized. The following modification of the method of dominated approximations suffices to obtain a representation instead of the whole nondominated set. In the Step_2 of the method not only the elements dominated by the nondominated element $\bar{f}$ (found in the Step_1 by minimization over the i-th criterion values) have to be rejected, but also elements dominated by $f = \bar{f} - T_i$. Hence, in Step_2 is modified to:

$$A_{k+1} = \Big(\{A_k \setminus N_{k+1}\} \setminus \{(\bar{f} - T_i + \tilde{\Lambda}) \cap (A_k \setminus N_{k+1})\}\Big) \cup N_{k+1}$$

Observe that because the representation contains less elements than the nondominated set, it will be obtained with a smaller computational effort. Figure 1 illustrates the role of the tolerance coefficients in the process of selecting a representation.

The author is not aware of the existence of any other methods that could be effectively applied for a problem with few hundreds or few thousands of alternatives.

4.5 Outline of the approach and introduction to DISCRET

To start the session with DISCRET the user has to supply the file containing set Q of the criteria values for all feasible alternatives, the file containing some problem and data specifications and (optionally) the file containing the set X^0 of feasible decisions (the *load* command). These files, called the data, the specification and the additional data file respectively, describe the problem under consideration.

After the problem generation and implementation phase the user may obtain the information about the criteria values ranges and he may put the lower and/or upper bounds on the values of some/all criteria (the *bounds* command).

The bounds setting may be utilized by the user for several purposes. This is the list of some most relevant:

- to eliminate irrelevant alternatives from further considerations,
- to specify his current region of interest in the objective space,
- to redefine his problem as a problem with a fewer criteria as the original one (as in the method of equality/inequality constraints—see Lin, 1976), for example, a bicriteria problem.

In the next step the user may run the DMOP solver (by executing the command *solve*) to eliminate the dominated alternatives by an explicit enumeration technique. The tolerances for criteria values play an important role here. If they are all equal zero or have small positive values that correspond to indifference limits of the DM's for criteria values, the whole set of the nondominated solutions will be obtained. If the values of tolerances are equal to some significant fractions of the corresponding criteria ranges, then a representation of the set of nondominated solutions will be obtained. The representation is a subset of the set of nondominated solutions preserving its shape and containing the smaller number of elements, the larger were chosen tolerance coefficients.

After the nondominated set or its representation has been obtained, the user may proceed in one of the following paths:

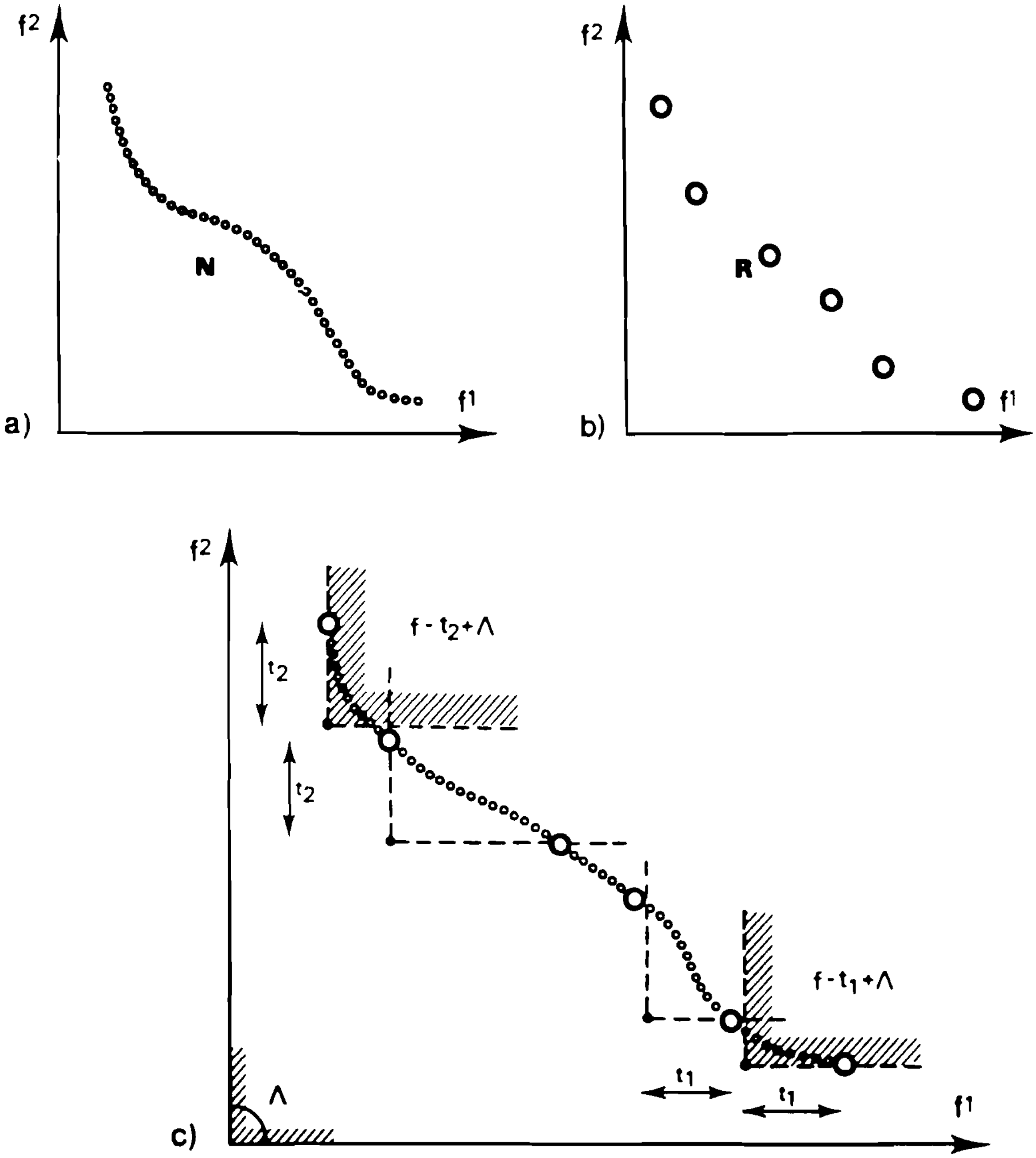

Figure 1: Selection of the representation $R = R(N) = R(Q)$ of the nondominated set $N = N(Q)$. Only nondominated elements are marked for the sake of simplicity of illustration.

a) the nondominated set N.

b) the representation R of the set N.

c) illustration of the tolerance mechanism.

- choose a new region of his interest by a proper bounds setting (by using the command ***bounds***),
- obtain a more or less dense representation by decreasing or increasing the tolerances (by using the commands *solve*),
- use graphic display to learn more about the problem and utilize the reference point approach (by using the command ***analyse***).

It is worth to mention here that—unlike in other known techniques of obtaining a representation of the nondominated set—our approach not only does not require any additional computational effort but even decreases the time of computation with the ratio of $\#R$ to $\#N$, where $\#N$ and $\#R$ are the number of elements in the nondominated set N and its representation R, respectively.

Once any subset of the set N of nondominated solutions has been obtained, one can select the corresponding decisions from the additional data file (the command ***pick***).

The DISCRET package provides also some more detailed information about the problem under consideration. A nondominated and a dominated linear approximations of the set of nondominated solutions are calculated (the command ***analyse***). These approximations are obtained in the following way. A linear function is defined by the combination of the criteria with coefficients determined by the criteria ranges. This function is then minimized and maximized over the set of nondominated elements to obtain the nondominated and dominated approximation, respectively.

The information contained in the lower and upper bounds for criteria, in criteria ranges and in nondominated and dominated approximations gives a good overview of the shape of nondominated set. To learn more about the variety of available alternatives, the user may use another facility provided by the DISCRET package (in the command ***analyse***), namely the graphical display of two-dimensional subproblems on the terminal screen. The user chooses two criteria for the vertical and horizontal axes, while the other criteria are:

- left unbounded—the whole problem is projected on the two-dimensional subspace of the space of objectives, just as if all but the two selected criteria were ignored,
- restrictively bounded—a two-dimensional "slice" is cut out of the original m-dimensional problem.

Enlargements of the chosen display fragments may be obtained simply by specifying new bounds for the criteria on the axes. Another display feature indicates how many elements does each of the 800 display points represent. This feature may be useful to detect and investigate the cluster structure of the problem.

The powerful tool of the reference point approach (Wierzbicki, 1979) is also available for the user (in the command ***analyse***). By determining a reference point, he exhibits his aspiration levels for criteria values, confronts them with the obtained solution and modifies them and the reference point. The graphical displays mentioned above could also be useful on this stage of the decision making process.

During a session with DISCRET the user does not have to necessarily follow the entire procedure presented above. Once the problem generation and specification phase has been completed, he may utilize the package facilities in any order, repeat some steps (commands) or their sequences.

The ability of ignoring some of the criteria temporarily (by specifying that they are to be neither minimized nor maximized) opens to the DM a possibility of using a lexicographic or group-lexicographic approach. He may also, besides the actual criteria, introduce in an identical way some additional criterion expressing his utility, goal or preference function or any global criterion and use them on any arbitrary chosen stage of the decision making process. Such additional criteria have to be evaluated for each alternative during the problem generation phase (just as in the case of the original criteria).

The package offers also the possibility of an immediate return to any of the previous stages of the session, provided that the user have saved them into files (the *save* and *load* commands).

5 Structure and features of the package

5.1 General description

The current pilot version of the DISCRET package consists of eight FORTRAN77 programs. In order to run any of them the user has to type an appropriate program name (command) on his terminal. A list of DISCRET programs is presented below.

- *test1* — first test problem generator (the Dyer's Engine Selection Problem), a separate program.
- *test2* — second test problem generator (the location-allocation problem), a separate program.
- *load* — loads the problem from the data and specification ASCII files.
- *bounds* — informs about the criteria values ranges (utopia and nadir points), nondominated and dominated approximations of the set of alternatives and supports setting of new bounds on criteria values.
- *solve* — solves the discrete multicriteria optimization problem with explicit alternatives (implicit constraints), i.e. finds the set of nondominated or weakly nondominated elements or its representation, keeping or rejecting duplicate elements.
- *analyse* — supports the reference point approach and simple graphic displays of the nondominated set.
- *save* — saves the problem into the data and specifications ASCII files.
- *sort* — sorts the alternatives in increasing/decreasing order with respect to the values of a specified criterion, a *save* subcommand.
- *pick* — finds decisions corresponding to the chosen outcomes in criteria space, a *save* subcommand.

During the command execution, the user controls the process by choosing suitable items from the displayed menu (a list of options available at the moment). The menu system has been chosen instead of a pseudo-language of control commands because it does not require from the user to learn and remember a set of commands.

Each menu contains an amount of information sufficient to make the decision which of the displayed options is the most suitable one. If the user is asked to enter some information, everything he types is checked. If he makes a mistake, a message is displayed on the screen.

Usually the message not only indicates the error but also shows the correct form of the required input.

In the next chapters the package commands will be briefly presented. We will not go into details of each menu since they are self-explanatory. The user will gather all the necessary experience during an introductory session with DISCRET. The test problems may be created by the commands *test1* and *test2*. The description of the test problems can be found in the user's training manual.

5.2 Problem loading phase

The command *load* loads the problem by reading the data file and the specification file. The user may also utilize it as an "unsave" facility which would allow him to return to any problem previously created and saved during the DISCRET session.

5.3 The bounds setting phase

The command *bounds* reads the input data, evaluates the criteria values ranges and displays them together with the nondominated and dominated approximation of the set of alternatives. If the user is not satisfied with the ranges of criteria values or with the values of approximations he can change them.

Knowing the ranges of criteria values, the DM may decide that some of the values of criteria does not interest him at all or at least temporarily. The command *bounds* makes it possible to change the DM's region of interest. By setting the appropriate lower and/or upper bounds for criteria values, the DM restricts further considerations to a smaller region of the objective space—his current region of interest. Only these alternatives that satisfy the bounds will be contained in the output data file produced by the command *bounds*.

Notice that the command *bounds* can select only a subset of alternatives from the input data. If the DM wants to consider a completely different region of interest, he has to supply the input data file containing that set of alternatives.

To illustrate this point assume, just for the sake of simplicity, that all criteria are to be minimized. Observe that if the decreasing of an upper bound for one criterion results in increase of the lower value for some other criterion, then it indicates that a part of the nondominated set did not satisfy the bounds and was rejected. If this was not the purpose of the user, he should return to less restrictive bounds. This remark may be useful on the initial stage of the problem analysis, when the user should become acquainted with the entire variety of the available alternatives.

5.4 The DMOP solving phase

The command *solve* results in solving the DMOP i.e. it selects the nondominated outcomes out of the set of feasible solutions. If the tolerances for all criteria values are equal to zero or have some small positive values corresponding to the computer arithmetic accuracy (for example, 1.0e–10) or criteria values measurement accuracy, then all nondominated outcomes are found. If the tolerances have larger positive values equal to some significant fractions of the criteria values ranges, then just a subset of the nondominated set, called its representation, is selected.

The command *solve* asks the user also about the type of the solution he is looking for. It has the ability to find either the set of nondominated outcomes or weakly nondominated outcomes. If there are duplicate outcomes (that is, if the same outcome vector corresponds to two different decisions), then they can be treated as distinguished ones (and all preserved) or as identical ones (and all but one rejected). Options more sophisticated than the default option (nondominated outcomes, duplicates rejected) do make sense in the cases when at least for some criteria rough values where initially given and they are supposed to be refined in some next stage of the decision making process, or when some of the criteria are more important then the other.

5.5 The problem saving phase

Once the nondominated set (or its representation or a part of it corresponding to the current region of interest of the user selected by setting of bounds) has been obtained, the user may wish to save it in order to continue the job later or to list its elements and analyse them.

The subcommand *sort* sorts the elements of the input data file according to increasing or decreasing values of criteria chosen by the user. Another option is to sort the alternatives in increasing or decreasing order according to their identifiers. When sorted before being printed, any set of alternatives appears to be more readable and hence more useful for analysis.

The subcommand *pick* selects from the additional input data file any additional information corresponding to the elements contained in the data file. Typically, this additional information describes the decisions leading to the obtained nondominated solutions.

The mechanism provided by the subcommands *sort* and *pick* may be especially useful in the case when the package user is an analyst. Properly sorted data (a nondominated set representation adequate to the current stage of the decision making process) will be more readable for the DM.

5.6 The phase of selecting final solution

The command *analyse* was designed to help the user to define his region of interest in a more precise way or to find his final solution.

At the beginning, the user will be informed about the criteria best and worse values—the utopia and nadir points. In order to provide some more detailed but still aggregated information about the shape of the nondominated set (or its representation or just a part of it) the nondominated and dominated linear approximations are evaluated.

A linear combination of criteria with coefficients proportional to the criteria ranges is minimized and maximized over the nondominated set to obtain its nondominated and dominated approximation respectively. Each of these approximations may be characterized by a single parameter standing for the percentage of the range it cuts off out of each criterion values range, see Figure 2 for illustration. Solutions obtained from the linear approximations are also displayed. This aggregated information seems to provide good aggregate data on the shape of the nondominated set, no matter how many criteria are under considerations.

In order to learn more about the criteria trade-offs, the user may display on the screen of his terminal a simple graphic figure for a two-dimensional subproblem. By setting bounds on all but two criteria he is able to cut a "slice" out of the m-dimensional problem. The entire subset selected in this way will be represented by 800 fields on the screen.

Finally, the user may enter the reference point approach, interactively introduce reference point exhibiting his aspiration levels for criteria values and analyse the obtained solutions. The reference points need not to be attainable and the obtained solution is the nondominated point nearest to the reference point in the sense of the scalarizing function. A scalarizing function based on the Euclidean-norm is used. Let q be the reference point introduced by the

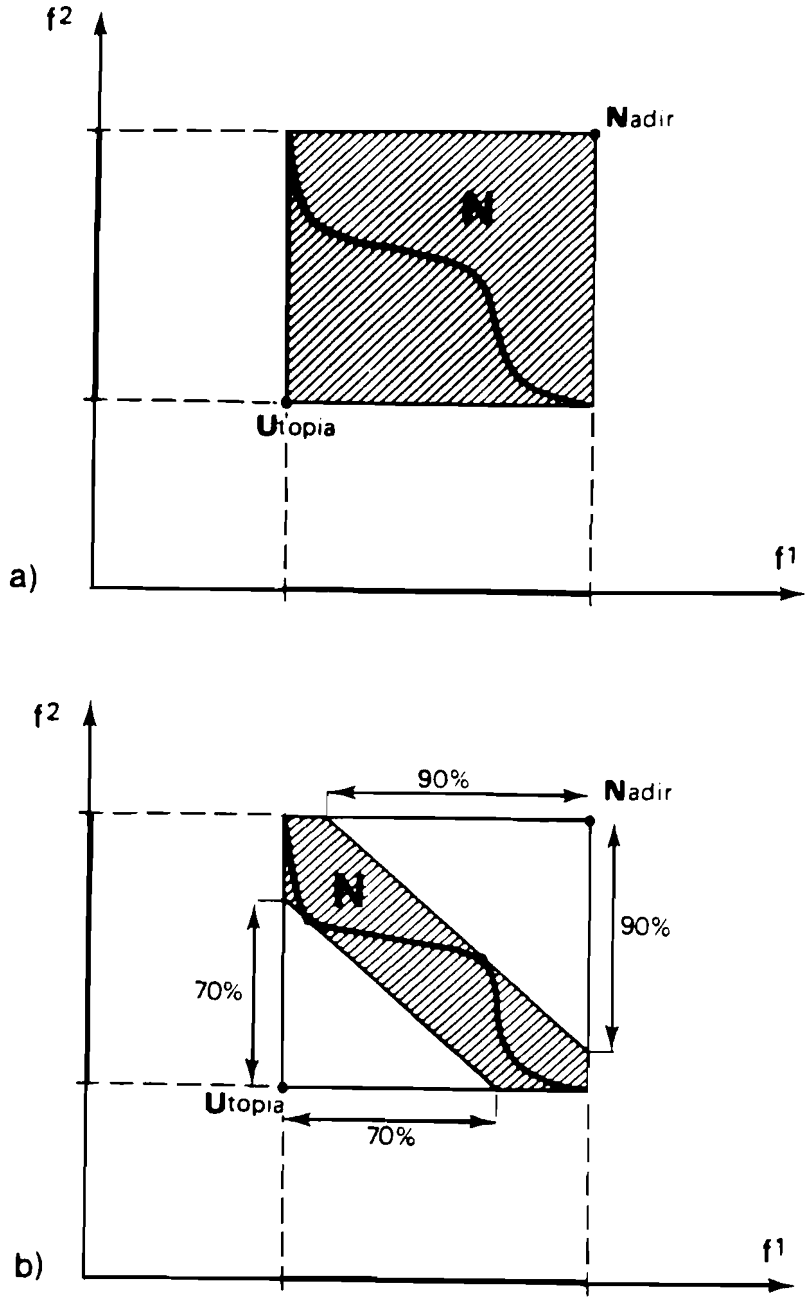

Figure 2: Two types of the aggregated information about the nondominated set N.

a) Information about the nondominated set N offered by the *utopia* point and the ***nadir*** point.

b) Information carried by the nondominated (70%) and dominated (90% of criteria range) approximations of the set N.

user. Then, assuming that all criteria are to be minimized, the following scalarizing function is minimized:

$$S(f - q) = -\|f - q\|^2 + \rho\|(f - q)_+\|^2$$

where $(f - q)_+$ denotes the vector with components $\max(0, f - q)$, $\|\bullet\|$ denotes the Euclidean norm and $\rho > 1$ is a penalty scalarizing coefficient. See (Wierzbicki, 1979), for example, for more information about the reference point approach.

6 Test examples

6.1 The Dyer's "Engine Selection Problem"

For the purpose of testing the package and to be used during introductory sessions with DISCRET, a generator of the Dyer's "Engine Selection Problem" (see Dyer, 1973, or Törn, 1980) has been implemented. This is a very simple example of a DMOP. However, it is rather well known in the literature devoted to this field of research and therefore it seems that it will suit well as a small illustrative test problem.

Let us consider a DM who designs a new automobile and he has to choose an engine for that car. Suppose that the variety of available engines is described by three parameters (decision variables):

x_1 — compression ratio

x_2 — carburation ratio (in square inches)

x_3 — piston displacement (in cubic inches)

Suppose that the DM's preferences are described by the following three criteria:

f_1 — cost of the engine

f_2 — horsepower

f_3 — mileage per gallon

The following DMOP was proposed by Dyer (1973)—see also (Törn, 1980).

Problem definition:

$$\begin{aligned}
&\textit{minimize}: & f_1(x) &= 133(x_1 - 8) + 10x_2 + x_3 + 2000\\
&\textit{maximize}: & f_2(x) &= 20(x_1 - 8) + x_2 + 0.5x_3\\
&\textit{maximize}: & f_3(x) &= -1.67(x_1 - 8) - 0.2x_2 - 0.05x_3 + 35
\end{aligned}$$

subject to:

bounds:

$$\begin{aligned}
8 \leq\ & x_1 \leq 11\\
& x_2 \leq 40\\
100 \leq\ & x_3 \leq 200
\end{aligned}$$

constraints:

$$50x_1 - 30x_2 + x_3 \leq 400$$
$$20x_1 - 3x_2 \leq 160$$
$$x_3/x_2 \leq 20$$

Problem generation:

Dyer and Törn proposed the following scheme to generate uniformly a set of decisions :

```
for $x_1 = l_1$ step $s_1$ until $u_1$
for $x_2 = l_2$ step $s_2$ until $u_2$
for $x_3 = l_3$ step $s_3$ until $u_3$
```

where l_i, u_i are the lower and upper bounds for x_i, $i = 1, 2, 3$ while s_i are the corresponding step size. If—following Dyer and Törn—the initial data are:

	l_i	s_i	u_i
$i = 1$	8	1	11
$i = 2$	10	10	40
$i = 3$	100	20	200

then 84 generated points satisfy the problem constraints. Another way to generate a test problem is a random generation of decision vectors x within bounds l and u. This test problem is generated by the DISCRET's command *test1*.

6.2 The location-allocation problem

The second test problem is a facility location-allocation problem. It is based on the problem presented by Lee, Green and Kim (1981).

A firm is evaluating six potential sites of plant location (in four different states) that would serve four customer centers. The problem is where should the plants be opened and what should be the production volume of each of the new opened plants. Let $i = 1, 2, \ldots, i_{\max} = 6$ be the locations index and let $j = 1, 2, \ldots, j_{\max} = 4$ be the customer center index.

Decision variables:

$y_i = 0/1$ if a plant is not opened / opened at location i,

z_i — production volume (size) of a plant opened at location i.

Model variables and parameters:

p_j — total demand of customer center j,

c_{ij} — unit transportation cost from facility i to the customer center j,

g_i — fixed cost of opening a facility at location i (in \$1000),

l_i — life quality score for location i,

z_i^u — production upper limit for facility at location i (due to the state environment quality standards),

z_i^l — production lower limit,

z_i^s — production increment step size,

k^i — location i production limits due to state environment quality standards,

d_{ij} — demand placed on facility i by the customer center j,

$$d_{ij} = \frac{y_i\, e^{-bc_{ij}}}{\sum_i y_i\, e^{-bc_{ij}}},$$

x_{ij} — quantity of units transported from location i to the customer center j,

$$x_{ij} = \frac{d_{ij}}{\sum_j d_{ij}} \min\left\{ z_i, \sum_j d_{ij} \right\},$$

n — number of opened facilities.

Constraints:

1. Fixed cost limitation (in \$1000):

$$\sum_i g_i y_i \leq 2000$$

2. Production limitations due to state environment quality standard:

$$z_i \leq k_i, \quad i = 1, \ldots, i_{\max}$$

3. Favored customer center service level :

$$z_1 y_1 + z_2 y_2 \geq 50$$

4. Number of opened facilities :

$$n_{\min} = 1 \leq n \leq 3 = n_{\max}$$

Criteria:

1. Unsatisfied demand level :

$$\min f_1 = \sum_i \sum_j (d_{ij} - x_{ij})$$

2. Favored customer center (no. 1) service level :

$$\max f_2 = \frac{\sum_i y_i x_{i1}}{p_1}$$

3. Total cost :

$$\min f_3 = f_5 + f_6 + f_7$$

4. Average life quality score :

$$\max f_4 = \frac{\sum_i y_i l_i}{\sum_i y_i}$$

5. Fixed cost :

$$\min f_5 = \sum_i y_i g_i$$

6. Transportation cost :

$$\min f_6 = \sum_i \sum_j s_{ij} x_{ij}$$

7. Production cost :

$$\min f_7 = \sum_i z_i e^{-\alpha z_i}$$

8. Unsold production :

$$\min f_8 = \sum_i \max\left\{0, \left(z_i - \sum_i x_{ij}\right)\right\}$$

Alternatives generation scheme:

The set of feasible alternatives is generated by the following three nested loops.
1. Consider opening $n = n_{\min}, \ldots, n_{\max}$ facilities.
2. Generate all n locations subsets of the set of locations (n—elements combinations of $i_{\max}$ elements set).
3. For each facility opened at location i consider its all available sizes z_i ranging from z_i^l to z_i^u with the increment step size z_i^s.

6.3 How to get started

At the very beginning of the session a problem to be solved has to be supplied. For the first session execute the DISCRET command *test1*. When the test problem is already generated, look at three files that were produced: the specification file, the data file and the additional data file.

Whenever you do not remember the names of the files you have created during the session, display the *history.fil* from your current directory. This file contains the history of your session.

In order to learn how to describe the details of your problem for DISCRET, print the specification file produced by the command *test1*. Then execute the specify command and try to create a specification file identical to that obtained from *test1*.

If you already know how to specify your problem, try some other DISCRET commands. For the first time, execute them in the following order: *bounds*, *solve*, *analyse*, *sort*, *pick*, just to learn what they can actually do for you.

Later on try to select your most preferable solution(s). Notice that DISCRET commands can be executed in any order (if only it does make any sense for you). Refer to the *history.fil* to recall the history of your session.

7 Conclusions

The DISCRET package for multicriteria optimization and decision making problems with finite number of discrete alternatives has been briefly presented. It is the author's hope that this report will attract the reader and encourage him to use the package.

DISCRET is an interactive package. The user may execute its commands in any order once the problem generation and specification phase has been completed. The variety of paths the user may follow guarantees flexibility in meeting his demands.

The author will be grateful for any critical remarks and comments concerning both the approach and the package itself. All such suggestions would be very helpful and may result in further package improvements.

8 References

Baum, S., W. Terry and U. Parekh (1981). Random sampling approach to MCDM. In J.N. Morse (ed): Organizations: Multiple Agents with Multiple Criteria, Lecture Notes in Economics and Math. Systems, 190.

Dyer, J.S. (1973). An empirical investigation of a man-machine interactive approach to the solution of a multiple criteria problem. In T.L. Cochrane and M. Zeleny (eds): Multiple Criteria Decision Making, University of South California Press.

Keeney, R.L. and H. Reiffa (1976). Decisions with Multiple Objectives: Preferences and Value Tradeoffs, New York, Wiley.

Köksalan, M., M.H. Karwan and S. Zionts (1984). An improved method for solving multiple criteria problems involving discrete alternatives. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-14, No. 1, pp. 24–34.

Kung, H.T., F. Luccio and F.P. Preparata (1975). On finding the maxima of a set of vectors. *Journal of the Association for Computing Machinery*, Vol. 22, No. 4, pp. 469–476.

Lee, S.M., G.I. Green and C.S. Kim (1981). A multiple criteria model for the location-allocation problem. *Comput. and Ops Res.*, Vol. 8, pp. 1–8.

Lin, J.G. (1976). Three methods for determining Pareto-optimal solutions of multiple-objective problems. In Ho and Mitter (eds.): Directions in Large-Scale Systems. Many-Person Optimization and Decentralized Control, Plenum Press, New York and London.

Majchrzak, J. (1984). Package DISCRET for multicriteria optimization and decision making problems with discrete alternatives. IIASA Conference on Plural Rationality and Interactive Decision Processes, Sopron, Hungary, 16–26 August, 1984.

Morse, J.N. (1980). Reducing the size of the nondominated set: pruning by clustering. *Comput. and Ops Res.*, Vol. 7, No. 1–2, pp. 55–66.

Payne, A.N. and E. Polak (1980). An interactive rectangle elimination method for biobjective decision making. *IEEE Transactions on Automatic Control*, Vol. AC-25, No. 3, pp. 421–432.

Polak, E. and A.N. Payne (1976). On multicriteria optimization. In Ho and Mitter (eds.): Directions in Large-Scale Systems. Many-Person Optimization and Decentralized Control, Plenum Press, New York and London.

Rivett, P. (1977). Multidimension scaling for multiobjective policies. *Omega*, Vol. 5, pp. 367–379.

Roy, B. (1971). Problems and methods with multiple objective functions. *Math. Programming*, Vol. 1, pp. 239–266.

Siskos, J. (1982). A way to deal with fuzzy preferences in multi-criteria decision problems. *Eur. J. Op. Res.*, Vol. 10, pp. 314–324.

Stahn, H. and U. Petersohn (1978). Discrete polyoptimization. *Systems Science*, Vol. 4, No. 2, pp. 101–109.

Steuer, R.E. and F.W. Harris (1980). Intra-set point generation and filtering in decision and criterion space. *Comput. and Ops Res.*, Vol. 7, No. 1–2, pp. 41–53.

Törn, A.A. (1980). A sampling-search-clustering approach for exploring the feasible/efficient solutions of MCDM problems. *Comput. and Ops Res.*, Vol. 7, No. 1–2, pp. 67–79.

Wierzbicki, A.P. (1979). A methodological guide to multiobjective decision making, WP-79-122, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Zionts, S. (1981). A multiple criteria method for choosing among discrete alternatives. *Eur. J. Op. Res.*, Vol. 7, pp. 143–147.

Part III
User Reference Manual

9 Introduction

DISCRET is a package created to solve basic multicriteria choice problems in which a finite set of feasible alternatives (and decisions) is explicitly given and, for each alternative, the values of all criteria describing its attributes interesting to the decision maker (DM) were evaluated and listed. The DM is assumed to be rational in the sense that he is looking for an efficient (Pareto-optimal) solution as his final solution of the problem.

It is assumed that the user is familiar with the contents of the DISCRET methodological guide. In the next chapters the DISCRET commands and files will be discussed.

DISCRET reads and produces standard PC DOS files. Therefore it can easily cooperate with some other packages in complex projects.

10 Structure of DISCRET

During the session with DISCRET, the user controls the process by choosing suitable items from the displayed menu (a list of options available at the moment). The menu system has been chosen instead of a pseudo-language of control commands because it does not require from the user to learn and remember a set of instructions.

Each menu contains an amount of information sufficient to make the decision which of the displayed options is the most suitable one. If the user is asked to enter some information, everything he types is checked. If he makes a mistake, a message is displayed on the screen. Usually the message not only indicates the error but also shows the correct form of the required input. We will not go into details of each menu since they are self-explanatory. Only the commands of the Main Menu will be discussed. They correspond to the basic functions DISCRET performs.

The Main Menu of DISCRET contains the following commands:

- `info` — displays some information about the system itself and the required structure of its input files;

- `load` — asks for the names of the input files containing the description of the problem and loads it;

- **bounds** — informs about the criteria values ranges (utopia and nadir points), nondominated and dominated approximations of the set of alternatives and supports setting of new bounds on criteria values;

- **solve** — solves the discrete multicriteria optimization problem with explicit alternatives (implicit constraints), i.e. finds the set of nondominated or weakly nondominated elements or its representation, keeping or rejecting duplicate elements;

- **analyze** — supports the reference point approach, simple two-dimensional graphic displays of the nondominated set and the DM's region of interest setting;

- **save** — saves the current state of the problem into the output data and specification files (saved problem may be loaded again);

- **exit** — terminates the session and returns to DOS.

The **save** command offers two useful utilities:

- **sort** — sorts the alternatives in increasing/decreasing order with respect either to the values of a specified criterion or the alternatives identifiers.
- **pick** — copies from the input additional data file into the output additional data file records containing the information concerning only these alternatives which were saved into the output data file.

Two programs are distributed together with DISCRET for the purpose of user's training:

- **test1** — first test problem generator (the Dyer's Engine Selection Problem);
- **test2** — second test problem generator (the location-allocation problem).

Refer to the methodological guide for the description of this problems.

11 DISCRET files

11.1 General remarks

Recall the discrete multicriteria optimization problem (DMOP) presented in the methodological guide. DISCRET assumes that for its purpose this problem is described in the following three input files:

- the data file which contains the criteria values for all alternatives,
- the specification file in which some details of the problem are given,
- the additional data file (optional) which contains some more information concerning the alternatives.

To start the interactions with DISCRET, the user has to supply only the data file and the specification file. This two files give a full description of the problem for DISCRET. The additional data file is optional and need not be supplied if the user decides so. The additional data file may contain any kind of information corresponding to the alternatives in the data file. For example it may contain the description of the decisions corresponding to the alternatives.

To provide an interactive mode of work DISCRET will produce the output files having the structure identical to the input files.

An example of the specification file and data file is presented at the end of this manual. Both files were produced by the **test1** command.

To create the files for DISCRET any software able to produce the standard ASCII DOS files can be used: text editors, spreadsheets, data bases, user's programs, etc.

11.2 The data file

The data file contains the criteria values for alternatives. Its format can be specified by the user almost freely. However, the following specific structure has to be preserved:

1. Alternatives have to be listed one after another.

2. Each alternative description has to start in a new record.
3. Each alternative description has to begin with an unique alternative identifier (integer number).
4. The alternative identifier should be contained in the first five characters of the first record of the alternative description.
5. The description of each alternative must be contained in the same number of records.

The two last requirements have to be fulfilled by the data file only if the `pick` command is planned to be used. The two last requirements are the only requirements for the additional data file.

We will show how to write an alternative into the data file in a Fortran program. Let us assume that the values of six criteria under consideration are contained in the real array `f`, while the value of the alternative identifier in the integer variable `iden`. The following Fortran statements will write the information about the alternative into the data file:

```
  write(1,2)iden,(f(i),i=1,6)
2 format(i5,100(5(1pg15.6):/5x))
```

The above format is called a DISCRET standard one. We would like to encourage the user to apply his own problem-specific data file formats, which will surely be much more readable.

The maximal number of criteria is in the current version of the package limited to 20, mainly due to the display facilities. The only restriction for the element identifiers is that they should form an increasing sequence of integer numbers.

11.3 The specification file

The specification file is composed of a number of sections. Each section must be provided in its specific structure. Sections may be separated any number of blank and/or non-blank lines (records). Inside the section, however, no additional lines can appear. The ordering of sections is not important.

Each of the sections of the specification file begins with a heading line, which can be followed by a number of immediately preceding comment lines. This is the only place where the comment lines may appear. Just after the last comment line (if any) the section body follows in the form specific for the given section. The contents of each section must be closed with a section-end line.

Heading line is a line starting with: three stars, single space and the section name. Comment line is a line that begins with a single star followed by a space. The sequence of the comment lines must immediately follow the section heading line. Section-end line is the line that starts with at least four stars.

Reading a section of the specification file DISCRET locates its heading line, ignores the immediately following comment lines, reads the section body and terminates on the section-end line. Hence, besides the sections for DISCRET, the specification file may also contain some other information useful for the user (another software specifications, for example).

11.4 Section SPSA of the specification file

The SPSA section begins with the heading line followed by number of comment lines.

Each of the line that do not begin with a star consists of a 38-character text and an integer value up to 5 characters, right-justified. The text is just a description of the value and is ignored. The ordering of the lines is important. Just as all sections of the specification file, the SPSA section ends with the section-end line.

The first item is a problem identifier and its only purpose is to help the user to identify the file and the problem it describes. Second item stands for the number of criteria. The third one is the number of alternatives in the data file or its overestimate. The fourth item gives the number of records (of the data file) in which the description of each alternative is contained. The last two items concern the additional data file and they indicate the number of decision variables and the additional data records number for each alternative listed in that file.

The first three items are of critical importance. They have to be specified correctly in the specification file. The last three are not so important. Even some dummy values may be assigned to them in the specification file but correct values will have to be given during the session. The fourth, fifth and sixth items of the specification section SPSA are important only for the `pick` subcommand of the save command.

11.5 Section SPSB of the specification file

In the specification section SPSB each line between the heading line followed by optional comment lines and the section-end line concerns one criterion and contains:

- criterion number,
- criterion name (up to 10 characters),
- parameter indicating whether this criterion is to be minimized (-1), maximized (+1) or ignored (0),
- criterion value tolerance — if for two alternatives the corresponding criterion values difference is lower than the tolerance, then the criterion values are assumed to be equal.

The criterion value tolerance is a parameter that plays a significant role. Its default value is zero but the user may assign to it some positive values for one of the following reasons:

- to suppress the computer floating point arithmetic rounding and truncating effects,
- to indicate the accuracy of the criterion values measurements,
- to introduce his subjective criterion value indifference quantity (for example 100 dollars when cars prices are considered),
- to force the DMOP solver to produce a representation of the nondominated set rather than the nondominated set itself.

The produced representation is a nondominated set subset preserving information about its shape but containing fewer elements. Thus, it is more convenient for the problem analysis.

11.6 Data Input Format Section DATIFMT

The Data Input Format Section DATIFMT contains the Fortran format of the data file. This format concerns the description of a single alternative. Hence, the formats of all alternatives descriptions in the data file have to be identical. The format length is restricted to 80 characters.

All Fortran requirements concerning the formats stored in character strings or arrays have to be preserved. In particular, the format has to be enclosed in parentheses.

DISCRET allows for the input data file in a (Fortran) list-directed format. In this case the DATIFMT section should contain the string "LIST-DIRECTED" with "L" being the first character of the record (no parentheses, no apostrophes). All restrictions mentioned in chapter 3.2 are valid also in this case, with the single exeption of number four — the alternative identifier does not have to occupy exactly the first five characters of the record.

If there is no DATIFMT section in the input specification file then the DISCRET standard format will be applied — see chapter 3.2.

The data input format may be used to select only some of the criteria or to separate the from a file containing also some other information.

11.7 Data Output Format Section DATOFMT

The Data Output Format Section DATOFMT contains the Fortran format of the output data file. This format may be different than that from the DATIFMT section. The format length is restricted to 80 characters.

The output data file format cannot be list-directed. A fixed format need to be specified. By specifying an appropriate format the output data file can be made more readable.

If there is no DATOFMT section in the input specification file then the DISCRET standard format will be applied — see chapter 3.2.

11.8 Additional data file

The first and very important information about the additional data file is that it is not obligatory but optional. If the user is able to handle his whole decision making process only on the base of the set of feasible alternatives he can forget about this file. However, it seems reasonable to have not only the information about the criteria values for the alternatives but also about the decisions that led to them, for example. In such case the user should generate the additional data file together with the data file.

The additional data file contains the information about the decisions corresponding with the alternatives listed in the data file. It may have one of the two following forms.

- The first and recommended form is identical with that of the data described previously, with the sole difference that instead of the criteria values the corresponding decision variables values are listed. The format of the additional data file may be different than that of the data file.

- The second form in which instead of only numerical values of decisions it may contain also some character strings describing the decisions.

The first form of the additional data file offers two advantages. Because it is identical with the data file form, it may be sometimes treated like the data file section. This fact gives the possibility of utilizing some package facilities designed originally for the data file. The bounds setting, sorting and the graphical display of the set (offered by the bounds, sort and analyze commands, respectively) seem to be most interesting in this context.

The second form of the additional data file, despite its less preferable features, may has to be used in some specific situations.

Basically however, the additional data file will be involved only during the execution of the save command. Its pick subcommand offers the possibility to select only the description of those alternatives which are present in the output data file.

Let us assume that the decision is described in 3 elements of the character array a of length 80. The following Fortran statements will write the information about the additional data element corresponding to the alternative iden into the additional data file:

```
  write(1,3)iden,(a(j),j=1,3)
3 format(i5,100(/a80))
```

12 How to get started

Copy the contents of all DISCRET diskettes into a new directory on your hard disk. Read the read.me file.

At the very beginning of the session a problem to be solved has to be supplied. For the first session execute the test1 program. When the test problem is already generated, look at three files that were produced: the specification file, the data file and the additional data file. You will learn how to describe the details of your problem for DISCRET.

If you already know how to specify your problem, try to run DISCRET. Execute the commands of the Main Menu in the order they appear, just to learn what they actually do. Later on try to select your most preferable solution(s).

Do not expect that DISCRET will tell you when to stop. You have to decide because you will take the responsibility for the decision.

Notice that DISCRET commands can be executed in any order (if only it does make any sense for you). Refer to the history.fil to recall the history of your session.

See the methodological guide for the DISCRET system if you want to learn more about the implemented approach an ways of utilizing it.

13 Conclusions

The DISCRET system for multicriteria optimization and decision making problems with finite number of discrete alternatives has been briefly presented. It is the author's hope that this report will attract the reader and encourage him to try the system.

DISCRET is an interactive system. The user may execute its commands in any order once the problem generation and specification phase has been completed. The variety of paths the user may follow guarantees flexibility in meeting his demands.

The author will be grateful for any critical remarks and comments concerning both the approach and the package itself. All such suggestions would be very helpful and may result in further package improvements.

A The example of the specification file

This is the specification file : test1.spc

the corresponding data file : test1.dat

and additional data file : test1.add

Files were produced by TEST1 program.

```
*** SPSA section
*   D E S C R I P T I O N            VALUE
identifier of the problem       idpr =  111
number of criteria              ncri =    3
number of alternatives          nalt =   84
no of records for one alternat. nral =    2
additional data dimension       nadd =    3
additional data records number  nadr =    2
****

*** SPSB section
*NO CRITERION   MIN/MAX  TOLERANCE
 1  COST in $      -1     1.000E-01
 2  HORSEPOWER      1     1.000E-02
 3  MILEAGE         1     1.000E-03
****

*** DATIFMT section
* Data Input  Format
(i5,100(5(1pg15.6):/5x))
****

*** DATOFMT section
* Data Output Format
(i5,100(5(1pg15.6):/5x))
****

*** ADDIFMT section  (just to remind - not used by "DISCRET")
* Additional Data Format
(i5,100(5(1pg15.6):/5x))
****

*** COMMENTS section
This is the "Dyer's Engine Selection problem".
It was generated by the TEST1 test problem generating program
 - supplied with the  "DISCRET" decision support system.
****
```

B The example of the data file

```
 1   2.200000E+03   6.000000E+01   2.800000E+01
 2   2.220000E+03   7.000000E+01   2.700000E+01
 3   2.240000E+03   8.000000E+01   2.600000E+01
 4   2.260000E+03   9.000000E+01   2.500000E+01
 5   2.280000E+03   1.000000E+02   2.400000E+01
 6   2.300000E+03   1.100000E+02   2.300000E+01
 7   2.300000E+03   7.000000E+01   2.600000E+01
 8   2.320000E+03   8.000000E+01   2.500000E+01
 9   2.340000E+03   9.000000E+01   2.400000E+01
10   2.360000E+03   1.000000E+02   2.300000E+01
11   2.380000E+03   1.100000E+02   2.200000E+01
12   2.400000E+03   1.200000E+02   2.100000E+01
13   2.400000E+03   8.000000E+01   2.400000E+01
14   2.420000E+03   9.000000E+01   2.300000E+01

...............................................
...............................................
...............................................

71   2.779000E+03   1.700000E+02   1.699000E+01
72   2.799000E+03   1.800000E+02   1.599000E+01
73   2.799000E+03   1.400000E+02   1.899000E+01
74   2.819000E+03   1.500000E+02   1.799000E+01
75   2.839000E+03   1.600000E+02   1.699000E+01
76   2.859000E+03   1.700000E+02   1.599000E+01
77   2.879000E+03   1.800000E+02   1.499000E+01
78   2.899000E+03   1.900000E+02   1.399000E+01
79   2.899000E+03   1.500000E+02   1.699000E+01
80   2.919000E+03   1.600000E+02   1.599000E+01
81   2.939000E+03   1.700000E+02   1.499000E+01
82   2.959000E+03   1.800000E+02   1.399000E+01
83   2.979000E+03   1.900000E+02   1.299000E+01
84   2.999000E+03   2.000000E+02   1.199000E+01
```

C The example of the additional data file

```
  1   8.000000E+00   1.000000E+01   1.000000E+02
  2   8.000000E+00   1.000000E+01   1.200000E+02
  3   8.000000E+00   1.000000E+01   1.400000E+02
  4   8.000000E+00   1.000000E+01   1.600000E+02
  5   8.000000E+00   1.000000E+01   1.800000E+02
  6   8.000000E+00   1.000000E+01   2.000000E+02
  7   8.000000E+00   2.000000E+01   1.000000E+02
  8   8.000000E+00   2.000000E+01   1.200000E+02
  9   8.000000E+00   2.000000E+01   1.400000E+02
 10   8.000000E+00   2.000000E+01   1.600000E+02
 11   8.000000E+00   2.000000E+01   1.800000E+02
 12   8.000000E+00   2.000000E+01   2.000000E+02
 13   8.000000E+00   3.000000E+01   1.000000E+02
 14   8.000000E+00   3.000000E+01   1.200000E+02

.............................................
.............................................
.............................................

 71   1.100000E+01   2.000000E+01   1.800000E+02
 72   1.100000E+01   2.000000E+01   2.000000E+02
 73   1.100000E+01   3.000000E+01   1.000000E+02
 74   1.100000E+01   3.000000E+01   1.200000E+02
 75   1.100000E+01   3.000000E+01   1.400000E+02
 76   1.100000E+01   3.000000E+01   1.600000E+02
 77   1.100000E+01   3.000000E+01   1.800000E+02
 78   1.100000E+01   3.000000E+01   2.000000E+02
 79   1.100000E+01   4.000000E+01   1.000000E+02
 80   1.100000E+01   4.000000E+01   1.200000E+02
 81   1.100000E+01   4.000000E+01   1.400000E+02
 82   1.100000E+01   4.000000E+01   1.600000E+02
 83   1.100000E+01   4.000000E+01   1.800000E+02
 84   1.100000E+01   4.000000E+01   2.000000E+02
```