International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

The Reference Point Optimization Approach - Methods of Efficient Implementation

Lewandowski, A. and Grauer, M.

IIASA Working Paper

WP-82-026

March 1982

WORKING PAPER

THE REFERENCE POINT OPTIMIZATION APPROACH - METHODS OF EFFICIENT IMPLEMENTATION

A. Lewandowski
M. Grauer

March 1982
WP-82-26

IIASA
International Institute
for Applied Systems Analysis

THE REFERENCE POINT OPTIMIZATION APPROACH - METHODS OF EFFICIENT IMPLEMENTATION

A. Lewandowski
M. Grauer

March 1982
WP-82-26

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

1. INTRODUCTION

The *reference point* approach introduced by Wierzbicki [1] has already been described in a series of papers and reports. This method is a generalization of the well-known goal programming method [2] and of the method of displaced ideals developed by Zeleny [3]. The basic idea of this method is as follows:

(I) The *decision-maker* (DM) thinks in terms of *aspiration levels*, i.e., he specifies acceptable values for given objectives.

(II) He works with the computer in an *interactive way* so that he can change his aspiration levels during the course of the analysis.

Practical experience with the DM has shown that these requirements are both realistic, which makes the approach very useful in practice. Other methods require the DM to provide rather unnatural information, e.g., the

methods based on the Morgenstern utility theory require the DM to compare the lotteries and to express his preferences in terms of probabilities [4]. It is also unreasonable to expect the DM to carry out a pairwise comparison of several alternatives. The *reference point* approach, in contrast, has proven its applicability in a number of practical problems [5, 6]. This approach has also been used in a study of the optimal structure of the chemical industry [7] and in a work dealing with the generation of efficient energy supply strategies [8].

In the authors' opinion, work on the reference point approach has now reached a stage at which efficient software based on this method can be developed. This paper will concentrate on the software package DIDASS (Dynamic Interactive Decision Analysis and Support System) being developed at IIASA to deal with linear and nonlinear multiple-criteria programming problems. There are several ways of increasing the efficiency of the software, in terms of both computing power and interaction between the user and the computer. Some of these improvements will also be discussed in this paper.

2. REFERENCE POINT OPTIMIZATION

The use of the *reference point* approach in the linear case has been discussed in an earlier paper [5].

Let A be in $R^{m \times n}$, C in $R^{p \times n}$, and b in R^m and consider the multicriteria linear program :

$$Cx = q \rightarrow \max \qquad \text{(MCLP-1)}$$

$$Ax = b \qquad \text{(MCLP-2)}$$

$$x \geq 0 \qquad \text{(MCLP-3)}$$

where the decision problem is to determine an n-vector x of decision variables satisfying $x \geq 0$ while taking into account the p-vector of objectives defined by $Cx = q$. We will assume that each component of q should be as large as possible.

An objective vector value $q = \bar{q}$ is *attainable* if there is a feasible x for which $Cx = \bar{q}$. A point $\bar{q}$ is *strictly Pareto inferior* if there is an attainable point q for which $q > \bar{q}$. If there is an attainable q for which $q \geq \bar{q}$ and the inequality is strict in at least one component, then q is *Pareto inferior*. An attainable point $\bar{q}$ is *weakly Pareto optimal* if it is not strictly Pareto inferior and it is *Pareto optimal* if there is no attainable point q such that $q \geq \bar{q}$ with a strict inequality for at least one component. Thus, a Pareto optimal point is also weakly Pareto optimal, and a weakly Pareto optimal point may be Pareto inferior. For brevity, we shall sometimes refer to a Pareto optimal point as a *Pareto point* and to the set of all such points as the *Pareto set*.

A *reference point* or *reference objective* is a suggestion $\bar{q}$ by the DM which reflects in some sense the "desired level" of the objective. According to Wierzbicki [9], an achievement scalarizing function $s(q - \bar{q})$ defined over the set of objective vectors q may be associated with reference point $\bar{q}$. The general forms of function s which result in Pareto optimal (or weakly Pareto optimal) minimizers of s over the attainable points q is given by Wierzbicki [10].

If we regard the function $s(q - \bar{q})$ as the "distance" between the points q and $\bar{q}$, then, intuitively, the problem of finding such a minimum may be interpreted as the problem of finding from within the Pareto set the point $\hat{q}$ "nearest" to the reference point $\bar{q}$. (However, as will be made clear later, the function s is not necessarily related to the usual notion of distance.) With this interpretation in mind, reference point optimization may be viewed as a way of guiding a sequence $\{\hat{q}^k\}$ of Pareto points generated from a sequence $\{\bar{q}^k\}$ of reference objectives. These sequences are generated in an interactive procedure and this should result in an interesting set of attainable points $\{\hat{q}^k\}$. If the sequence $\{\hat{q}^k\}$ converges, the limit may be seen as the solution to the decision problem.

The decision maker may be provided with initial information by maximizing

all objectives separately. A matrix D_S which yields information on the range of numerical values of the objectives is then constructed. We shall call this the *decision support matrix:*

$$D_S = \begin{bmatrix} q_1^* & q_2^1 & \cdots & q_p^1 \\ q_1^2 & q_2^* & \cdots & q_p^2 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ q_1^i & q_2^i & \cdots & q_p^i \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ q_1^p & q_2^p & \cdots & q_p^* \end{bmatrix}$$

Row i corresponds to the solution vector x_i which maximizes objective q_i.The vector with elements $q_i^i = q_i^*$, i.e., the diagonal of D_S, represents the *utopia (ideal) point*. This point is not attainable (if it were, it would be the solution of the proposed decision problem), but it can be used and presented to the decision maker as an upper limit to the sequence $\{\bar{q}^k\}$ of reference objectives . Let us consider column i of the matrix D_S . The maximum value in the column is q_i^*. Let q_n^i be the minimum value, where

$$\min_{1 \le j \le p} \left\{ q_j^i \right\} = q_n^i$$

We shall call this the *nadir* value. The vector with elements $q_n^1, q_n^2, \ldots, q_n^p$ represents the *nadir point*, and can be seen as a lower limit to the values of the decision maker's objectives.

In the following analysis we shall use the notation $w \equiv (q - \bar{q})$. A practical form of the achievement scalarizing function $s(w)$, where minimization results in a linear programming formulation, is then given by:

$$s(w) = -\min \left\{ \rho \min_i w_i \; ; \; \sum_{i=1}^{p} w_i \right\} - \varepsilon w \qquad (1)$$

Here ρ is an arbitrary coefficient which is greater than or equal to p: and $\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p)$ is a nonnegative vector of parameters. In the special case $\rho = p$, (1) reduces to

$$s(w) = -\rho \min_i w_i - \varepsilon w \tag{2}$$

In our experience, eqn. (1) has proven to be the most suitable form of achievement scalarizing function. Other practical forms are given in Wierzbicki [9].

For any scalar $\hat{s}$, the set $S_{\hat{s}}(\bar{q}) \equiv \left\{ q \mid s(w) \geq \hat{s}, w = (q - \bar{q}) \right\}$ is called a level set. Some level sets for function (1) are illustrated in Figure 1 for the cases $\rho = p$, $\rho > p$ and $\rho >> 0$ with $\varepsilon = 0$. In each case, if $w \neq 0$, then $s(w)$ is given by (2); i.e., the functional value is proportional to the worst component of w. If $\rho = p$, the same is also true for $w \geq 0$. If $w > 0$, then for large enough ρ (see the case $\rho \gg p$) $s(w)$ is given by $\sum w_i$. In the general case when $\rho > p$, the situation is as shown in the central part of Figure 1. When $w \geq 0$ and its components are sufficiently close together (that is, $(\rho - 1)w_1 \geq w_2$ and $(\rho - 1)w_2 \geq w_1$ for $p=2$), then $s(w)$ is given by $\sum w_i$. All other cases are represented by eqn. (2).

For $\varepsilon = 0$, scalarizing function (1) guarantees only weak Pareto optimality for its minimizer. However, as will be shown in Lemma 1 below, if $\varepsilon > 0$, Pareto optimality is guaranteed.

The problem of minimizing $s(q - \bar{q})$ as defined by (1) over the attainable points q can be formulated as a linear programming problem, as mentioned above. In particular, making the substitution $w = (q - \bar{q}) = (Cx - \bar{q})$ and introducing an auxiliary decision variable y, this minimization problem may be restated as follows (P):

find y, w, and x to

$$\min \ (y - \varepsilon w) \tag{P-1}$$

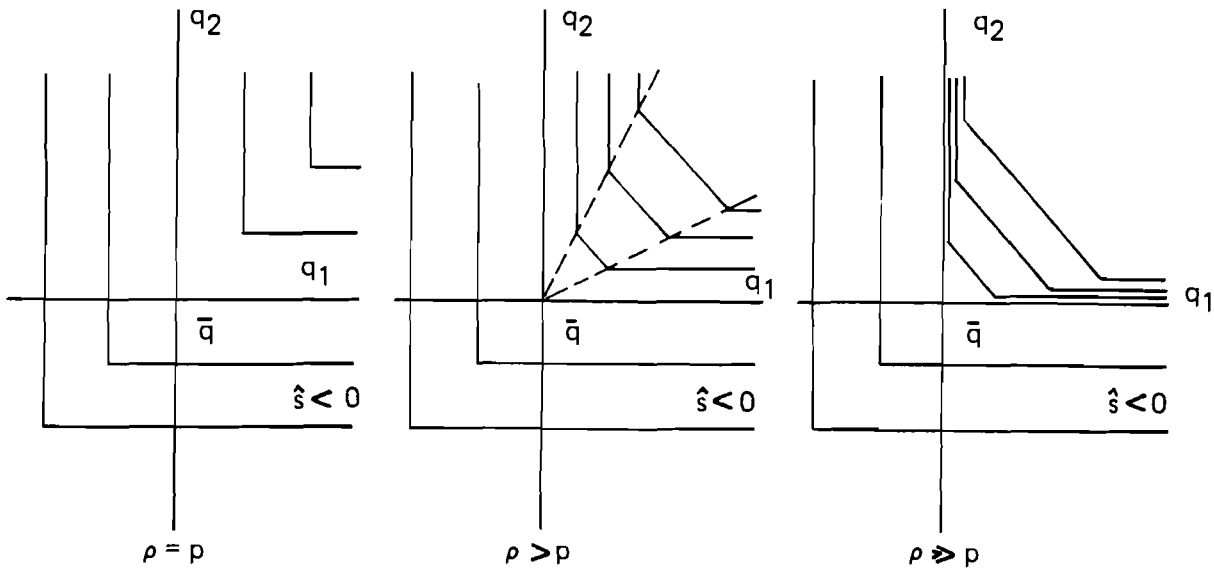

Figure 1 Level sets for achievement scalarizing functions (1) and (2) for $\varepsilon = 0$.

$$E\,y + D\,w \leq 0 \qquad \text{(P-2)}$$

$$-w + C\,x = \bar{q} \qquad \text{(P-3)}$$

$$A\,x = b \qquad \text{(P-4)}$$

$$x \geq 0 \qquad \text{(P-5)}$$

where D and E are appropriate vectors and matrices. Furthermore , $D \leq 0$, and if $w = w$ and $y = y$ are optimal for (P), then $s = y - \varepsilon\, w$ is the minimum value attained for the achievement scalarizing function s.

In the detailed formulation of (P) let $W \equiv \left\{ w \mid -w + Cx = \bar{q} \;,\; Ax = b \;,\; x \geq 0 \right\}$ denote the feasible set for vector w. Then, using the achievement scalarizing function (1), the reference point optimization problem (P) becomes:

$$= \min_{w \in W} \left\{ -\min \left\{ \rho \min_i w_i \ , \ \sum_i w_i \right\} - \varepsilon w \right\}$$

$$= \min_{w \in W} \left\{ \max \left\{ \max_i (-\rho w_i) \ ; \ -\sum_i w_i \right\} - \varepsilon w \right\}$$

$$= \min_{w \in W} \left\{ \max \left\{ \max_i (-\rho w_i - \varepsilon w) \ , \ -\sum_i w_i - \varepsilon w \right\} \right\}$$

$$= \min_{\substack{w \in W \\ z \in R}} \left\{ z \mid z \geq -\rho w_i - \varepsilon w \ ; \text{for } all \ i ; z \geq -\sum_i w_i - \varepsilon w \right\}$$

$$= \min_{\substack{w \in W \\ y \in R}} \left\{ y - \varepsilon w \mid -y - \rho w_i \leq 0 \ ; \ \text{for } all \ i \ ; \ -y - \sum_i w_i \leq 0 \right\} \qquad (3)$$

where we have substituted $y = z + \varepsilon w$.

The optimal solution of this problem is characterized by the following result:

LEMMA 1. *Let* $(y, w, x) = (\hat{y}, \hat{w}, \hat{x})$ *be an optimal solution and let* δ, μ, *and* π *be the corresponding dual vectors related to constraints (P-2), (P-3), and (P-4), respectively. Denote by* $\hat{q} = C\hat{x}$ *the corresponding objective vector, by* $\hat{s} = \hat{y} - \varepsilon\hat{w}$ *the optimal value of the achievement scalarizing function, and by* Q *the attainable set of objective vectors* q. *Then* $\hat{q} \in Q \cap S_{\hat{s}}(\bar{q})$ *and the hyperplane* $H = \left\{ q \mid \mu(\hat{q} - q) = 0 \right\}$ *separates* Q *and* $S_{\hat{s}}(\bar{q})$. *Furthermore*, $\mu \geq \varepsilon$ *and* $q = \hat{q}$ ***maximizes*** μq *over* $q \in Q$; *i.e.,* $\hat{q}$ *is Pareto optimal if* $\varepsilon > 0$ *and* $\hat{q}$ *is weakly Pareto optimal if* $\varepsilon \geq 0$.

Remark : As illustrated in Figure 2 , the hyperplane H approximates the Pareto set in the neighborhood of $\hat{q}$. Thus the dual vector μ may be viewed as a vector of tradeoff coefficients which tells us roughly how much we have to give up in one objective in order to gain a given small amount in another objective .

This Lemma is proved in [5].

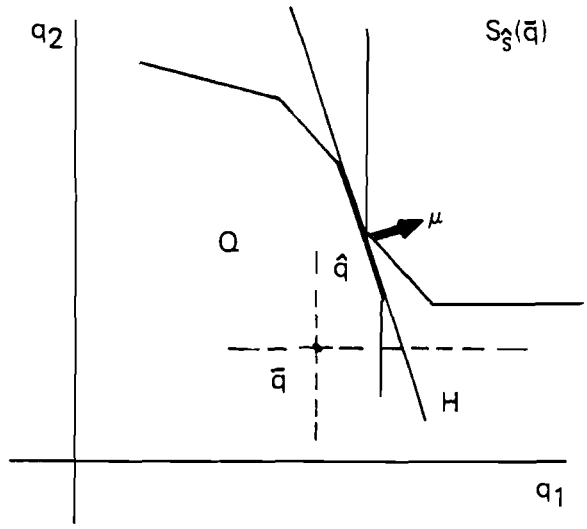

Figure 2 An illustration of Lemma 1.

3. COMPUTER IMPLEMENTATION

3.1 The Basic LP Version

The basic computer LP implementation of DIDASS consists of three parts. These are:

- The interactive "editor" for manipulating the reference point and the objectives (lpmod)
- The preprocessor, which converts the input model file containing the model description in standard MPSX format into its single-criterion equivalent (lpmulti)
- The postprocessor, which extracts the information from the LP system output file, computes the values of the objectives, and displays the necessary

information (lpsol).

This pre- and postprocessing of the LP problems makes the LP (DIDASS) system both flexible and portable. The only machine-dependent point is the format of the output file, which differs between LP packages.

All of the programs work in the interactive mode; however, the efficiency of interaction depends on the size of the LP model. Currently, one session of a 150×100 model on the VAX with the MINOS LP system (see[11]) takes about five to ten minutes CPU time. This makes the interactive analysis of quite nontrivial decision problems possible. The structure of the system and the information flow between components are presented in Figure 3.

3.2 The Extended Version - Approximation of the Pareto Set

Experience with this basic version of the software has shown that it is efficient enough to solve quite complex practical problems. However, this version has one disadvantage - if the DM changes the reference point components it is necessary to solve the LP problem again. For medium-sized LP models this usually takes at least 10 minutes of CPU time. After a brief analysis of the solution, the DM may conclude that the proposed reference point was evidently unacceptable, return to the previous solution and make a new trial. There is a simple way of avoiding such losses of time - instead of calculating a new solution corresponding to the new reference point, this solution could be estimated approximately. If this approximate solution is acceptable to the DM the exact solution can then be calculated.

The procedure for calculating the approximate solution (in the objective space) can be formulated on the basis of Lemma 1. In essence, the hyperplane H separating sets Q and $S_{\hat{s}}(\bar{q})$ can be used as the local approximation of the Pareto set.

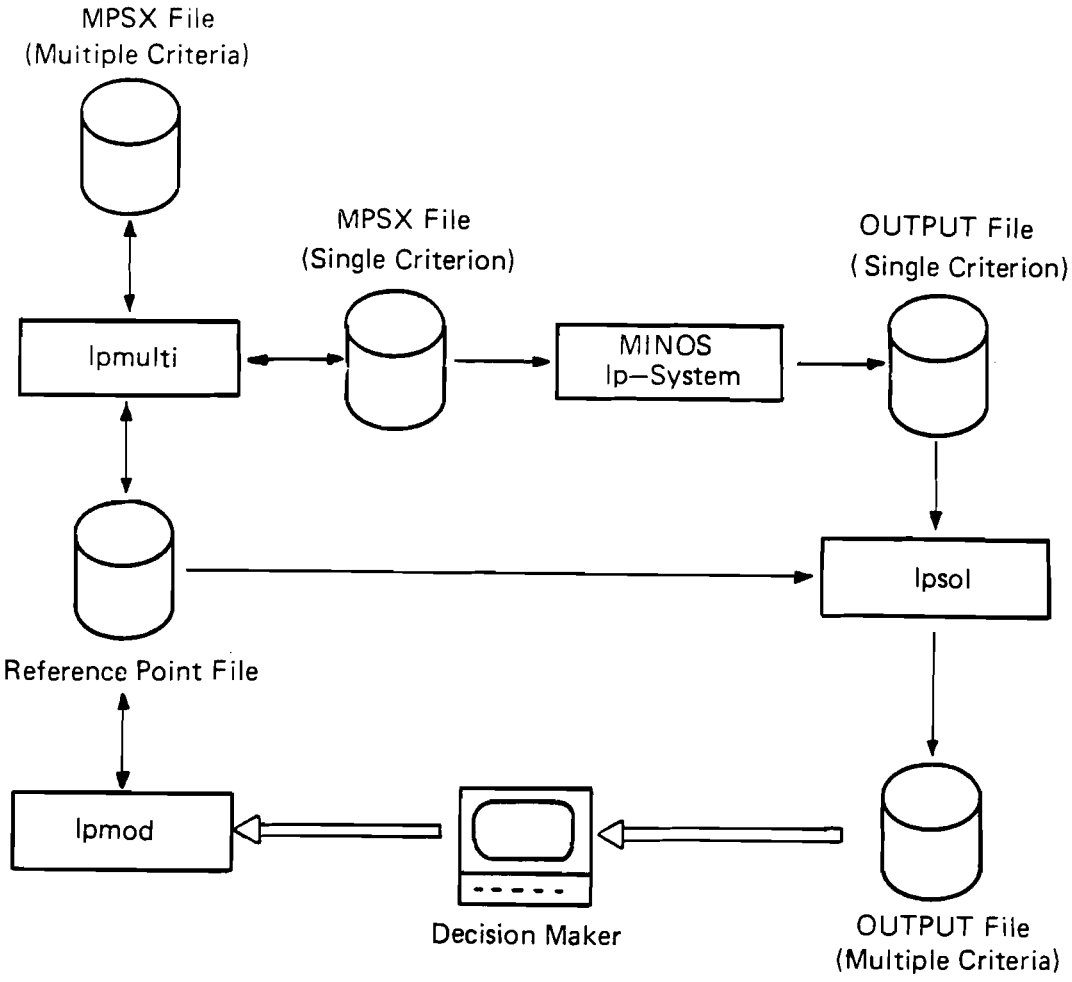

Figure 3 The structure of the multiple-criteria LP package DIDASS.

Let us assume that after the sequence of sessions we have collected the hyperplanes corresponding to each reference point

$$H_i = \left\{ q \mid \mu_i (\hat{q}_i - q) = 0 \right\} \tag{4}$$

The approximate solution can be computed as follows (AP problem):

$$\min_q S(q - \bar{q}) \tag{AP-1}$$

$$\mu_i (\hat{q}_i - q) \leq 0 \tag{AP-2}$$

Repeating the MCLP procedure, we can reformulate the problem described above as a standard LP problem. It should be noted that this problem is much simpler than the original one - the dimensionality of the decision space in this case is equal to the dimensionality of the objective space. Moreover, in view of the special structure of this problem, a simple computational procedure can be formulated; use of the LP algorithm is not necessary.

The simplest version of this algorithm has been implemented by extending the lpmod program of the package so that the DM can obtain approximate values for the objectives immediately after specifying the new reference point. This version of the program has been used in [7]; experience shows that even such a simplified approach reduces the computational effort significantly.

The procedure for calculating the approximate solution for $\rho = p$ (i.e., the scalarizing function takes the form of eqn.(2)) is simple - it is sufficient to project the vector

$$r = \bar{q}_N - \bar{q}_O \tag{5}$$

(where $\bar{q}_N$ is the new reference point and $\bar{q}_O$ is the old reference point) onto the hyperplane

$$\mu\,(\hat{q}_1 - q) = 0. \tag{6}$$

(see Figure 4).

The result of this projection is

$$\hat{\bar{q}}_N - \hat{\bar{q}}_O = \left[1 - \frac{w > < \mu}{< \mu, w >}\right](\bar{q}_N - \bar{q}_O) \tag{7}$$

where $*> <*$ denotes the outer product and $w = \bar{q}_O - \hat{q}_O$.

The solution for $\rho > p$ is more complicated; in this case the standard LP algorithm must be used.

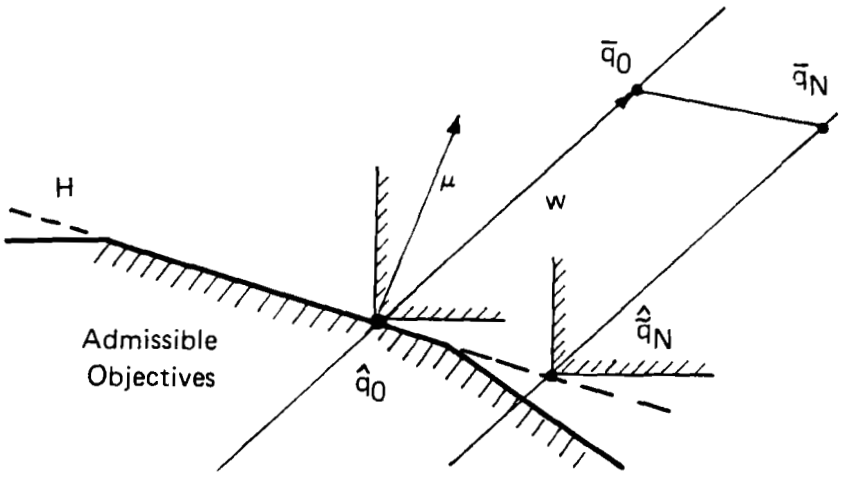

Figure 4 Estimation of the solution corresponding to a new reference point $\bar{q}_n$ starting from an "old" one $\bar{q}_0$.

Some other useful information can be obtained from the above formula, e.g., it is possible to calculate the sensitivity coefficients

$$S_{ij} = \frac{\partial \hat{q}_{Ni}}{\partial \bar{q}_{Nj}}. \tag{8}$$

It is also possible to use this equation to solve the AP problem. This may be done in the following steps:

- apply eqn.(7) and calculate q
- calculate the vector $\xi = \bar{q}_N - \hat{\bar{q}}_N$
- find the smallest nonnegative number k such that

$$q = \hat{q}_N + k\ \xi \tag{9}$$

satisfies the set of inequalities (AP.2)

$$\mu_i \, (\, \hat{q}_i - q \,) \leq 0 \qquad (10)$$

However this approximation procedure can sometimes give results that are obviously wrong (see Figure 5).

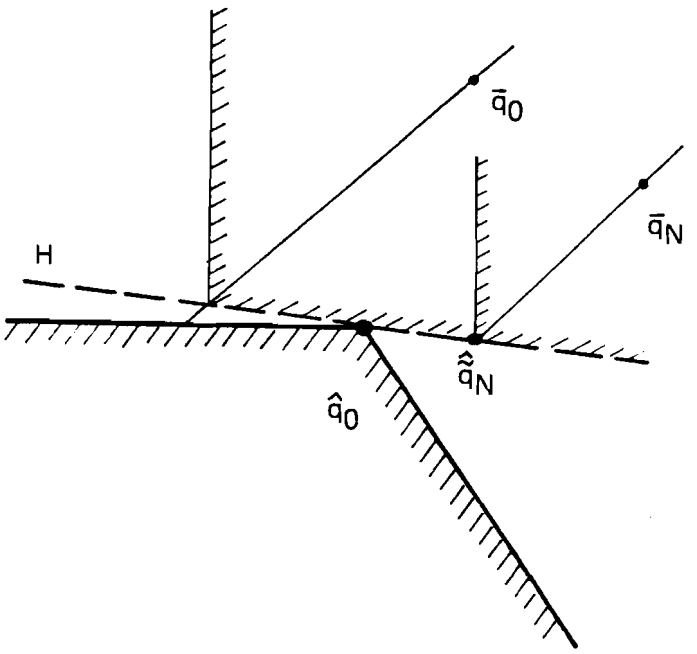

Figure 5 A case in which the hyperplane H gives an inaccurate estimate of the new solution q_N.

To avoid this situation, it is possible to use the convex hull of $\{ \hat{q}_i \}$ to approximate the set of attainable objectives. We could propose algorithms based on this technique, but as yet we do not have much numerical experience with this approach.

The above method is very simple to implement. It is only necessary to extend the lpsol program in order to generate the file containing the history of the session ($\bar{q}$,$\hat{q}$, and μ) and to modify the lpmod program in order to calculate the approximate solution (see Figure 6).

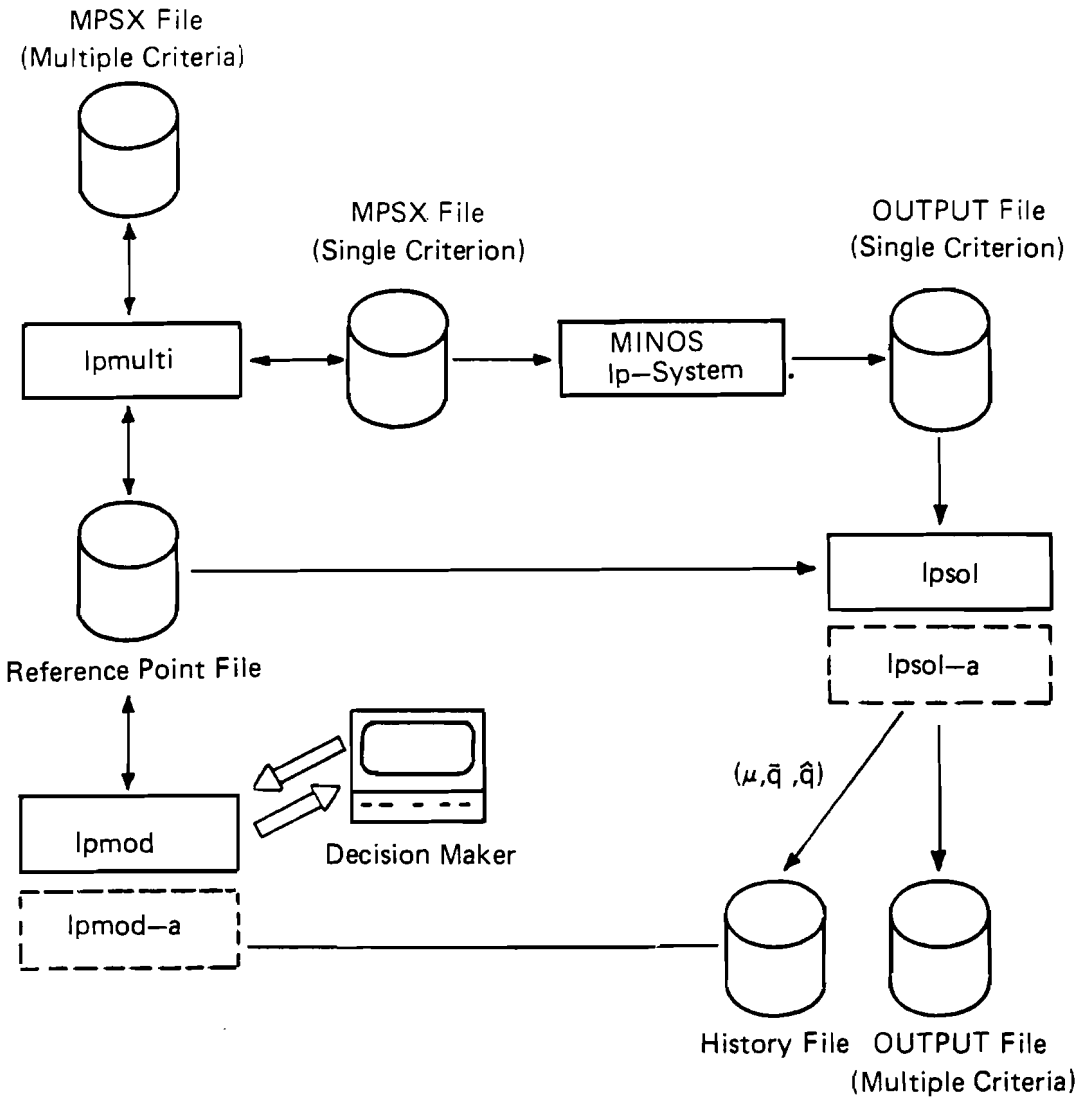

Figure 6 The structure of the extended multiple-criteria LP package with approximation of the Pareto set.

3.3 Parametric Programming Approach

Another useful approach is based on the parametric programming technique. It is easy to see that the reference point appears on the right-hand side of the constrained set in the equivalent LP problem (P). The transition from old to new reference point can therefore be parameterized as follows:

$$\bar{q} = \bar{q}_O + \xi (\bar{q}_N - \bar{q}_O) \quad , \; 0 \leq \xi \leq 1 \tag{11}$$

Let us assume that we have computed the solution $\hat{q}_O$ corresponding to a given point $\bar{q}_O$. The following procedure is carried out:

- Starting from the basis corresponding to the obtained solution, adjust the parameter ξ in the direction of ξ_1 - the value for which the perturbed problem becomes infeasible
- Calculate the values of the objectives without changing the basis
- Ask the DM whether the direction of change is acceptable. If the answer is yes, calculate the new basis and continue; if no, return to $\bar{q}_O$ and ask the DM to generate a new $\bar{q}_N$.

The basic advantage of this method lies in the fact that if the value of $\bar{q}_N$ is obviously wrong we can interrupt the calculations as soon as possible.

The parametric approach also has another advantage - by changing the parameter ξ from one basis to the other we can simultaneously collect information about approximation hyperplanes. In this way we can obtain a much more detailed approximation of the Pareto set with virtually no additional computational effort.

The basic disadvantage of the method is that it is necessary to have a specially adapted LP system (Figure 7). In many cases when the source code of the existing system is not available it will be impossible to make the necessary changes. However, even in this case the parametric LP algorithm could be used to improve the quality of the local approximation of the Pareto set.

3.4 Incorporating Constraints for Objectives

Some other programs based on the modified reference point approach are being developed and tested. One of these approaches allows the DM to force or "amplify" his preferences using the penalty function technique. In this

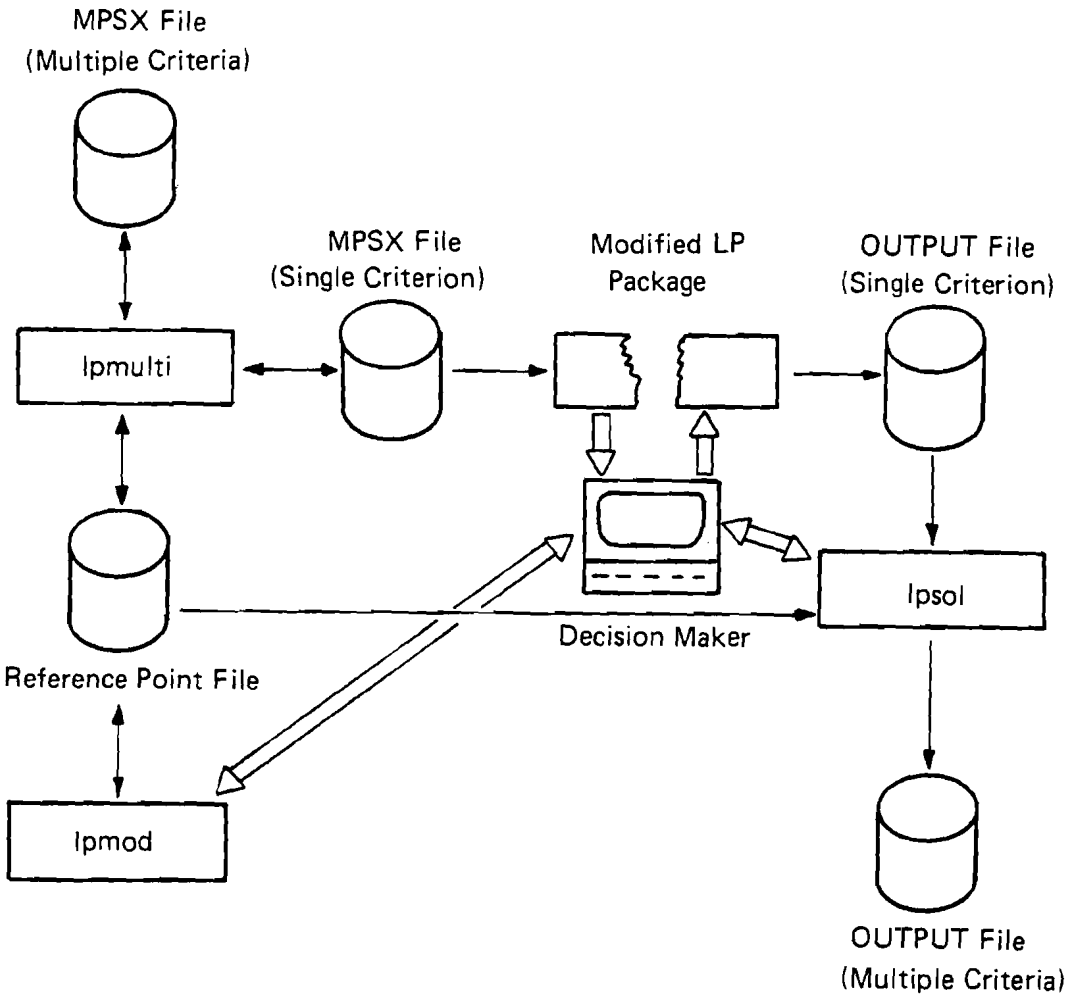

Figure 7 The structure of the multiple-criteria LP package based on the parametric approach.

approach, if the DM wishes to prevent the value of the objective changing in the wrong direction (becoming too large in a case of minimization or too small in a case of maximization), he can add a penalty function to the achievement scalarizing function.

Let J be a set of objectives for which the penalty term has been added. The modified (or nonsymmetric) scalarizing function has the following form (using (2) for simplicity):

$$s(w) = -\rho \min_i w_i - \varepsilon w + \max_{i \in J} (0 , -\rho_i w_i) \tag{12}$$

This problem can be transformed to the equivalent LP problem

$$\min s(w) = \min_{w \in W} \left\{ \max_i (-\rho w_i) - \varepsilon w + \max_{i \in J} (0, -\rho_i w_i) \right\} =$$

$$\min_{\substack{w \in W \\ y,p \in R}} \left\{ y - \varepsilon w + p \mid y \geq -\rho w_i , p \geq -\rho_j w_j , p \geq 0 , j \in J \right\} \tag{13}$$

The coefficients ρ_i in the formula express the "power" of the DM to keep the constraints

$$w_i \geq 0 \tag{14}$$

unviolated. In other cases, it is necessary to introduce two-sided constraints for the selected objectives. This type of problem arises frequently in cases of trajectory optimization when we want to ensure that a certain (reference) trajectory will be traced. In this case the achievement scalarizing function has the following form:

$$s(w) = -\rho \min_i w_i - \varepsilon w + \max_{i \in J} (0, -\rho_i w_i) + \max_{i \in M} (0, -\rho_i w_i) + \max_{i \in M} (0, \rho_i w_i) , \tag{15}$$

where M is the set of objectives for which two-sided constraints have been introduced. Transformation of this function into the equivalent LP problem is straightforward.

Programs based on these concepts have been written and testing has begun; further work on development and testing will be necessary.

3.5 Reference Point Approach With a Partly Nonlinear Objective Function

The LP approach presented in previous sections can be extended to the nonlinear case. If we consider the performance vector as an extesion of (MCLP-1)

$$f(x) + Cx = q \tag{16}$$

the equivalent nonlinear programming problem can be formulated as follows

$$\max_{\substack{w \in W \\ y \in R}} \left\{ y - \varepsilon w \mid y - \rho w_i \leq 0 \ , \ -y - \sum_i w_i \leq 0 \right\} \tag{17}$$

where

$$W \equiv \left\{ w \mid -w + f(x) + Cx = \bar{q} \ , \ Ax = b \ , \ x \geq 0 \right\}. \tag{18}$$

Implementation in this case is quite straightforward -- the standard version of the package can be used, the only difference being the need to write a FORTRAN procedure to calculate $f(x)$. The resulting nonlinear programming problem can be solved using the MINOS system or a similar package without any changes in the system.

3.6 The General Nonlinear Version

The basic nonlinear version of DIDASS also uses the idea of pre- and post-processing described in Section 3.1 (see Figure 8). In the nonlinear version of DIDASS, the decision support matrix D_S is calculated in the first step (Utopia) and the information about the utopia point and the nadir point is used to help the DM to choose the reference points. The interactive editor (NLPmod), the preprocessor(NLPmulti) and the postprocessor (NLPsol) operate similar as in the linear case .

The nonlinear constrained multiple-criteria problem to be solved must be expressed in the following standard form:

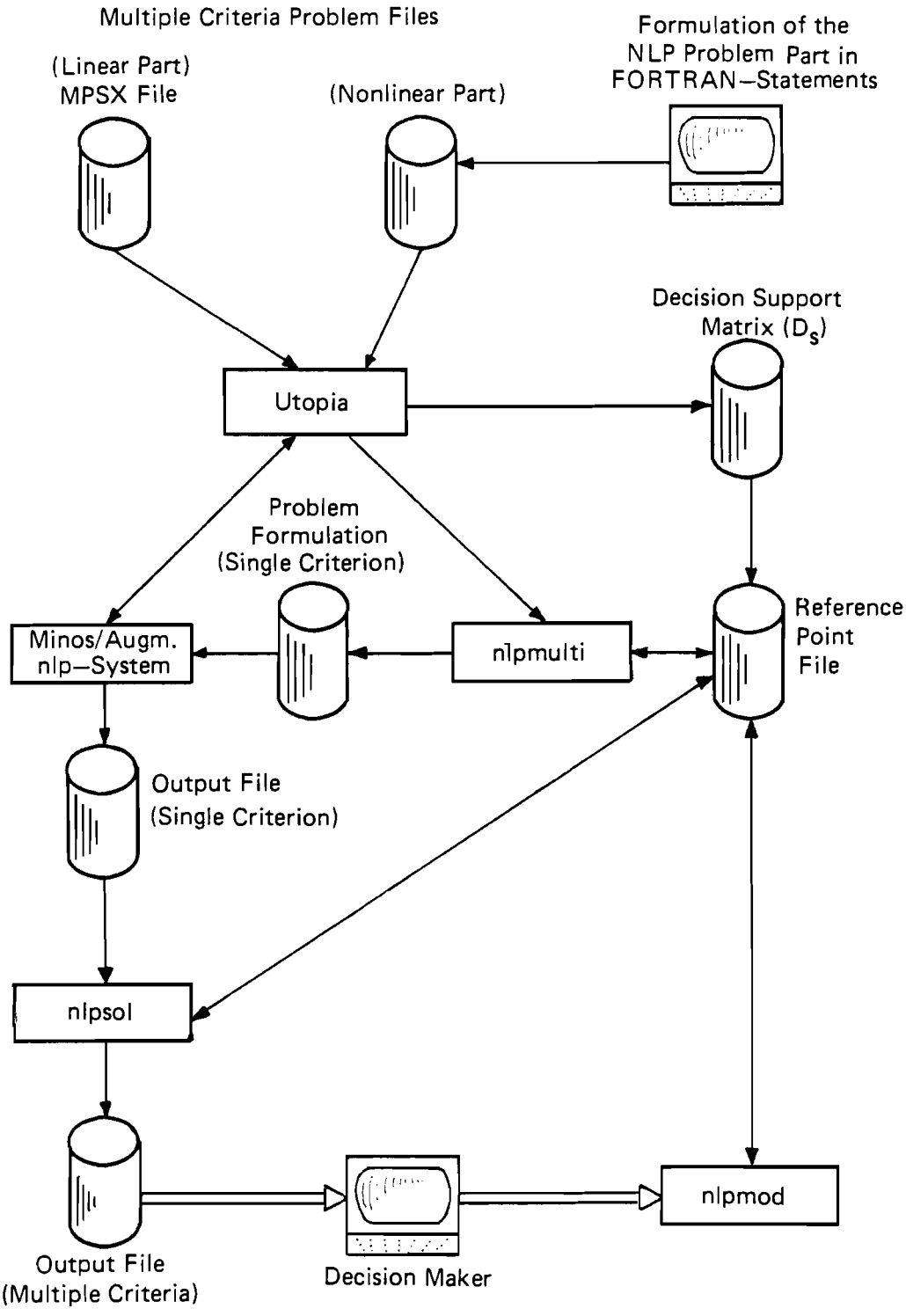

Figure 8 The structure of the nonlinear multiple-criteria package DIDASS.

$$\max_{x_{nl},x_l} \begin{bmatrix} f_1(x_{nl}) + c_1^T x_{nl} + d_1^T x_l = q_1 \\ f_2(x_{nl}) + c_2^T x_{nl} + d_2^T x_l = q_2 \\ \cdots \quad \cdots \quad \cdots \\ f_p(x_{nl}) + c_p^T x_{nl} + d_p^T x_l = q_p \end{bmatrix} \tag{19}$$

subject to:

$$g(x_{nl}) + A_1 x_l \leq b_1 \tag{20}$$

$$A_2 x_{nl} + A_3 x_l \leq b_2 \tag{21}$$

$$l \leq \begin{bmatrix} x_{nl} \\ x_l \end{bmatrix} \leq u \tag{22}$$

where $g(x_{nl}) = \left[g_1(x_{nl}), g_2(x_{nl}), \cdots, g_m(x_{nl})\right]^T$ is the vector of nonlinear constraints. The independent variables are divided into two subsets: (x_{nl}) - a vector of "nonlinear" variables and (x_l) - a vector of "linear" variables.

The following two achievement scalarizing functions have undergone preliminary testing with positive results:

$$s(w) = -\sum_{i=1}^{p} w_i^2 \tag{23}$$

where $w_i = (q_i - \bar{q}_i)/\bar{q}_i$ and $\bar{q}$ is not attainable and further

$$s(w) = -\frac{1}{\rho} \ln \left[\frac{1}{p} \sum_{i=1}^{p} (w_i)^\rho \right] \tag{24}$$

where $w_i = (\tilde{q}_i - q_i)/(\tilde{q}_i - \bar{q}_i)$, and $\tilde{q}_i$ is an upper limit for the sequence of reference points.

However further testing of the numerical features of suitable achievement scalarizing functions for the nonlinear case is necessary.

The nonlinear and linear versions of DIDASS differ in that the user must write FORTRAN statements for the nonlinear parts of the performance criteria $f_1(x_{nl}), f_2(x_{nl}), \ldots, f_p(x_{nl})$ in (19) and the nonlinear parts of the constraints $g(x_{nl})$ in (20) in the nonlinear case. The resulting single-criterion nonlinear pro-

gramming problem obtained using (23) or (24) is solved using the MINOS/AUGMENTED system [11].

4. RELATED PROBLEMS

One of the crucial points in designing interactive multiple-criteria optimization systems is that the interaction between the DM and the computer should be as simple as possible.

A number of important points should be taken into account:

- The DM is usually not a computer specialist, and for this reason the dialogue should be as simple as possible, free of technical details and easy to interpret. In particular, error messages should be self-explanatory. The command language should be as close to the natural language as possible. An interesting outline of this problem can be found in [12], and a more general discussion is given in [13].
- A special effort should be made to present the information in a simple form, preferably graphically. In the simplest case, two-dimensional projections of the Pareto point in the objective space can be very useful [7]; the cuts (or slices) of the Pareto set can give valuable information that is easy to understand.
- Special software must be designed to obtain results from the LP system output file quickly and easily. If the DM is obliged to go through hundreds of pages of computer printout to find the required information, the interaction is not efficient enough. Software systems such as PERUSE [14] can help to overcome this problem.
- Experience with DMs shows that they can usually remember only the results obtained during the last 5 - 10 iterations. In many cases the DM specifies a reference point which has already been specified or which is very close to

one specified in the past; in other cases the DM is not self-consistent and the preferred directions of change contradict those expressed in previous sessions. These situations should be detected and the DM informed.

The general structure of the DM-computer interface is displayed in Figure 9.

It should be pointed out that a number of multiple-criteria packages with a reasonably good interface already exist [15]. This paper represents only an initial stage of development of a Decision Support System from an existing Multiple Criteria Optimization package - much work still remains to be done.

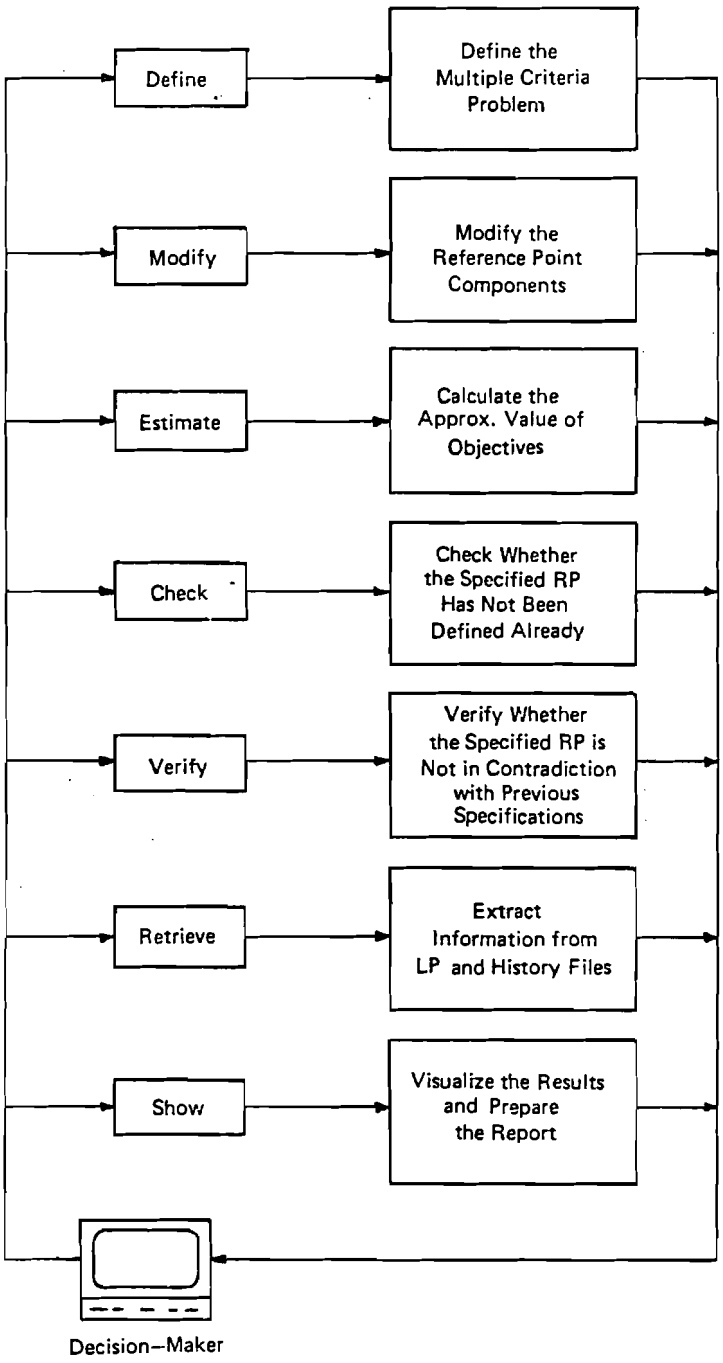

Figure 9 Structure of the DM-computer interface.

References

1. A. Wierzbicki, "A mathematical basis for satisficing decision making ," WP-80-90, IIASA (1980).

2. J.P. Ignizio , "A review of goal programming: a tool for multiobjective analysis," *J. Opl. Res. Soc.* **29**(11) pp. 1109-1119 (1978).

3. M. Zeleny , *Linear multiobjective programming* , Springer-Verlag, Heidelberg, Berlin, New York (1974).

4. R.L. Keeney and A. Sicherman, "An interactive computer program for assessing and analyzing preferences concerning multiple objectives," RM-75-12, IIASA (1975).

5. M. Kallio, A. Lewandowski , and W. Orchard-Hays , "An implementation of the reference point approach for multiobjective optimization.," WP-80-35, IIASA (1980).

6. J.P. Kindler, P. Zielinski, and L. de Mare, "An interactive procedure for multiobjective analysis of water resources allocation," WP-80-35, IIASA (1980).

7. G. Dobrowolski, J. Kopytowski, A. Lewandowski, and M. Zebrowski, "Generating the efficient alternatives for the development process of the chemical industry," CP-82, IIASA (1982).

8. M. Grauer , L. Schrattenholzer, and A. Lewandowski, "Use of the reference level approach for the generation of efficient energy supply strategies," WP-82-19, IIASA (1982).

9. A.P. Wierzbicki, *On the Use of Penalty Functions in Multiobjective Optimization*, Institute of Automatics, Technical University of Warsaw. (1978).

10. A. Wierzbicki, "The use of reference objectives in multiobjective optimization. Theoretical implications and practical experiences.," WP-79-66, IIASA (1979).

11. B.A. Murtagh and M.A. Saunders, "Minos/Augmented," Technical Report SOL-80-14, Systems Optimization Laboratory, Stanford University (1980).

12. B. Melichar, "Nonprocedural communication between users and application software," RR-81-22, IIASA (1981).

13. R.A. Guedj, "Methodology of Interaction," *Proceedings of the IFIP, Workshop on Methodology of Interaction*, North-Holland Publ. Comp., (1980).

14. W.G. Kurator and R.P. O'Neill, "PERUSE: An interactive system for mathematical programs," *ACM Trans. Math.Softw.* **6**(4) pp. 489-509 (1980).

15. M. Sakawa and F. Seo, "An Interactive Computer Program for Subjective Systems and Its Applications," WP-80-64, IIASA (1980).