*Computers Math. Applic.* Vol. 17, No. 12, pp. 1535–1543, 1989
Printed in Great Britain. 

0097-4943/89 $3.00 + 0.00


# AN INTERACTIVE MOLP PROCEDURE USING EPSILON-CONSTRAINTS

M. A. VENKATARAMANAN[1], J. MOTE[2] and D. L. OLSON[3]
[1]School of Business, Indiana University, Bloomington, IN 47405, U.S.A.
[2]University of Texas at Austin, Austin, TX 78712, U.S.A.
[3]Texas A & M University, College Station, TX 77843, U.S.A.

(*Received* 16 *June* 1988)

**Abstract**—Interactive multiobjective programming seeks to aid decision making in complex problems where it is difficult to explicitly state decision maker utility. A decision making aid is presented which uses a controlled pattern of objective attainments to generate new alternatives for decision maker selection. This procedure follows the concept of Steuer's algorithm, but avoids the need for filtering by use of constraints on objective attainment. In addition, the technique is not limited to original model corner points. The overall system seeks to obtain the benefits of Steuer's method, but requires only standard linear programming code, and adds the ability to identify improved solutions relative to Steuer's method when nonlinear utility exists.

## 1. INTRODUCTION

Interactive multiple objective programming provides a means to aid decision making under conditions of complex tradeoffs. Multiple objective analyses can be viewed as a means of maximizing decision maker utility. The concept of a utility function is useful theoretically. However, identification of a working utility function is difficult at best. In multiple objective linear programming (MOLP), it is assumed that the DM is able to choose between a limited number of options offered, and that the DM's choice behavior is compatible with his or her implicit, that is, unstated, utility function. According to White [1], the basic intent is to determine the "nondominated compromise" (or near best) solution to a specific problem, without having to generate any precise value function about the decision maker's (DM) preference characteristics.

Multiple objective programming can be considered to be constrained optimalization of utility. It is generally assumed that while a DM may have an implicit utility function that explains choice, the DM is not able, or does not wish to spend the time and tedium necessary, to fully express utility in a sufficiently comprehensive form to allow direct solution. In addition, utility is often assumed to be nonlinear (more of a good is preferable to less of a good, but at some decreasing rate). In practice, nonlinear solution methods are usually intractable. Interactive multiple objective programming seeks a means to obtain the most preferred decision in a workable manner by using choice selections of the DM. Desirable features of such methods are that they: (1) provide the DM with information leading to better understanding of the decision problem; (2) assure rational (nondominated) solutions; and (3) do not unduly burden the DM.

A number of recent approaches provide useful tools. However, they tend to be limited by assuming an approximately linear utility by using weights on sums of the multiple objectives. The method of Zionts and Wallenius [2, 3] provides a search technique based upon multiple objective linear programming (MOLP) simplex, with the useful result of a linear approximation to DM utility. De Samblanckx *et al.* [4] tested this method in a student setting, and found that it improved decision making and was relatively easy to use. However, for large problems, with many nonbasic paths to check, this technique can be intractable if a special code is not available. An alternative technique by Steuer [5, 6] provides an effective means of analyzing large models. Steuer's approach utilizes linear programming corner points, and filters nondominated solutions [7]. A number of applications have been presented [8–12]. That technique, however, is limited in that near linear utility is assumed, and only original model corner points are considered. In addition, special computer code is needed to filter the number of alternatives presented to the DM at each decision iteration.

The epsilon-constraint technique has been used in the STEP method [13] and in the surrogate worth tradeoff method [14]. The concept of this method is that the DM's selection of bounds on objective attainments can be interactively imposed through constraints as new information concerning tradeoffs is obtained. The primary benefit of imposing bounds on some objectives while optimizing others is that if utility is nonlinear, superior nondominated solutions may not be at original corner points. By adding constraints bounding selected objective function values, new corner points are created, and solutions yielding higher utility may be obtained.

The purpose of this paper is to demonstrate how conventional linear programming code can be used to obtain the solutions yielded by Steur's method, and how bounds on objective attainments can be used as constraints to yield improved solutions when utility is nonlinear (this nonlinear utility is required to be convex).

This method can assist DM learning by identifying tradeoffs, assuring nondominated solutions, and reflecting nonlinear utility.

*General multi-objective formulation*

A general formulation of the multi-objective programming problem is:

$$\sum_{j=1}^{n} c_{ij} x_j$$

$$\sum_{j=1}^{n} c_{2j} x_j$$

$$\vdots$$

$$\sum_{j=1}^{n} c_{kj} x_j \tag{1}$$

subject to $A\mathbf{x} = \mathbf{b}$, $x_j \geqslant 0$ for $j = 1, n$ where $A\mathbf{x} = \mathbf{b}$ is the feasible region defined by linear constraints and $k$ is the number of objectives. In order to be solved with a linear programming code a composite objective function must be used. Weights could be obtained by utilizing the eigenvectors associated with the DM's pairwise comparison of the objectives, or some other technique [15, 16]. The problem can be stated as:

$$\max \tilde{Z} = W_1 z_1 + W_2 z_2 + \ldots + W_k z_k \tag{2}$$

subject to $A\mathbf{x} = b$, $W_k > 0$, $x_j \geqslant 0$ for $j = 1, n$, where $W_k$ are the set of weights of DM preference among the $k$ objectives. All weights are strictly positive to assure nondominated solutions.

Let the initial solution be $\tilde{Z}^0$ with attainments $z_1, z_2, \ldots, z_k$ for the $k$ objectives. The weights of (1) can be varied to obtain a payoff table for each objective.

## 2. A PROCEDURE AND ITS RATIONALE

*Step 1: develop tradeoff table*

A composite objective function of the form given in (2) is formed. All objectives are converted to maximization form. For each of the $k$ objectives, the weights are assigned such that one objective is given a high weight, and all other objects a minimal weight. This assures that nondominated solutions are obtained. Each of these $k$ weighted combinations are then used to obtain a solution to the linear programming constraint set. The $k$ solutions then yield a payoff table, exhibited in Table 1. From this table, maximum and minimum attainments for each objective are obtained. This is presented to the DM, providing a compact demonstration of the tradeoffs involved. While the minimum nondominated attainment level for objectives is not guaranteed [17], obtaining that information is difficult. The payoff table provides a quick means of demonstrating the tradeoffs between objectives to the DM. The measures obtained are:

$z_k^{\mathrm{M}}$ = the maximum attainment for objective $k$.

Table 1. Payoff table

| | Attainments | | | |
|---|---|---|---|---|
| | $z_1$ | $z_2$ | ... | $z_k$ |
| Maximize $z_1$ | $z_1^1$ | $z_1^2$ | ... | $z_1^k$ |
| Maximize $z_2$ | $z_2^1$ | $z_2^2$ | ... | $z_2^k$ |
| ⋮ | | | | |
| Maximize $z_k$ | $z_k^1$ | $z_k^2$ | ... | $z_k^k$ |

†Diagonal elements are maximal nondominated attainments for each of the $k$ objective functions.
$z_i^j$ = attainment of $z_j$ while maximizing $z_i$.

*Step 2: obtain initial solution*

An objective function reflecting positive weight for each of the $k$ objectives is required. This could be a linear approximation of initial utility, such as use of analytic hierarchy process [16, 18], could be obtained from analysis utilizing Steuer's technique, or some other method. We have found that the composite objective function must "point" in a direction yielding improved utility, or else objective function bounds will not be guaranteed to yield improved solutions. In this analysis, we use Steuer's method to obtain the weights, as well as to identify the best original model corner point solution. The DM is given the attainments of each objective maximum as well as the additional solution generated by this composite objective function (there may be duplications). The search will iterate until no improved solutions are identified.

*Step 3: generate new solutions*

If $k < 5$ go to Step 3a.

If $k \geqslant 5$ go to Step 3b.

This step enables varying the step length used in generating new solutions depending upon the number of objectives considered. The reason for controlling the number of solutions depending upon the number of objectives is that it is desirable to limit the number of alternative solutions presented to the DM. For a small number of objectives ($k \leqslant 5$), two distance metrics, $m_1$ and $m_2$, can be formed yielding $2k$ potential new alternatives for consideration. For problems with more than five objectives, $k$ potential new solutions should provide sufficient alternatives for DM consideration.

If a very good starting point is obtained, as with using Steuer's method, the search can be confined to small changes in the direction of improving each objective in turn.

*Step 3a: develop 2k new solutions (for $k \leqslant 5$).* Two distance parameters, $m_1$ and $m_2$, are created. These distance parameters are used to vary the bounds on each objective in turn, in order to generate new solutions in a controlled manner. The parameter $m_1$ is used to generate alternatives close to, but perturbed by some amount, from the current solution. For problems with few objectives (say, less than 5), another parameter $m_2$ is used to generate new solutions in a broader search pattern. We arbitrarily set values of for $m_1 = 0.05$ and $m_2 = 0.25$. This provides a controlled search pattern in the direction of each objective. Since the intent of these values is to determine the attractiveness of moving in the direction of each objective, the specific values used could be adjusted. In this step, the bounds for each objective are varied in turn by creating constraints:

$$z_k \geqslant z_k^c + (z_k^M - z_k^c)\, m_1$$

and

$$z_k \geqslant z_k^c + (z_k^M - z_k^c)\, m_2 \tag{3}$$

where $z_k^c$ is the current attainment for objective $k$, which are added to the constraint set. Two LPs are solved for each of the $k$ objectives, adding one of the constraints in (3) for each objective. This yields $2k$ new solutions for DM consideration. Go to Step 4.

*Step 3b: Develop k new solutions (for $k > 5$).* Generate one constraint for each objective:

$$z_k \geqslant z_k^c + (z_k^M - z_k^c) m. \tag{4}$$

For each objective $k$, solve LP (2) s.t. $\tilde{x} \in S$ in addition to [4]. This operation will generate up to $k$ new solutions, to be considered in addition to the current solution. Go to Step 4.

*Step 4: DM review*

Present all of the alternative solutions generated in Steps 3a or 3b to the DM along with the current solution. The DM selects the most favored solution. By being presented with the attainments of solutions obtained by perturbing in the direction of each objective, the DM is given additional information concerning the tradeoffs between objectives in the region of the current solution. If a newly generated alternative is selected as the most preferred solution, go to Step 3.

If the DM selects the current solution (the solution selected in the prior iteration), the DM can continue with Step 5 if a finer search pattern is desired, continue with Step 3, or quit (go to Step 6).

*Step 5: DM desires to explore solution space in a finer pattern*

The distance parameter $m$ (or $m_1$ and $m_2$ if two metrics are used) can be fine tuned by having it, or if the DM prefers, some other values less than the prior $m$. Go to Step 3.

*Step 6: Finish*

Print the current best solution $\tilde{Z}^c$ along with current attainments for each objective, maximum solutions for each objective $z_k^M$, and minimum objective attainment identified, $z_k^-$. Stop.

## 3. COMPUTATIONAL COMMENTS

The constraint method of generating alternative solutions for DM consideration is computationally appealing, because it controls the attainments of objectives in a favorable manner. Steuer's method operates by manipulating the objective function. Steuer's procedure can be unpredictable because various objective attainments may change radically, as only original model corner points are considered. Our experience is that the most difficult step of Steuer's method is the filtering required to cull the desired number of new solutions at each iteration. Steuer has encoded that procedure, but that code is not widely available. The approach presented here eliminates the need for filtering by generating controlled patterns in objective attainment. Further, obtaining nondominated solutions that are not at original model extreme points is made possible.

## 4. A NUMERICAL EXAMPLE

To demonstrate the method, a network problem, given in the Appendix, is solved. The problem had 3 objectives, 36 variables, and 47 constraints. The problem is sketched in Fig. 1.

*Step 1: develop tradeoff table.* Composite objective functions, maximizing each objective in turn, with small weights on the other two objectives to avoid dominated solutions yield:

| | Attainments | | |
|---|---|---|---|
| | $z_1$ | $z_2$ | $z_3$ |
| (1) max $1000 z_1 + 1z_2 + 1z_3$ | 9277* | 5442 | 7493 |
| (2) max $1z_1 + 1000 z_2 + 1z_3$ | 7162 | 7454* | 5814⁻ |
| (3) max $1z_1 + 1z_2 + 1000 z_3$ | 6693⁻ | 4530⁻ | 11524* |

where "*" indicates the maximum for an objective and "⁻" indicates the minimum identified attainment for an objective. This information is presented to the DM, as part of a learning process. The DM knows what the best attainable value is for each function, as well as having some idea of the range of tradeoffs involved. This should provide a basis for evaluating alternatives at later iterations.

*Step 2: develop a working objective function.* A working objective function is used for the subsequent LP models used to generate new alternative solutions. We use the results of Steuer's method:

$$\max\ 0.555\, z_1 + 0.222\, z_2 + 0.222\, z_3.$$

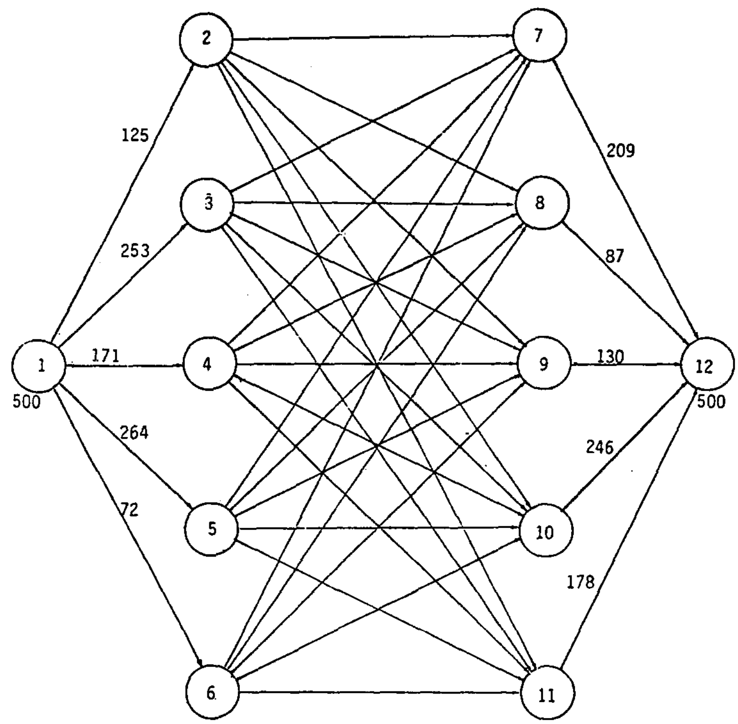


Fig. 1. Network problem.

Originally, we simply used weights of 1 for each objective. This will provide useable solutions. However, accurate estimates of the objective function weights are necessary to ensure improved solutions. We found Steuer's method an expeditious means of obtaining these weight estimates. Table 2 gives the results of the analysis using Steuer's method, obtained by following procedures published by Steuer, using an LP code.

The true implicit utility function is assumed to drive preferences of the DM, without requiring utility to be expressible mathematically. Steuer's technique seeks to reduce the decision maker burden by presenting a controllable number of alternative solutions, usually five, for comparison at any one point. Utility is inferred by DM selection.

In this example, we will use a continuous, nonlinear function to represent the utility function of the DM. The function is

$$\text{Max } 1.0\left[3+\frac{z_1^c - z_1^-}{z_1^M - z_1^-}\right]^3 + 1.1\left[3+\frac{z_2^c - z_2^-}{z_2^M - z_2^-}\right]^3 + 1.2\left[3+\frac{z_3^c - z_3^-}{z_3^M - z_3^-}\right]^3.$$

This function has the feature that more of a good is better than less. The function plays no part in the analysis other than as a means to select among alternative solutions.

In this example, the DM is given $k + 1$ (four) solutions for consideration:

| Attainments | $z_1$ | $z_2$ | $z_3$ | Utility |
|---|---|---|---|---|
| $z_1^M$ | 9277 | 5442 | 7493 | 146.581 |
| $z_2^M$ | 7162 | 7454 | 5814 | 135.000 |
| $z_3^M$ | 6693 | 4530 | 11,524 | 133.500 |
| $\tilde{Z}^0$ | 9050 | 4784 | 9314 | 148.825 |

We start with Steuer's solution, $\tilde{Z}^0$, as that has the highest utility.

For a linear utility function, this would be the most preferred solution possible, given DM consistency. However, nonlinear utility may result in a noncorner point having higher utility. We propose using bounds on each objective in turn to generate new alternatives, giving DMs the ability to verify preference.

Table 2. Results of Steuer analysis

| Weights | | | Attainments | | | | |
|---|---|---|---|---|---|---|---|
| Obj 1 | Obj 2 | Obj 3 | Obj 1 | Obj 2 | Obj 3 | Utility | Unique solution |
| Iteration 1 | | | | | | | |
| 0.333 | 0.333 | 0.333 | 8119 | 4751 | 10,691 | 145.51 | 1 |
| 0.998 | 0.001 | 0.001 | 9277 | 5442 | 7493 | 146.85† | 2 |
| 0.001 | 0.998 | 0.001 | 7162 | 7454 | 5814 | 135.00 | 3 |
| 0.001 | 0.001 | 0.998 | 6693 | 4530 | 11,524 | 133.50 | 4 |
| 0.111 | 0.444 | 0.444 | 7180 | 5075 | 11,047 | 140.89 | 5 |
| 0.444 | 0.111 | 0.444 | 7828 | 4543 | 11,131 | 143.42 | 6 |
| 0.444 | 0.444 | 0.111 | 8994 | 5768 | 7471 | 145.76 | 7 |
| Iteration 2 | | | | | | | |
| 0.499 | 0.499 | 0.002 | 8273 | 6858 | 5752 | 139.33 | 8 |
| 0.555 | 0.222 | 0.222 | 9050 | 4784 | 9314 | 148.82† | 9 |
| 0.722 | 0.222 | 0.056 | 9277 | 5442 | 7493 | (146.85) | |
| 0.666 | 0.167 | 0.167 | 9156 | 5049 | 8678 | 148.59 | 10 |
| 0.722 | 0.056 | 0.222 | 9050 | 4751 | 9347 | 148.74 | 11 |
| 0.499 | 0.002 | 0.499 | 7828 | 4543 | 11,131 | (143.42) | |
| Iteration 3 | | | | | | | |
| 0.778 | 0.111 | 0.111 | 9277 | 5442 | 7493 | (146.85) | |
| 0.527 | 0.361 | 0.112 | 9249 | 5547 | 7332 | 146.55 | 12 |
| 0.639 | 0.222 | 0.139 | 9191 | 5184 | 8353 | 148.31 | 13 |
| 0.611 | 0.194 | 0.195 | 9050 | 4751 | 9347 | (148.74) | |
| 0.639 | 0.139 | 0.222 | 9050 | 4751 | 9347 | (148.74) | |
| 0.527 | 0.112 | 0.361 | 8512 | 4543 | 10,339 | 146.10 | 14 |
| Iteration 4 | | | | | | | |
| 0.666 | 0.167 | 0.167 | 9156 | 5049 | 8678 | 148.59 | 15 |
| 0.541 | 0.292 | 0.167 | 9191 | 5184 | 8353 | (148.31) | |
| 0.597 | 0.222 | 0.181 | 9156 | 5049 | 8678 | (148.59) | |
| 0.584 | 0.208 | 0.208 | 9050 | 4784 | 9314 | (148.83) | |
| 0.597 | 0.181 | 0.222 | 9050 | 4751 | 9347 | (148.74) | |
| 0.541 | 0.167 | 0.292 | 9037 | 4751 | 9373 | 148.73 | 16 |
| Iteration 5 | | | | | | | |
| 0.610 | 0.195 | 0.195 | 9050 | 4784 | 9314 | (148.83) | |
| 0.548 | 0.257 | 0.195 | 9143 | 5088 | 8665 | 148.70 | 17 |
| 0.576 | 0.222 | 0.202 | 9050 | 4784 | 9314 | (148.83) | |
| 0.570 | 0.215 | 0.215 | 9050 | 4784 | 9314 | (148.83) | |
| 0.576 | 0.202 | 0.222 | 9050 | 4751 | 9347 | (148.74) | |
| 0.548 | 0.195 | 0.257 | 9050 | 4751 | 9347 | (148.74) | |
| Iteration 6 | | | | | | | |
| 0.552 | 0.240 | 0.208 | 9050 | 4784 | 9314 | (148.83) | |
| 0.566 | 0.212 | 0.222 | 9050 | 4751 | 9347 | (148.74) | |
| 0.552 | 0.208 | 0.240 | 9050 | 4751 | 9347 | (148.74) | |
| 0.562 | 0.219 | 0.219 | 9050 | 4784 | 9314 | (148.83) | |
| 0.561 | 0.217 | 0.222 | 9050 | 4751 | 9347 | (148.74) | |
| 0.554 | 0.215 | 0.231 | 9050 | 4751 | 9347 | (148.74) | |

†Denotes best solution to date.
Parentheses enclose utility of duplicate solutions.

*Iteration 1*

*Step 3: generate new solutions.* Because there are fewer than 5 objectives, the following objective bounds will be used:

$$m_1 = 0.05 \quad m_2 = 0.25.$$

Table 3 provides the right-hand side limits imposed, as well as the resulting utility of that solution. Six new solutions are generated for DM consideration. Go to Step 4.

*Step 4: DM consideration.* The DM would now have $2k$ new solution attainments to compare with the previous solution. The highest utility is provided by Steuer's solution, with a utility value of 146.185. Because of the good starting solution, when returning to Step 3, we assume the DM will choose a finer search pattern, with only one metric. We set $m = 0.01$.

*Iterations 2–4*

Table 3 presents the results of the iterations 2–4, using Step 3 to generate new solutions. In these iterations, therefore, the process switches between Steps 3 and 4. It must be noted that at some iterations, it is necessary to force attainment of the target objective, as the objective function weights pull the LP solution to the next original model corner point. The forced constraints are

Table 3. Bounded constraint results

| | Attainments | | | |
|---|---|---|---|---|
| | Obj 1 | Obj 2 | Obj 3 | Utility |
| Ideal solution | 9277 | 7454 | 11,524 | |
| Starting solution | 9050 | 4784 | 9314 | 148.82 |
| Iteration 1: ($m_1 = 0.05$) | | | | |
| Obj 1 ≥ 9061.35 | 9061.35 | 4779.37 | 9278.91 | 148.69 |
| Obj 2 ≥ 4917.50 | 9074.80 | 4917.50 | 9074.20 | 148.77 |
| Obj 3 ≥ 9424.50 | 9014.11 | 4733.83 | 9424.50 | 148.57 |
| Iteration 1: ($m_2 = 0.25$) | | | | |
| Obj 1 ≥ 9106.75 | 9106.75 | 4892.87 | 9006.5 | 148.53 |
| Obj 2 ≥ 5451.50 | 9064.31 | 5451.50 | 8062.37 | 147.11 |
| Obj 3 ≥ 9866.50 | 8812.68 | 4585.95 | 9866.50 | 147.29 |
| Iteration 2: ($m = 0.01$) | | | | |
| Obj 1 ≥ 9052.27 | 9052.27 | 4756.67 | 9333.38 | 148.73 |
| Obj 2 ≥ 4810.70 | 9041.09 | 4810.70 | 9305.10 | 148.881† |
| Obj 3 ≥ 9336.10 | 9050 | 4751 | 9347 | 148.74 |
| Obj 3 = 9336.10 | 9050.00 | 4761.90 | 9336.10 | 148.77 |
| Iteration 3: ($m = 0.01$) | | | | |
| Obj 1 ≥ 9043.46 | 9050 | 4751 | 9347 | 148.74 |
| Obj 1 = 9043.46 | 9043.46 | 4751 | 9360.09 | 148.74 |
| Obj 2 ≥ 4837.13 | 9042.65 | 4837.13 | 9267.09 | 148.884† |
| Obj 3 ≥ 9327.29 | 9050 | 4770.71 | 9327.29 | 148.79 |
| Iteration 4: ($m = 0.01$) | | | | |
| Obj 1 ≥ 9044.99 | 9050 | 4751 | 9347 | 148.74 |
| Obj 1 = 9044.99 | 9044.99 | 4751 | 9357.02 | 148.74 |
| Obj 2 ≥ 4863.30 | 9053.12 | 4863.30 | 9204.28 | 148.84 |
| Obj 3 ≥ 9289.66 | 9050 | 4751 | 9347 | 148.74 |
| Obj 3 = 9289.66 | 9054.06 | 4794.14 | 9289.66 | 148.81 |

†Denotes best solution to date.

indicated in Table 3 by the strict equality. The original model corner point obtained is given in Table 3 as well.

In iteration 4, the highest utility found among the newly generated solutions was less than that of the solution obtained at iteration 3. In fact, from a decision making standpoint, the added utility of the solution obtained at iteration 3 is probably difficult to distinguish from that of iteration 2. Whenever the DM is satisfied that further iteration would not yield improved solutions, the process would stop. Improving the best solution after iteration 4 would require an $m$ value less than 0.01, so we assume the DM will stop.

Step 6 yields the report of the current alternative, as well as the maximum attainments and worst nondominated attainments identified for each objective. The final solution is not at an original model corner point, and has a utility value exceeding that of all original model corner points.

## 5. CONCLUSIONS

A major element of real decision making is that decision makers may well have nonlinear utility functions, although they are not likely to be expressible. The nonlinear programming problem is therefore compounded with the difficulty that the nonlinear objective function (utility) is not known. The search procedure presented is basically a nonlinear programming method, seeking the optimal utility function value subject to the constraint set, without expressing the utility function. The entire procedure can be supported with a linear programming code, using decision maker choice to direct the nonlinear search.

The procedure presented closely follows Steuer's method, with the relative benefit of utilizing an unmodified linear programming code. A major problem with Steuer's method is that it utilizes model corner points, and if utility is nonlinear, may well not consider the true optimal alternatives. Steuer's method operates by generating new solutions, and then filtering them down to the desired number of alternatives which are sufficiently diverse. With the proposed solution procedure, new alternatives are generated through constraints on objective attainment, creating new corner points in the nondominated set. Steuer's method will provide good solutions given the requirement for being at an original model corner point. The method we present works as a means to fine tune Steuer's method.

Our example operated by constraining each objective one at a time. Lower bounds on particular objectives could easily be incorporated to reflect additional information the decision maker might want to include.

The proposed procedure is contended to be quite simple, requiring commonly available linear programming computer support, and giving a means for decision makers to more thoroughly analyze their alternatives in complex decision tasks. It is generally accepted that rational decision makers should balance tradeoffs by maximizing their utility. However, utility is often difficult to express directly. Through use of linear programming models, decision makers can use the procedure presented to generate new alternatives, and balance conflicting objectives to improve their utility.

## REFERENCES


1. D. J. White, Multi-objective interactive programming. *J. opl Res. Soc.* **31,** 517–523 (1980).
2. S. Zionts and J. Wallenius, Identifying efficient vectors: some theory and computational results. *Opns Res.* **28,** 788–793 (1980).
3. S. Zionts and J. Wallenius, An interactive multiple objective linear programming method for a class of underlying nonlinear functions. *Mgmt Sci.* **29,** 519–529 (1983).
4. S. De Samblanckx, P. Depraetere and H. Muller, Critical considerations concerning the multicriteria analysis by the method of Zionts and Wallenius. *Eur. J. Op. Res.* **10,** 70–66 (1982).
5. R. E. Steuer, An interactive multiple-objective linear programming procedure. *TIMS Stud. Mgmt Sci.* **6,** 225–239 (1977).
6. R. E. Steuer, Vector-maximum gradient cone contraction techniques. In *Multiple Criteria Problem Solving: Proceedings—Buffalo, 1977* (Ed. S. Zionts), pp. 462–481. Springer, New York (1978).
7. R. E. Steuer and F. W. Harris, Intra-set point generation and filtering in decision and criterion space. *Comput. Opns Res.* **7,** 41–53 (1980).
8. K. R. Balachandran and R. E. Steuer, An interactive model for the CPA firm audit staff planning problem with multiple objectives. *Accounting Rev.* **57,** 125–140 (1982).
9. R. E. Steuer and A. T. Schuler, An interactive multiple-objective linear programming approach to a problem in forest management. *Opns Res.* **26,** 254–269 (1978).
10. R. E. Steuer and M. J. Wallace Jr, An interactive multiple objective wage and salary administration procedure. In *Personnel Management: A Computer-Based System* (Eds S. M. Lee and C. D. Thorp), pp. 159–176. Petrocelli, New York (1978).
11. R. E. Steuer, Multiple criterion function goal programming applied to managerial compensation planning. *Comput. Opns Res.* **10,** 299–309 (1983).
12. R. E. Steuer, Sausage blending using multiple objective linear programming. *Mgmt Sci.* **30,** 1376–1384 (1984).
13. R. Benayoun, J. De Montgolfier, J. Tergny and D. Larichev, Linear programming with multiple objective functions: Step method (STEM). *Math. Program.* **1,** 366–375 (1971).
14. Y. Y. Haimes, W. Hall and H. Freedman, *Multiobjective Optimization in Water Resources System: The Surrogate Worth Trade-Off Method.* Elsevier, Amsterdam (1975).
15. S. I. Gass, The setting of weights in linear goal-programming problems. *Comput. Opns Res.* **14,** 227–229 (1987).
16. D. L. Olson, M. Venkataramanan and J. Mote, A technique using analytic hierarchy process in multiobjective planning models. *Socio-econ. Plan. Sci.* **20,** 361–368 (1986).
17. H. R. Weistroffer, Careful usage of pessimistic values is needed in multiple objective optimization. *Opns Res. Lett.* **4,** 23–25 (1985).
18. T. L. Saaty, A scaling method for priorities in hierarchical structures. *J. math. Psychol.* **15,** 234–281 (1977).


## APPENDIX

*Objectives*

$$z_1 = 5x_{27} + 7x_{28} + 8x_{29} + 7x_{37} + 8x_{38} + 10x_{39} + 9x_{47} + 10x_{48} + 10x_{49} + 8x_{57} + 9x_{58} + 15x_{59} + 12x_{67} + 10x_{68} + 16x_{69} + 11x_{210} + 11x_{211} + 14x_{310} + 17x_{311} + 15x_{410} + 19x_{411} + 18x_{510} + 24x_{511} + 20x_{610} + 23x_{611};$$

$$z_2 = 13x_{27} + 10x_{28} + 11x_{29} + 9x_{37} + 7x_{38} + 7x_{39} + 9x_{47} + 8x_{48} + 10x_{49} + 5x_{57} + 1x_{58} + 2x_{59} + 8x_{67} + 5x_{68} + 7x_{69} + 15x_{210} + 19x_{211} + 10x_{310} + 17x_{311} + 13x_{410} + 22x_{411} + 5x_{510} + 13x_{511} + 11x_{610} + 15x_{611};$$

$$z_3 = 10x_{27} + 8x_{28} + 13x_{29} + 23x_{37} + 16x_{38} + 21x_{39} + 13x_{47} + 5x_{48} + 18x_{49} + 20x_{57} + 13x_{58} + 24x_{59} + 15x_{67} + 3x_{68} + 10x_{69} + 4x_{210} + 17x_{211} + 12x_{310} + 25x_{311} + 2x_{410} + 13x_{411} + 17x_{510} + 25x_{511} + 8x_{610} + 18x_{611}.$$

max $W_1 z_1 + W_2 z_2 + W_3 z_3$

s.t.

$$x_{12} + x_{13} + x_{14} + x_{15} + x_{16} = 500$$

$$-x_{12} + x_{27} + x_{28} + x_{29} + x_{210} + x_{211} = 0$$

$$-x_{13} + x_{37} + x_{38} + x_{39} + x_{310} + x_{311} = 0$$

$$-x_{14} + x_{47} + x_{48} + x_{49} + x_{410} + x_{411} = 0$$

$$-x_{15} + x_{57} + x_{58} + x_{59} + x_{510} + x_{511} = 0$$

$$-x_{16} + x_{67} + x_{68} + x_{69} + x_{610} + x_{611} = 0$$

$$x_{27} + x_{37} + x_{47} + x_{57} + x_{67} - x_{712} = 0$$
$$x_{28} + x_{38} + x_{48} + x_{58} + x_{68} - x_{812} = 0$$
$$x_{29} + x_{39} + x_{49} + x_{59} + x_{69} - x_{912} = 0$$
$$x_{210} + x_{310} + x_{410} + x_{510} + x_{610} - x_{1012} = 0$$
$$x_{211} + x_{311} + x_{411} + x_{511} + x_{611} - x_{1112} = 0$$
$$x_{712} + x_{812} + x_{912} + x_{1012} + x_{1112} = 500$$

$$\begin{array}{llll}
x_{12} \leqslant 125 & x_{47} \leqslant 86 & x_{310} \leqslant 123 & x_{712} \leqslant 209 \\
x_{13} \leqslant 253 & x_{48} \leqslant 44 & x_{311} \leqslant 139 & x_{812} \leqslant 87 \\
x_{14} \leqslant 171 & x_{49} \leqslant 65 & x_{410} \leqslant 86 & x_{912} \leqslant 130 \\
x_{15} \leqslant 264 & x_{57} \leqslant 105 & x_{411} \leqslant 86 & x_{1012} \leqslant 246 \\
x_{16} \leqslant 72 & x_{58} \leqslant 44 & x_{510} \leqslant 123 & x_{1112} \leqslant 178 \\
x_{27} \leqslant 63 & x_{59} \leqslant 65 & x_{511} \leqslant 89 & \\
x_{28} \leqslant 44 & x_{67} \leqslant 36 & x_{610} \leqslant 36 & \\
x_{29} \leqslant 63 & x_{68} \leqslant 36 & x_{611} \leqslant 36 & \\
x_{37} \leqslant 105 & x_{69} \leqslant 36 & & \\
x_{38} \leqslant 44 & x_{210} \leqslant 63 & & \\
x_{39} \leqslant 65 & x_{211} \leqslant 63 & &
\end{array}$$

$W_k > 0$ for $k = 1$ to 3, $x_j \geqslant 0$ for $j = 1$ to the number of variables.