

Theory and Methodology

Unified interactive multiple objective programming

Lorraine R. Gardiner

Department of Management, Auburn University, Auburn, AL 36849, USA

Ralph E. Steuer

Department of Management Science and Information Technology, Brooks Hall, University of Georgia, Athens, GA 30602, USA

Received August 1991; revised February 1992

Abstract: Previous research has stressed the philosophical, strategic, and taxonomical differences among the procedures of interactive multiple objective programming. In this paper, however, we pursue a completely different tack by examining the all but overlooked implementation similarities of the interactive procedures. In doing so, the paper demonstrates how the preponderance of interactive procedures can be made to fit a single algorithmic outline. Then, by introducing the concept of the *unified sampling program*, and using only a modest inventory of component subroutines, it is shown how these procedures can be unified into a single algorithmic product. In addition, the unified approach of this paper supports *procedure-switching*, thus enabling a user to start with one procedure and switch to other procedures during the interactive process if so desired. A unified algorithm embracing STEM, the Geoffrion–Dyer–Feinberg procedure, interactive goal programming, Wierzbicki's Aspiration Criterion Vector method, and five other prominent procedures is used to illustrate the degree to which the procedures of interactive multiple objective programming can be folded into one another when pursuing the unified strategy.

Keywords: Multiple criteria decision making; Multiple criterion optimization; Interactive multiple objective programming; Interactive procedures

1. Introduction

Numerous interactive procedures have been developed for solving the multiple objective pro-

gramming problem

$$\begin{aligned} &\max \quad \{f_1(x) = z_1\} \\ &\quad \vdots \\ &\max \quad \{f_k(x) = z_k\} \\ &\text{s.t.} \quad x \in S, \end{aligned}$$

Correspondence to: Prof. R.E. Steuer, Department of Management Science and Information Technology, Brooks Hall, University of Georgia, Athens, GA 30602, USA.

in which the f_i are bounded over S and the DM's (decision maker's) utility function $U: R^k \rightarrow R$ is

coordinatewise increasing ('more is always better than less of each criterion'). These procedures include [2,6,8–10,12–15,17,18,20,21,23–25,27,30–38,40,45,47,48,50,53,55,57,58]. While this list is long, the consensus is that a range of interactive procedures is necessary because the most appropriate one to use is often application and user decision-making style dependent. Aksoy [1] provides a recent bibliography on interactive multiple objective programming, and recent reviews of interactive multiple objective programming are found in Galas, Nykowski and Zolkiewski [16, Section 4.5], Vincke [49, Chapter 6], Zionts [56], and Shin and Ravindran [39].

Whereas previous research has stressed the philosophical, strategic, and taxonomical differences among the procedures of interactive multiple objective programming, the purpose of this paper is to exploit the all but overlooked implementation similarities of the procedures so as to construct a *unified algorithm* out of which can fall as special cases most of the prominent procedures of interactive multiple objective programming. This paper represents the culmination of unifying ideas which the authors have presented at professional meetings since the mid-1980s (e.g., Steuer and Gardiner [46]).

It is possible to build such a unified algorithm because virtually all of the procedures of interactive multiple objective programming more or less follow the same *general algorithmic outline*. As portrayed in Figure 1, the general algorithmic outline includes

- (1) the setting of *controlling parameters* (e.g., weights, multipliers, criterion value lower bounds, aspiration criterion vectors) for the first iteration;
- (2) the optimization of one or more mathematical programming problems (called *sampling programs*);
- (3) the examination of the criterion vector results;
- (4) the resetting of controlling parameters for the next iteration.

Another observation is that the sampling programs of most of the procedures can be viewed as special cases of a more general formulation, called a *unified sampling program*. Using the unified sampling program in Step (2) of the general algorithmic outline, a *unified algorithm* is proposed. To demonstrate how a spectrum of interactive multiple objective programming procedures can

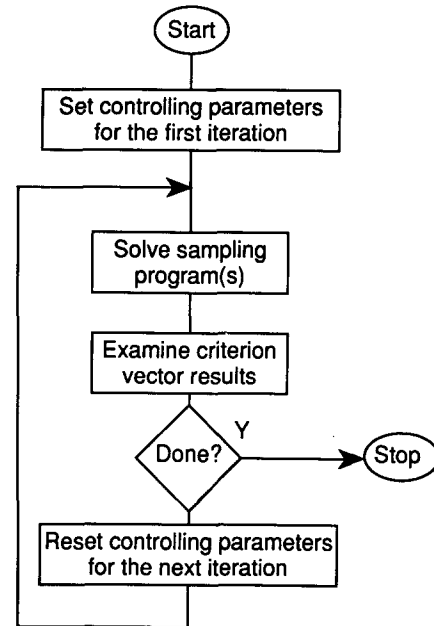


Figure 1. General algorithmic outline

be included, this paper shows how the following nine well-known interactive procedures can be incorporated into the unified algorithm.

1. ECON: e-Constraint Method (see for example, Chankong and Haimes [7, Section 6.4]).
2. STEM: Benayoun, de Montgolfier, Tergny and Larichev [2].
3. GDF: Geoffrion–Dyer–Feinberg Procedure [17].
4. IGP: Interactive Goal Programming (see for example, Dyer [10]; Franz and Lee [14]; Spronk [41]).
5. WIERZ: Aspiration Criterion Vector Method (see Wierzbicki [52–54]; Lewandowski and Grauer [26]; Lewandowski and Wierzbicki [28]).
6. TCH: Tchebycheff Procedure: Augmented and Lexicographic Versions (Steuer and Choo [45]; updated descriptions in Steuer [43]).
7. SATIS: Satisficing Tradeoff Method (Nakayama and Sawaragi [35]; updated description in Nakayama [34]).
8. VIA: Visual Interactive Approach (Korhonen and Laakso [21,22]).
9. RACE: Pareto Race (Korhonen and Wallenius [23]).

With ECON and STEM being *reduced feasible region* methods; TCH being a *reduced weighting*

vector space method; GDF, VIA and RACE being line search methods; and IGP, WIERZ and SATIS being reference point methods, these procedures are representative of the diversity of procedures that can be included. While most of the other procedures of interactive multiple objective programming could be similarly incorporated, there are a few procedures that we are not recommending at the present time for inclusion in the unified algorithm because of special requirements. One is the Zionts–Wallenius Procedure [57,58] because of the way efficient nonbasic variables are identified. Another is the procedure of Marcotte and Soland [32] because of its branch and bound intricacies.

In addition to including multiple procedures under one umbrella, the unified algorithm supports *procedure-switching*. That is, a user may switch from one procedure to another, one iteration to the next, custom-tailoring the search process. For example, TCH might be used as a starfinder in the beginning iterations to locate a promising area of search, RACE might be used in the middle iterations to explore the promising area of search, and WIERZ might be used in the ending iterations to pinpoint a final solution. Comparative studies of interactive multiple objective programming procedures (Brockhoff [4]; Buchanan and Daellenbach [5]) indicate that hybrid approaches are attractive to decision makers.

The remainder of the paper is organized as follows. Section 2 presents the small amount of background material required to make the unified approach self-contained. Section 3 specifies the unified algorithm and demonstrates the small inventory of *subroutine building blocks* required to fuel the unified procedure. Section 4 presents the unified sampling program and discusses the robustness of its customizations. Sections 5, 6, 7 and 8 show how the nine prominent interactive procedures listed above can be included in the unified algorithm. Section 9 contains concluding remarks.

2. Background concepts

Characterizing the modest mathematical background necessary for driving the unified approach, we have the following concepts and terminology. In the multiple objective program

$$\begin{aligned} &\max \quad \{f_1(x) = z_1\} \\ &\quad \vdots \\ &\max \quad \{f_k(x) = z_k\} \\ &\text{s.t.} \quad x \in S \end{aligned}$$

the z_i are *criterion values* and S is the feasible region in *decision space*. Let $Z \subset R^k$ be the feasible region in *criterion space* where $z \in Z$ if and only if there exists an $x \in S$ such that $(f_1(x), \dots, f_k(x)) = z$. Let $K = \{1, \dots, k\}$. *Criterion vector* $\bar{z} \in Z$ is *nondominated* if and only if there does not exist another $z \in Z$ such that $z_i \geq \bar{z}_i$ for all $i \in K$ and $z_i > \bar{z}_i$ for at least one $i \in K$. And criterion vector $\bar{z} \in Z$ is *weakly-nondominated* if and only if there does not exist another $z \in Z$ such that $z_i > \bar{z}_i$ for all $i \in K$. The set of all nondominated criterion vectors is designated N and is called the *nondominated set*. The set of weakly nondominated criterion vectors is designated N^w and is called the *weakly-nondominated set*. If $z^0 \in Z$ maximizes the DM's utility function U over Z , z^0 is an *optimal criterion vector*. Our interest in the nondominated set N stems from the fact that when U is coordinate-wise increasing, $z^0 \in N$. We are sometimes interested in $N^w \supset N$ because N^w , when different from N , is usually only slightly larger (see Steuer [42, Section 9.4]). In this paper we will also make reference to the following:

A *payoff table* is of the form

	z_1	z_2		z_k
z^1	z_{11}^*	z_{12}		z_{1k}
z^2	z_{21}	z_{22}^*		z_{2k}
			\ddots	
z^k	z_{k1}	z_{k2}		z_{kk}^*

where the rows are criterion vectors resulting from individually maximizing the objectives. The maximal values z_i^* along the main diagonal of the payoff table form the $z^* \in R^k$ *reference criterion vector*. The components of a $z^{**} \in R^k$ *reference criterion vector* are given by

$$z_i^{**} = z_i^* + \varepsilon_i$$

where the ε_i are moderately small positive values (see Steuer [42, pp. 420–423]).

Let *multiplier space* be

$$\Lambda = \left\{ \lambda \in R^k \mid \lambda_i \in (0, 1), \sum_{i \in K} \lambda_i = 1 \right\}.$$

Among the mathematical programming problems employed by the interactive procedures in Step (2) of the general algorithmic outline for probing the nondominated set are the *weighted-sums* sampling program

$$\begin{aligned} \max \quad & \{\lambda^T z\} \\ \text{s.t.} \quad & f_i(x) = z_i, \quad i \in K, \\ & x \in S, \\ & z \in R^k \text{ unrestricted,} \end{aligned}$$

the *lexicographic Tchebycheff* sampling program

$$\begin{aligned} \text{lex min} \quad & \left\{ \alpha, - \sum_{i \in K} z_i \right\} \\ \text{s.t.} \quad & \alpha \geq \lambda_i(z_i^{**} - z_i), \quad i \in K, \\ & f_i(x) = z_i, \quad i \in K, \\ & x \in S, \\ & 0 \leq \alpha \in R, z \in R^k \text{ unrestricted,} \end{aligned}$$

and the *augmented Tchebycheff* sampling program

$$\begin{aligned} \min \quad & \left\{ \alpha - \rho \sum_{i \in K} z_i \right\} \\ \text{s.t.} \quad & \alpha \geq \lambda_i(z_i^{**} - z_i), \quad i \in K, \\ & f_i(x) = z_i, \quad i \in K, \\ & x \in S, \\ & 0 \leq \alpha \in R, z \in R^k \text{ unrestricted,} \end{aligned}$$

in which, when not zero, ρ is a small positive scalar and $\lambda = (\lambda_1, \dots, \lambda_k)$. Without loss of generality, λ is usually a member of Λ , but is sometimes a member of $\bar{\Lambda} = \{\lambda \in R^k \mid \lambda_i \geq 0\}$.

To explain how the lexicographic and augmented Tchebycheff sampling programs generate nondominated criterion vectors, let $\bar{\lambda} \in \Lambda$. Consider the unbounded line segment emanating from

z^{**} in the direction

$$-\left(\frac{1}{\bar{\lambda}_1}, \dots, \frac{1}{\bar{\lambda}_k}\right).$$

Now imagine the *closest intersecting (translated) nonnegative orthant* (CITNO) associated with $\bar{\lambda} \in \Lambda$. This is the nonnegative orthant of R^k whose origin has been translated to a point on the unbounded line segment *closest* to z^{**} such that the translated nonnegative orthant still intersects Z . Visualizing the lexicographic Tchebycheff sampling program, a criterion vector solution to the first optimization stage is a point in Z on the CITNO. Among the points in Z on the CITNO, at least one is nondominated. If there is only one point \bar{z} in Z on the CITNO, we are done. If not, the second optimization stage is invoked in order to terminate with a point \bar{z} in Z on the CITNO that is also closest to z^{**} according to the L_1 metric. In this way, the criterion vector $\bar{z} \in N$ generated by the lexicographic Tchebycheff sampling program depends upon $\bar{\lambda} \in \Lambda$.

Although \bar{z} may not be at the vertex of the CITNO associated with $\bar{\lambda}$, \bar{z} will be at the vertex of the CITNO associated with $\lambda \in \Lambda$ where

$$\lambda_i = \frac{1}{(z_i^{**} - \bar{z}_i)} \left[\sum_{j \in K} \frac{1}{(z_j^{**} - \bar{z}_j)} \right]^{-1}.$$

When $\bar{z} < z^{**}$, a $\lambda \in \Lambda$ that satisfies the above relationship is said to be the *T-vertex* λ -vector defined by z^{**} and \bar{z} .

The augmented Tchebycheff sampling program is designed to simulate the lexicographic Tchebycheff sampling program. Because the tie-breaking feature of the second optimization stage of the lexicographic Tchebycheff program is incorporated into the objective function of the augmented Tchebycheff program as a perturbation term, sometimes slightly different criterion vectors are generated for the same λ -vector. Nevertheless, as long as $\rho > 0$, the criterion vector generated is nondominated.

In TCH, in particular, the λ -vectors in the sampling programs come from *interval defined subsets*

$$\Lambda^{(h)} = \left\{ \lambda \in R^k \mid \lambda_i \in (l_i^{(h)}, \mu_i^{(h)}), \sum_{i \in K} \lambda_i = 1 \right\}$$

of multiplier space Λ where h is the iteration number;

$$\mu_i^{(h)} - l_i^{(h)} = \mu_j^{(h)} - l_j^{(h)}$$

for all $i \neq j$; and

$$(l_i^{(h)}, \mu_i^{(h)}) \subset (0, 1)$$

for all $i \in K$. Let $\bar{\lambda} \in \Lambda$ and $\eta^{(h)} \in (0, 1]$ be a Λ size factor. Then, using the routine of Liou [29], an interval defined subset $\Lambda^{(h)}$ centered about $\bar{\lambda}$ can be created such that $\Lambda^{(h)}$ is $\eta^{(h)}$ of the size of Λ . This is accomplished by computing the half-width r associated with $\bar{\lambda}$ and $\eta^{(h)}$ in order to define the intervals

$$(l_i^{(h)}, \mu_i^{(h)}) = \begin{cases} (0, 2r) & \text{if } \bar{\lambda}_i - r \leq 0, \\ (1 - 2r, 1) & \text{if } \bar{\lambda}_i + r \geq 1, \\ (\bar{\lambda}_i - r, \bar{\lambda}_i + r) & \text{otherwise.} \end{cases}$$

Note that throughout the paper we will be using the convention that superscripts contained in parentheses indicate iteration numbers. In order to obtain small sets of λ -vectors *well-spaced* over $\Lambda^{(h)}$, we use the LAMBDA code from ADBASE (Steuer [44]) to generate an overabundance of points uniformly distributed over $\Lambda^{(h)}$. Then the *first point outside the neighborhood* heuristic (Steuer [42] and White [51]) can be used to 'filter' the points to obtain a set of λ -vectors well-spaced over $\Lambda^{(h)}$.

3. Unified algorithm and subroutine building blocks

As an elaboration of the general algorithmic outline, the steps of the unified interactive multiple objective programming algorithm are as follows:

Step 0. DM specifies interactive procedure for the first iteration.

Step 1. $h = 0$.

Step 2. Initialize current interactive procedure.

Step 3. $h = h + 1$. DM answers questions posed by the current interactive procedure.

Table 1
Initialization routines

Label	Routine
I-1	Construct a payoff table and form z^* reference criterion vector.
I-2	$H = I_R = \emptyset$.
I-3	DM specifies an initial feasible criterion vector $z^{(0)}$.
I-4	DM (or analyst) specifies ρ .
I-5	Form a z^{**} reference criterion vector.
I-6	DM (or analyst) specifies oversampling factor ν .
I-7	Let $(l_i^{(1)}, \mu_i^{(1)}) = (0, 1)$ for each $i \in K$.
I-8	DM estimates criterion value range widths r_i for each of the objectives over N^w .
I-9	For each $i \in K$, let r_i equal the difference between z_i^* and the minimum value in the i -th column of the payoff table.
I-10	Let step size $s^{(1)} = 0$, the initial direction vector $d^{(1)}$ be set according to the rules in [23, p. 619], and $I_R = \emptyset$.
I-11	$s^{(2)} = b$ where b is the default base speed [23, p. 619].

Step 4. Configure and solve unified sampling program(s).

Step 5. Present resulting criterion vector(s) and related information (if any) to the DM.

Step 6. DM selects $z^{(h)}$.

Table 2
Calculation routines

Label	Routine
C-1	Calculate $\lambda \in \bar{\Lambda}$ according to the rules in [2, p. 370].
C-2	DM specifies a weighting vector $\lambda \in \bar{\Lambda}$, or one is calculated from his or her responses (Dyer [11, p. 207]) that is reflective of the local marginal tradeoffs at $z^{(h-1)}$.
C-3	The line between $z^{(h-1)}$ and $y^{(h)}$ is divided into segments so as to create P equally spaced criterion vectors, where $y^{(h)}$ is an optimal criterion vector solution to a weighted-sums sampling program.
C-4	Compute T-vertex λ -vector.
C-5	Compute the $(l_i^{(h)}, \mu_i^{(h)})$ for all $i \in K$ associated with the T-vertex λ -vector and $\eta^{(h)}$.
C-6	Form $\Lambda^{(h)} = \{\lambda \in R^K \mid \lambda_i \in (l_i^{(h)}, \mu_i^{(h)}), \sum_{i \in K} \lambda_i = 1\}$.
C-7	Randomly generate uniformly distributed λ -vectors over $\Lambda^{(h)}$.
C-8	Filter a group of vectors down to a reduced number of well-spaced representatives.
C-9	Let aspiration criterion value $q_i^{(h)} = z_i^{(h-1)}$ for each $i \in I_A$, and using dual variable information, calculate $q_i^{(h)}$ for each $i \in I_R$.
C-10	Compute λ -vector such that $\lambda_i = 1/r_i$, for each $i \in K$.
C-11	Let direction vector $d^{(h)} = q^{(h)} - z^{(h-1)}$.

Table 3
Elicitation routines

Label	Routine
E-1	DM selects a primary objective $f_j(x) = z_j$.
E-2	$H = K - \{j\}$ and DM specifies criterion value lower bounds e_i , for each $i \in H$.
E-3	DM specifies (a) the set of indices I_R of the components of $z^{(h)}$ to be relaxed and (b) the maximum amounts of relaxation Δz_i , for each $i \in I_R$.
E-4	$H = K$, $e_i = z_i^{(h)} - \Delta z_i$, for each $i \in I_R$, and $e_i = z_i^{(h)}$, for each $i \in (H - I_R)$.
E-5	DM specifies P , the number of criterion vectors to be presented.
E-6	DM selects the most preferred of the presented criterion vectors and designates it $z^{(h)}$.
E-7	DM specifies (a) the number of lexicographic levels L , (b) the target values t_i , $i \in I$, and u_i , $i \in J$, and (c) the within lexicographic level weights $w_{(m)i}^-$, $m = 1, \dots, L$, $i \in I$, and $w_{(m)i}^+$, $m = 1, \dots, L$, $i \in J$.
E-8	DM specifies an aspiration criterion vector.
E-9	DM specifies a size factor $\eta^{(h)}$.
E-10	DM specifies (a) index set I_I of components of $z^{(h-1)}$ to be improved, (b) index set I_R of components to be relaxed, and (c) index set I_A of components to be held at their current values.
E-11	DM specifies $q_i^{(h)}$ for each $i \in I_I$.
E-12	DM may further adjust components of $q^{(h)}$ if desired.
E-13	DM selects the most preferred criterion vector described by the trajectories and designates it $z^{(h)}$.
E-14	Either the DM presses microcomputer keys to specify (a) an updated $s^{(h)}$, (b) an updated $d^{(h)}$, (c) an updated I_R and associated criterion value lower bounds e_i , or we let (i) $s^{(h)} = s^{(h-1)}$, (ii) $d^{(h)} = d^{(h-1)}$, (iii) I_R and the associated e_i remain as they are.

Step 7. If the DM wishes to switch to a different interactive procedure, go to Step 9. Otherwise, DM answers any further questions posed by current interactive procedure.

Step 8. If the unified algorithm is to be terminated at this point, go to Step 10. Otherwise, go to Step 3.

Table 4
Presentation routines

Label	Routine
P-1	Present criterion vector(s) and related information (if any) to the DM.
P-2	Present the criterion value trajectory for each objective to the DM.

Table 5
Transfer of control routines

Label	Routine
T-1	Continue.
T-2	If the DM wishes to cease iterating,
T-3	If $I_R = \emptyset$.
T-4	Go to Step 3.
T-5	If the DM does not wish to make further changes to the number of lexicographic levels, target values, or within lexicographic level weights,
T-6	Pause until DM hits space bar.

Step 9. The DM specifies the new interactive procedure. Go to Step 2.

Step 10. Stop with $z^{(h)}$ as the final solution.

At execution time, each step of the unified algorithm is made up of a sequence of subroutines which depend upon the interactive procedure currently in effect. The subroutines can be thought of as coming from five categories and are summarized in Tables 1–5. While these tables contain some terms not yet defined, brief verbal descriptions are given. Terms will become clear when used in Sections 5–8. Additional index sets used in the subroutines are: H , I , J , I_A , I_I , $I_R \subset K$.

In the second category, we have the calculation routines of Table 2 which are used to calculate λ -vectors and perform other computational chores. In the third category, we have the elicitation routines of Table 3 which are to establish values for the controlling parameters. In the fourth and fifth categories, we have the presentation and transfer of control routines of Tables 4 and 5.

4. Unified sampling program and its customizations

The unified sampling program is at the heart of the unified algorithm. Its feature is that it can be straightforwardly customized to form the sampling programs of the interactive procedures included in the unified algorithm. The unified sam-

pling program formulation is given by

$$\begin{aligned}
 \text{lex min} \quad & \{s_1(z, x, \alpha), s_2(z, x, \alpha), \dots, \\
 & s_L(z, x, \alpha)\} \\
 \text{s.t.} \quad & \alpha \geq \lambda_i(q_i + \theta d_i - z_i), \quad i \in G, \\
 & z_i \geq e_i \quad i \in H, \\
 & z_i + d_i^- \geq t_i, \quad i \in I, \\
 & z_i - d_i^+ \leq u_i, \quad i \in J, \\
 & f_i(x) = z_i, \quad i \in K, \\
 & x \in S, \\
 & 0 \leq \alpha \in R, \quad z \in R^k \text{ unrestricted}, \\
 & d^-, d^+ \geq 0,
 \end{aligned}$$

where

$s_m(z, x, \alpha)$ = Scalar-valued function of m -th lexicographic level.

L = Number of lexicographic levels.

λ = Weighting or multiplier vector.

q = Aspiration criterion vector.

d = Direction vector.

θ = RHS step size.

e = Vector of criterion value lower bounds.

t = Vector of satisficing target values.

u = Vector of saturation target values.

and $G, H, I, J \subset K$. The scalar-valued function of the m -th lexicographic level is given by

$$\begin{aligned}
 s_m(z, x, \alpha) = & \sigma_{(m)}\alpha - \rho_{(m)} \sum_{i \in K} \mu_{(m)i} z_i \\
 & + \tau_{(m)} \left[\sum_{i \in I} w_{(m)i}^- d_i^- + \sum_{i \in J} w_{(m)i}^+ d_i^+ \right]
 \end{aligned}$$

where $\sigma_{(m)}$, $\rho_{(m)}$ and $\tau_{(m)}$ are procedure-dependent constants which serve mostly as 'switches' to include portions of the scalar-valued functions, $\mu_{(m)}$ is a vector of weighting coefficients, and the $w_{(m)}^-$ and $w_{(m)}^+$ are vectors of penalty weights for deviations from target values. The controlling parameters which may be manipulated in the unified sampling program are $\lambda, q, d, \theta, e, t, u$ and $\mu_{(m)}, w_{(m)}^-, w_{(m)}^+$, for $m = 1, \dots, L$.

Customizing the unified sampling program to form the *e-constraint* sampling program used in ECON, we let

(1) $L = 1$. There is only one lexicographic level.

(2) $\sigma_{(1)} = 0$, $\rho_{(1)} = 1$, $\tau_{(1)} = 0$ and vector $\mu_{(1)}$ is replaced by the j -th column of a $k \times k$ identity matrix where $f_j(x)$ has been selected as the *primary objective*.

(3) $G = I = J = \emptyset$. There are no minimax or goal constraints.

(4) $H = K - \{j\}$. There is a lower bound constraint for each objective function where $i \neq j$.

From this emerges

$$\begin{aligned}
 \min \quad & \{-z_j\} \\
 \text{s.t.} \quad & z_i \geq e_i, \quad i \in K - \{j\}, \\
 & f_i(x) = z_i, \quad i \in K, \\
 & x \in S, \\
 & z \in R^k \text{ unrestricted}.
 \end{aligned}$$

Customizing the unified sampling program to form the *minimax e-constraint* sampling program used in STEM, we let

(1) $L = 1$. There is only one lexicographic level.

(2) $\sigma_{(1)} = 1$ and $\rho_{(1)} = \tau_{(1)} = 0$.

(3) $G = K$, $q = z^*$ and $\theta = 0$.

(4) $H = K$.

(5) $I = J = \emptyset$. There are no goal constraints.

From this emerges

$$\begin{aligned}
 \min \quad & \{\alpha\} \\
 \text{s.t.} \quad & \alpha \geq \lambda_i(z_i^* - z_i), \quad i \in K, \\
 & z_i \geq e_i, \quad i \in K, \\
 & f_i(x) = z_i, \quad i \in K, \\
 & x \in S, \\
 & 0 < \alpha \in R, \quad z \in R^k \text{ unrestricted}.
 \end{aligned}$$

Customizing the unified sampling program to form the *weighted-sums* sampling program mentioned in Section 2 that is used in GDF, we let

(1) $L = 1$.

(2) $\sigma_{(1)} = 0$, $\rho_{(1)} = 1$, $\tau_{(1)} = 0$ and vector $\mu_{(1)} \in R^k$ is replaced by a vector of local marginal tradeoff weights $\lambda \in \bar{\Lambda}$.

(3) $G = H = I = J = \emptyset$.

Customizing the unified sampling program to form the *goal programming* sampling program used in IGP, we let

(1) L be the number of lexicographic levels defined by the DM.

(2) $\sigma_{(m)} = \rho_{(m)} = 0$ and $\tau_{(m)} = 1$ for each $m = 1, \dots, L$.

(3) I be the set of indices of criterion values assigned *greater than or equal to* target values.

(4) J be the set of indices of criterion values assigned *less than or equal to* target values.

(5) The $w_{(m)i}^-$, $i \in I$, and the $w_{(m)i}^+$, $i \in J$, for all $m = 1, \dots, L$, be the *within lexicographic level* weights.

(6) $G = H = \emptyset$. There are no minimax or e-constraints.

From this emerges

$$\begin{aligned} \text{lex min} \quad & \left\{ \sum_{i \in I} w_{(1)i}^- d_i^- + \sum_{i \in J} w_{(1)i}^+ d_i^+, \right. \\ & \left. \dots, \sum_{i \in I} w_{(L)i}^- d_i^- + \sum_{i \in J} w_{(L)i}^+ d_i^+ \right\} \\ \text{s.t.} \quad & z_i + d_i^- \geq t_i, \quad i \in I, \\ & z_i - d_i^+ \leq u_i, \quad i \in J, \\ & f_i(x) = z_i, \quad i \in K, \\ & x \in S, \\ & z \in R^k \text{ unrestricted,} \\ & d^-, d^+ \geq 0. \end{aligned}$$

Customizing the unified sampling program to form the augmented and lexicographic Tchebycheff sampling programs mentioned in Section 2 that are used in WIERZ, TCH and SATIS, we let in the augmented case

- (1) $L = 1$.
- (2) $\sigma_{(1)} = 1$, $\rho_{(1)} = \rho$, $\mu_{(1)i} = 1$ for each $i \in K$, and $\tau_{(1)} = 0$.
- (3) $G = K$, $q = z^{**}$ and $\theta = 0$.
- (4) $H = I = J = \emptyset$.

In the lexicographic case, we let

- (1) $L = 2$.
- (2) $\sigma_{(1)} = 1$ and $\rho_{(1)} = \tau_{(1)} = 0$.
- (3) $\sigma_{(2)} = 0$, $\rho_{(2)} = 1$, $\mu_{(2)i} = 1$ for each $i \in K$, and $\tau_{(2)} = 0$.
- (4) $G = K$, $q = z^{**}$ and $\theta = 0$.
- (5) $H = I = J = \emptyset$.

5. Unified algorithm for ECON and STEM

In the e-Constraint Method (ECON), one of the objectives is selected as a *primary objective* for maximization subject to e_i lower bounds on each of the other objectives. Then the resulting e-constraint sampling program is solved for criterion vector $z^{(1)}$. From $z^{(1)}$ it is hoped that a new primary objective and/or new e_i lower bounds will suggest themselves. With its new configuration, the e-constraint sampling program is solved for $z^{(2)}$. From $z^{(2)}$ it is hoped that an even better configuration of the e-constraint sampling program will come to mind, and so forth. ECON terminates when the DM wishes to cease iterat-

ing. When ECON is the current interactive procedure, the steps (except for Steps 0, 9 and 10 which are permanent) of the unified algorithm would be:

Step 1. $h = 0$.

Step 2. (Routine T-1) Continue.

Step 3. $h = h + 1$.

(Routine E-1) DM selects a primary objective $f_j(x) = z_j$.

(Routine E-2) $H = K - \{j\}$ and DM specifies criterion value lower bounds e_i , for each $i \in H$.

Step 4. One e-constraint customization of the unified sampling program is solved for $z^{(h)}$.

Step 5. (Routine P-1) Present criterion vector $z^{(h)}$ to the DM.

Step 6. (Routine T-1) Continue.

Step 7. If the DM wishes to switch to a different interactive procedure, go to Step 9. Otherwise, (Routine T-1) Continue.

Step 8. (Routine T-2) If the DM wishes to cease iterating, go to Step 10. Otherwise, go to Step 3.

Using its own rules to compute a $\lambda \in \bar{\Lambda}$, STEM solves the minimax e-constraint sampling program to obtain criterion vector $z^{(1)}$. Determining which components of $z^{(1)}$ are to be relaxed and the amounts of each relaxation, e-constraints are introduced and a new $\lambda \in \bar{\Lambda}$ is computed. Using the new λ -vector and e-constraints, the minimax e-constraint sampling program is again solved to produce criterion vector $z^{(2)}$. Determining which components of $z^{(2)}$ are to be relaxed and the amounts of each relaxation, the old set of e-constraints is replaced by a new set and another $\lambda \in \bar{\Lambda}$ is computed, and so forth. STEM terminates on iteration h when the DM is unwilling to relax any of the components of $z^{(h)}$. When STEM is the current interactive procedure, the steps of the unified algorithm would be:

Step 1. $h = 0$.

Step 2. If Routines I-1 and I-2 have not already been performed,

(Routine I-1) Construct a payoff table and form z^* reference criterion vector.

(Routine I-2) $H = I_R = \emptyset$.

If $h > 0$,

(Routine E-3) DM specifies

(i) the set of indices I_R of the components of

$z^{(h)}$ to be relaxed, and

(ii) the maximum amounts of relaxation Δz_i , $i \in I_R$.

(Routine E-4) $H = K$, $e_i = z_i^{(h)} - \Delta z_i$, for each $i \in I_R$, and $e_i = z_i^{(h)}$, for each $i \in (H - I_R)$.

Step 3. $h = h + 1$.

(Routine C-1) Compute $\lambda \in \bar{\Lambda}$ according to the rules in [2, p. 370].

Step 4. Solve one minimax e-constraint customization of the unified sampling program for $z^{(h)}$.

Step 5. (Routine P-1) Present $z^{(h)}$ and z^* to the DM.

Step 6. (Routine T-1) Continue.

Step 7. If the DM wishes to switch to a different interactive procedure, go to Step 9. Otherwise,

(Routine E-3) DM specifies

(i) the set of indices I_R of the components of $z^{(h)}$ to be relaxed, and

(ii) the maximum amounts of relaxation Δz_i , $i \in I_R$.

(Routine E-4) $H = K$, $e_i = z_i^{(h)} - \Delta z_i$, for each $i \in I_R$, and $e_i = z_i^{(h)}$, for each $i \in (H - I_R)$.

Step 8: (Routine T-3) If $I_R = \emptyset$, go to Step 10. Otherwise, go to Step 3.

To assure that the criterion vectors $z^{(h)}$ generated by ECON and STEM are nondominated, we could define each of their sampling programs with a second lexicographical level, where the second lexicographic level is the same as in the lexicographic Tchebycheff sampling program.

6. Unified algorithm for GDF and IGP

The Geoffrion–Dyer–Feinberg Procedure (GDF) begins with the specification by the DM of an initial feasible criterion vector $z^{(0)}$. Then the DM specifies a weighting vector $\lambda \in \bar{\Lambda}$ that is to be reflective of the local marginal tradeoffs at $z^{(0)}$. Using this λ , the weighted-sums sampling program is solved for criterion vector $y^{(1)}$. Then the line through the feasible region in criterion space Z that starts at $z^{(0)}$ and ends at $y^{(1)}$ is divided into segments so as to create *equally spaced* criterion vectors. The most preferred of the equally spaced criterion vectors becomes $z^{(1)}$. Then the DM specifies a new $\lambda \in \bar{\Lambda}$ that is to be reflective of the local marginal tradeoffs at $z^{(1)}$.

Using this λ , the weighted-sums sampling program is solved for $y^{(2)}$. Then the line segment through Z that starts at $z^{(1)}$ and ends at $y^{(2)}$ is divided into segments so as to create equally spaced criterion vectors. The most preferred of these equally spaced criterion vectors becomes $z^{(2)}$, and so forth.

Although several possible stopping conditions have been mentioned for GDF (Steuer [42, p. 369]), the one modelled here is that the DM wishes to cease iterating. When GDF is the current interactive procedure, the steps of the unified algorithm would be:

Step 1. $h = 0$.

Step 2. If $h = 0$,

(Routine I-3) DM specifies an initial feasible criterion vector $z^{(0)}$.

Step 3. $h = h + 1$.

(Routine C-2) DM specifies a weighting vector $\lambda \in \bar{\Lambda}$, or one is calculated from his or her responses (Dyer [11, p. 207]), that is reflective of the local marginal tradeoffs at $z^{(h-1)}$.

Step 4. One weighted-sums customization of the unified sampling program is solved for $y^{(h)}$.

Step 5. (Routine E-5) DM specifies P , the number of equally spaced criterion vector solutions to be presented. (Routine C-3) The line between $z^{(h-1)}$ and $y^{(h)}$ is divided into segments so as to create P equally spaced criterion vectors. (Routine P-1) The P equally spaced criterion vectors are presented to the DM.

Step 6. (Routine E-6) DM selects the most preferred of the P equally spaced criterion vectors and designates it $z^{(h)}$.

Step 7. If the DM wishes to switch to a different interactive procedure, go to Step 9. Otherwise, (Routine T-1) Continue.

Step 8. (Routine T-2) If the DM wishes to cease iterating, go to Step 10. Otherwise, go to Step 3.

In Interactive Goal Programming (IGP) we specify the number of lexicographic levels, target values, and within lexicographic level weights. Then the goal programming sampling program is solved to obtain $z^{(1)}$. In the light of $z^{(1)}$, the DM makes changes to the number of lexicographic levels, target values, and within lexicographic level weights. Then the resulting goal programming sampling problem is solved for $z^{(2)}$, and so forth.

IGP stops when the DM no longer wishes to make further adjustments to the number of lexicographic levels, target values, and within lexicographic level weights. When IGP is the current interactive procedure, the steps of the unified algorithm would be:

Step 1. $h = 0$.

Step 2. (Routine T-1) Continue.

Step 3. $h = h + 1$.

(Routine E-7) DM specifies (a) the number of lexicographic levels L , (b) the target values t_i , $i \in I$, and u_i , $i \in J$, and (c) the within lexicographic level weights $w_{(m)i}^-$, $i \in I$, and $w_{(m)i}^+$, $i \in J$, for all $m = 1, \dots, L$.

Step 4. One goal programming customization of the unified sampling program is solved for $z^{(h)}$, $d^{-(h)}$ and $d^{+(h)}$.

Step 5. (Routine P-1) Present $z^{(h)}$, $d^{-(h)}$ and $d^{+(h)}$ to the DM.

Step 6. (Routine T-1) Continue.

Step 7. If the DM wishes to switch to a different interactive procedure, go to Step 9. Otherwise,

(Routine T-1) Continue.

Step 8. (Routine T-5) If the DM does not wish to make further changes to the number of lexicographic levels, target values, or within lexicographic level weights, go to Step 10. Otherwise, go to Step 3.

7. Unified algorithm for WIERZ, TCH, and SATIS

The Aspiration Criterion Method (WIERZ) begins by asking the DM to specify an aspiration criterion vector $q^{(1)} < z^{**}$. Using the T-vertex λ -vector defined by $q^{(1)}$ and z^{**} , the augmented Tchebycheff sampling program is solved to produce $z^{(1)}$. In the light of $z^{(1)}$, the DM specifies a new aspiration criterion vector $q^{(2)}$. Using the T-vertex λ -vector defined by $q^{(2)}$ and z^{**} , the augmented Tchebycheff sampling program is solved to produce $z^{(2)}$. In the light of $z^{(2)}$, the DM specifies a third aspiration criterion vector $q^{(3)}$, and so forth. WIERZ terminates when the DM wishes to cease iterating.

When WIERZ is the current interactive procedure, the steps of the unified algorithm would be:

Step 1. Let $h = 0$.

Step 2. (Routine I-4) DM (or analyst) specifies ρ . If Routines I-1 and I-5 have not already been performed,

(Routine I-1) Construct a payoff table and form z^* reference criterion vector.

(Routine I-5) Form a z^{**} reference criterion vector.

Step 3. $h = h + 1$.

(Routine E-8) DM specifies an aspiration criterion vector $q^{(h)}$.

(Routine C-4) Compute T-vertex λ -vector defined by $q^{(h)}$ and z^{**} .

Step 4. One augmented Tchebycheff customization of the unified sampling program is solved for $z^{(h)}$.

Step 5. (Routine P-1) Present $z^{(h)}$, $q^{(h)}$, and z^{**} to the DM.

Step 6. (Routine T-1) Continue.

Step 7. If the DM wishes to switch to a different interactive procedure, go to Step 9. Otherwise,

(Routine T-1) Continue.

Step 8. (Routine T-2) If the DM wishes to cease iterating, go to Step 10. Otherwise, go to Step 3.

Letting P be the number of solutions to be presented to the DM at each iteration, the Augmented Version of the Tchebycheff Method (TCH) begins by generating $2P$, where 2 is the *oversampling factor*, well-spaced λ -vectors from $\Lambda^{(1)} = \Lambda$. Then the augmented Tchebycheff sampling program is solved for each of the λ -vectors. From the P most different of the criterion vectors, the DM selects his or her most preferred $z^{(1)}$. At this point, the DM (or analyst) specifies a Λ size factor $\eta^{(2)}$. About the T-vertex λ -vector defined by $z^{(1)}$ and z^{**} , an interval defined subset $\Lambda^{(2)}$ is formed whose size is $\eta^{(2)}$ times the size of Λ . Then $2P$ well-spaced λ -vectors are generated from $\Lambda^{(2)}$ and the augmented Tchebycheff sampling program is solved for each of these λ -vectors.

From the P most different of the resulting criterion vectors, a $z^{(2)}$ is selected. An $\eta^{(3)} < \eta^{(2)}$ size factor is specified. About the T-vertex λ -vector defined by $z^{(2)}$ and z^{**} , a smaller interval defined subset $\Lambda^{(3)}$ is formed whose size is $\eta^{(3)}$ times the size of Λ , and so forth. Although TCH can be calibrated to terminate in a fixed number of iterations (Steuer [42, p. 447]), TCH is mod-

elled here in free-form allowing the DM to terminate whenever he or she wishes to cease iterating. When the augmented version of TCH is the current interactive procedure, the steps of the unified algorithm would be:

- Step 1.* $h = 0$.
Step 2. If Routines I-1 and I-5 have not already been performed,
 (Routine I-1) Construct a payoff table and form z^* reference criterion vector.
 (Routine I-5) Form a z^{**} reference criterion vector.
 (Routine E-5) DM specifies P , the number of new criterion vectors to be presented.
 (Routine I-4) DM (or analyst) specifies ρ .
 (Routine I-6) DM (or analyst) specifies oversampling factor ν .
 If $h = 0$,
 (Routine I-7) Let $(I_i^{(1)}, \mu_i^{(1)}) = (0, 1)$ for all $i \in K$.
Step 3. $h = h + 1$.
 If $h > 1$,
 (Routine E-9) DM specifies Λ size factor $\eta^{(h)}$.
 (Routine C-4) Compute T-vertex λ -vector defined by $z^{(h-1)}$ and z^{**} .
 (Routine C-5) Compute the $(I_i^{(h)}, \mu_i^{(h)})$ associated with the T-vertex λ -vector and $\eta^{(h)}$ for all $i \in K$.
 (Routine C-6) Form $A^{(h)} = \{\lambda \in R^k \mid \lambda_i \in (I_i^{(h)}, \mu_i^{(h)}), \sum_{i \in K} \lambda_i = 1\}$.
 (Routine C-7) Randomly generate $50k$ uniformly distributed λ -vectors over $A^{(h)}$.
 (Routine C-8) Filter the $50k$ λ -vectors to obtain νP well-spaced representatives.
Step 4. Solve νP augmented Tchebycheff customizations of the unified sampling program, one for each of the νP λ -vectors.
Step 5. (Routine C-8) Filter the νP resulting criterion vectors to obtain the P most different.
 (Routine P-1) Present the P most dispersed criterion vectors along with, if $h > 1$, $z^{(h-1)}$ to the DM.
Step 6. (Routine E-6) DM selects the most preferred of the presented criterion vectors and designates it $z^{(h)}$.
Step 7. If the DM wishes to switch to a different interactive procedure, go to Step 9. Otherwise,
 (Routine T-1) Continue.

Step 8. (Routine T-2) If the DM wishes to cease iterating, go to Step 10. Otherwise, go to Step 3.

The steps of the unified algorithm for the Lexicographic Version of TCH are the same as for the Augmented Version except that Routine I-4 is deleted from Step 2 and νP lexicographic Tchebycheff customizations are solved in Step 4 instead of νP augmented Tchebycheff customizations.

The Satisficing Tradeoff Method (SATIS) begins with the specification of a z^{**} reference criterion vector and an initial aspiration criterion vector $q^{(1)}$. Then the augmented Tchebycheff sampling program is solved using the T-vertex λ -vector defined by $q^{(1)}$ and z^{**} to produce $z^{(1)}$. The DM then specifies the components of $z^{(1)}$ that are to be increased, the amounts of each increase, and which components are to be relaxed. Using dual variable information available at $z^{(1)}$, the amounts of relaxation are determined so as to form a second aspiration criterion vector $q^{(2)}$. Using the T-vertex λ -vector defined by $q^{(2)}$ and z^{**} , the augmented Tchebycheff sampling program is solved to produce $z^{(2)}$.

The DM then specifies which components of $z^{(2)}$ are to be increased, the amounts of each increase, and which components are to be relaxed. Using dual variable information available at $z^{(2)}$, the amounts of relaxation are determined so as to form $q^{(3)}$. Using the T-vertex λ -vector defined by $q^{(3)}$ and z^{**} , the augmented Tchebycheff sampling program produces $z^{(3)}$, and so forth.

SATIS terminates when the DM wishes to cease iterating. When SATIS is the current interactive procedure, the steps of the unified algorithm would be:

- Step 1.* $h = 0$.
Step 2. If Routines I-1 and I-5 have not already been performed,
 (Routine I-1) Construct a payoff table and form z^* reference criterion vector.
 (Routine I-5) Form a z^{**} reference criterion vector.
 (Routine I-4) DM (or analyst) specifies ρ .
 If $h > 0$,
 (Routine C-4) Compute T-vertex λ -vector defined by $z^{(h)}$ and z^{**} .

One augmented Tchebycheff customization of the unified sampling program is solved to regenerate $z^{(h)}$.

(Routine T-4) Go to Step 3.

(Routine E-8) DM specifies an aspiration criterion vector $q^{(1)}$.

Step 3. $h = h + 1$.

If $h > 1$,

(Routine E-10) DM specifies (a) index set I_1 of components of $z^{(h-1)}$ to be improved, (b) index set I_R of components to be relaxed, and (c) index set I_A of components to be held at their current values. For an objective function index to be included in I_R , the local marginal tradeoff value must not be zero.

(Routine E-11) DM specifies $q_i^{(h)}$ for each $i \in I_1$.

(Routine C-9) Let $q_i^{(h)} = z_i^{(h-1)}$ for each $i \in I_A$. Using dual variable information, calculate $q_i^{(h)}$ for each $i \in I_R$.

(Routine E-12) DM may further adjust components of $q^{(h)}$ if desired.

(Routine C-4) Compute T-vertex λ -vector defined by $q^{(h)}$ and z^{**} .

Step 4. One augmented Tchebycheff customization of the unified sampling program is solved for $z^{(h)}$.

Step 5. (Routine P-1) Present $z^{(h)}$, $q^{(h)}$, z^{**} , and local marginal tradeoff values to the DM.

Step 6. (Routine T-1) Continue.

Step 7. If the DM wishes to switch to a different interactive procedure, go to Step 9. Otherwise,

(Routine T-1) Continue.

Step 8. (Routine T-2) If the DM wishes to cease iterating, go to Step 10. Otherwise, go to Step 3.

8. Unified algorithm for VIA and RACE

With reference to an initial feasible criterion vector $z^{(0)}$, the Visual Interactive Approach (VIA) begins by asking the DM to specify (a) estimated criterion value range widths r_i for each of the objectives over N^w and (b) an aspiration criterion vector $q^{(1)}$. Then we let $\lambda_i = 1/r_i$ for each $i \in K$. To project onto N^w the unbounded line segment emanating from $z^{(0)}$ through $q^{(1)}$, we solve, when

$h = 1$, the *Tchebycheff parametric sampling program*

min $\{\alpha\}$

s.t. $\alpha \geq \lambda_i(z_i^{(h-1)} + \theta d_i^{(h)} - z_i)$, $i \in K$,

$f_i(x) = z_i$, $i \in K$,

$x \in S$,

$\alpha \in R$ unrestricted, $z \in R^k$ unrestricted,

as θ goes from 0 to ∞ where the direction vector $d^{(h)} = q^{(h)} - z^{(h-1)}$. The Tchebycheff parametric sampling program for iteration h is formed from the unified sampling program by letting

(1) $L = 1$.

(2) $\sigma_{(1)} = 1$ and $\rho_{(1)} = \tau_{(1)} = 0$.

(3) $G = K$, $q = z^{(h-1)}$ and $d = d^{(h)}$.

(4) $H = I = J = \emptyset$.

(5) $\alpha \in R$ unrestricted.

Variable $\alpha \in R$ is made unrestricted in case any of the points on the unbounded line segment defined by $z^{(h-1)}$ and $d^{(h)}$ are feasible.

Using computer graphics, the *trajectories* of criterion values describing the projection of the line segment onto N^w are displayed. After viewing the trajectories, the DM specifies his or her most preferred criterion vector along the projected line segment to yield $z^{(1)}$. At this point the DM specifies a new aspiration criterion vector $q^{(2)}$. Then, to project onto N^w the unbounded line segment emanating from $z^{(1)}$ through $q^{(2)}$, we solve the Tchebycheff parametric sampling program for $h = 2$. After viewing the resulting trajectories, the DM selects a $z^{(2)}$ along the projected line segment. At this point the DM specifies a third aspiration criterion vector $q^{(3)}$, and so forth. Although Korhonen and Laakso [21] outline an optimality check involving generators of the cone of feasible nondominated directions at the point of termination, the mathematics of this check have not yet been completed (Halme and Korhonen [19]). Consequently, VIA is modelled here to terminate when the DM wishes to cease iterating. When VIA is the current interactive procedure, the steps of the unified algorithm would be:

Step 1. Let $h = 0$.

Step 2. If Routine I-1 has not already been performed,

(Routine I-8) DM estimates criterion value range widths r_i for each of the objectives over N^w .

If Routine I-1 has been performed,

(Routine I-9) For each $i \in K$, let r_i equal the difference between z_i^* and the minimum value in the i -th column of the payoff table.

(Routine C-10) Compute λ -vector such that $\lambda_i = 1/r_i$, for each $i \in K$.

If $h = 0$,

(Routine I-3) DM specifies an initial feasible criterion vector $z^{(0)}$.

Step 3. $h = h + 1$.

(Routine E-8) DM specifies aspiration criterion vector $q^{(h)}$.

(Routine C-11) Let $d^{(h)} = q^{(h)} - z^{(h-1)}$.

Step 4. One Tchebycheff parametric customization of the unified sampling program is solved to project the unbounded line segment emanating from $z^{(h-1)}$ in the direction $d^{(h)}$ onto N^w .

Step 5. (Routine P-2) The criterion value trajectory of each objective describing the projection is presented to the DM.

Step 6. (Routine E-13) DM selects the most preferred criterion vector described by the trajectories and designates it $z^{(h)}$.

Step 7. If the DM wishes to switch to a different interactive procedure, go to Step 9. Otherwise,

(Routine T-1) Continue.

Step 8. (Routine T-2) If the DM wishes to cease iterating, go to Step 10. Otherwise, go to Step 3.

Pareto Race (RACE) enables the DM to explore N^w as if one were driving a vehicle over the surface of the moon. Microcomputer keys are used to control the 'drive' (see [23,24]). RACE begins by asking the DM to specify

(a) an aspiration criterion vector designated $z^{(0)}$ and

(b) estimated criterion value range widths r_i for each of the objectives over N^w .

Then we let $\lambda_i = 1/r_i$ for each $i \in K$ and form, according to special rules, an initial direction vector $d^{(1)}$. In order to project $z^{(0)}$ onto N^w to obtain $z^{(1)}$, we solve, with $h = 1$ and RHS step size $s^{(1)} = 0$, the Tchebycheff step size sampling

program

$$\begin{aligned} \min \quad & \{\alpha\} \\ \text{s.t.} \quad & \alpha \geq \lambda_i (z_i^{(h-1)} + s^{(h)} d_i^{(h)} - z_i) \quad i \in K, \\ & z_i \geq e_i, \quad i \in I_R, \\ & f_i(x) = z_i, \quad i \in K, \\ & x \in S, \\ & \alpha \in R \text{ unrestricted, } z \in R^k \text{ unrestricted.} \end{aligned}$$

The Tchebycheff step size sampling program for iteration h is formed from the unified sampling program by letting

- (1) $L = 1$.
- (2) $\sigma_{(1)} = 1$ and $\rho_{(1)} = \tau_{(1)} = 0$.
- (3) $G = K$, $\theta = s^{(h)}$ and $d = d^{(h)}$.
- (4) $H = I_R$.
- (5) $I = J = \emptyset$.
- (6) $\alpha \in R$ unrestricted.

Variable α is unrestricted for the same reason as in VIA. After considering $z^{(1)}$, microcomputer keys are used to determine $s^{(2)}$, form a new direction vector $d^{(2)}$, and optionally impose or remove lower bounds on the criterion values. Then the Tchebycheff step size sampling program for $h = 2$ is solved to obtain $z^{(2)}$. After considering $z^{(2)}$, microcomputer keys are used to determine a new step size $s^{(3)}$, a third direction vector $d^{(3)}$, and alter lower bounds on the criterion values. Then the Tchebycheff step size sampling program for $h = 3$ is solved to obtain $z^{(3)}$, and so forth. When RACE is the current interactive procedure, the steps of the unified algorithm would be:

Step 1. $h = 0$.

Step 2. If Routine I-1 has not already been performed,

(Routine I-8) DM estimates criterion value range widths r_i for each of the objectives over N^w .

If Routine I-1 has been performed,

(Routine I-9) For each $i \in K$, let r_i equal the difference between z_i^* and the minimum value in the i -th column of the payoff table.

(Routine C-10) Compute λ -vector such that $\lambda_i = 1/r_i$, for each $i \in K$.

If $h = 0$,

(Routine E-8) DM specifies an aspiration criterion vector designated $z^{(0)}$.

(Routine I-10) Let step size $s^{(1)} = 0$, the initial

direction vector $d^{(1)}$ be set according to the rules in [23, p. 619], and $I_R = \emptyset$.

Step 3. $h = h + 1$.

If $h = 2$,

(Routine I-11) $s^{(2)} = b$ where b is the default base speed [23, p. 619].

If $h > 1$,

(Routine E-14) Either the DM presses micro-computer keys to specify (a) an updated $s^{(h)}$, (b) an updated $d^{(h)}$, (c) an updated I_R and associated criterion value lower bounds e_i ; or we let (i) $s^{(h)} = s^{(h-1)}$, (ii) $d^{(h)} = d^{(h-1)}$, (iii) I_R and the associated e_i values remain as they are.

(Routine T-6) Pause until DM hits space bar.

Step 4. One Tchebycheff step size customization of the unified sampling program is solved for $z^{(h)}$.

Step 5. (Routine P-1) Present criterion vector $z^{(h)}$ to the DM.

Step 6. (Routine T-1) Continue.

Step 7. If the DM wishes to switch to a different interactive procedure, go to Step 9. Otherwise,

(Routine T-1) Continue.

Step 8. (Routine T-2) If the DM wishes to cease iterating, go to Step 10. Otherwise, go to Step 3.

9. Concluding remarks

In this paper we have taken a nontraditional look at interactive multiple objective programming. Instead of classifying and categorizing the different interactive procedures according to their search strategies or the way they elicit information from the DM, we have looked at interactive multiple objective programming from the other side, the implementation/installation side. By employing the unified sampling program along with a relatively modest inventory of component subroutines, many of which would not involve more than a few lines of code, we have shown how seemingly different interactive procedures are often no more than simple variants of the unified algorithm.

The advantages of viewing interactive multiple objective programming from a unified perspective are several. One is pedagogical. It might be easier to learn interactive multiple objective programming

by studying the unified algorithm first, and then seeing how different interactive procedures fall out as special cases second. Another advantage of the unified approach is that it is consistent with the emerging practice of modern interactive multiple objective programming in which DMs wish to have ranges of interactive procedures at their disposal. In addition, the unified design in this paper supports procedure-switching which is likely to grow in popularity. Furthermore, by providing a blueprint for algorithm design, the paper ideally will encourage the authors of future interactive multiple objective programming software to include up to five or ten procedures in their products for robust deployment, thus helping to bring to an end the fragmented software picture that exists in the field today.

References

- [1] Aksoy, Y., "Interactive multiple objective decision making: A bibliography (1965–1988)", *Management Research News* 13/2 (1990) 1–8.
- [2] Benayoun, R., de Montgolfier, J., Tergny, J., and Larichev, O., "Linear programming with multiple objective functions: Step method (STEM)", *Mathematical Programming* 1/3 (1971) 366–375.
- [3] Brandner, P., and Böhm, B., "Interactive optimization: Manual for the IAO-program package", Research Report No. 95, Institute for Economics, Operations Research and Systems Theory, University of Technology, Vienna, Austria, 1989.
- [4] Brockhoff, K., "Experimental test of MCDM algorithms in a modular approach", *European Journal of Operational Research* 22/2 (1985) 159–166.
- [5] Buchanan, J.T., and Daellenbach, H.G., "A comparative evaluation of interactive solution methods for multiple objective decision models", *European Journal of Operational Research* 29/3 (1987) 353–359.
- [6] Chankong, V., and Haimes, Y.Y., "The interactive surrogate worth trade-off (ISWT) method for multiobjective decision-making", in: S. Zionts (ed.), *Multiple Criteria Problem Solving, Lecture Notes in Economics and Mathematical Systems*, Vol. 155, (1978) 42–67.
- [7] Chankong, V., and Haimes, Y.Y., *Multiobjective Decision Making: Theory and Methodology*, North-Holland, New York, 1983.
- [8] Climaco, J.C.N., and Antunes, C.H., "TRIMAP – An interactive tricriteria linear programming package", *Foundations of Control Engineering* 12/3 (1987) 101–120.
- [9] Dror, M., and Gass, S.I., "Interactive scheme for a MOLP problem given two partial orders: One on variables and one on objectives", *Applied Mathematics and Computation* 24 (1987) 195–209.
- [10] Dyer, J.S., "Interactive goal programming", *Management Science* 19/1 (1972) 62–70.

- [11] Dyer, J.S., "An empirical investigation of a man-machine interactive approach to the solution of the multiple criteria problem", in: J.L. Cochrane and M. Zeleny (eds.), *Multiple Criteria Decision Making*, University of South Carolina Press, Columbia, SC, 1973, 202–216.
- [12] Evren, R., "Interactive compromise programming", *Journal of the Operational Research Society* 38/2 (1987) 163–172.
- [13] Fichet, J., "GPSTEM: An interactive multi-objective optimization method", in: *Progress in Operations Research*, Vol. 1, North-Holland, Amsterdam, 1976, 317–332.
- [14] Franz, L.S., and Lee, S.M., "A goal programming based interactive decision support system", in: J.N. Morse, (ed.), *Organizations: Multiple Agents with Multiple Criteria, Lecture Notes in Economics and Mathematical Systems*, Vol. 190, (1981) 110–115.
- [15] Gabbani, D., and Magazine, M., "An interactive heuristic approach for multi-objective integer-programming problems", *Journal of the Operational Research Society* 37 (1986) 285–291.
- [16] Galas, Z., Nykowski, I., and Zolkiewski, Z. *Programowanie Wielokryterialne*, Państwowe Wydawnictwo Ekonomiczne, 1987.
- [17] Geoffrion, A.M., Dyer, J.S., and Feinberg, A., "An interactive approach for multicriterion optimization, with an application to the operation of an academic department", *Management Science* 19/4 (1972) 357–368.
- [18] Grauer, M., "A dynamic interactive decision analysis and support system (DIDASS): User's guide", WP-83-60, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1983.
- [19] Halme, M., and Korhonen, P.J., "Nondominated trade-offs and termination in interactive multiple objective linear programming", in: A.G. Lockett and G. Islei (eds.), *Improving Decision Making in Organizations, Lecture Notes in Economics and Mathematical Systems*, Vol. 335, (1989) 410–423.
- [20] Hemming, T., "A new method for interactive multi-objective optimization: A boundary point ranking method", in: H. Thiriez and S. Zionts (eds.), *Multiple Criteria Decision Making, Lecture Notes in Economics and Mathematical Systems*, Vol. 130, (1976) 333–340.
- [21] Korhonen, P.J., and Laakso, J., "A visual interactive method for solving the multiple criteria problem", *European Journal of Operational Research* 24/2 (1986) 277–287.
- [22] Korhonen, P.J., and Laakso, J., "Solving generalized goal programming problems using a visual interactive approach", *European Journal of Operational Research* 26/3 (1986) 355–363.
- [23] Korhonen, P.J., and Wallenius, J., "A Pareto race", *Naval Research Logistics* 35/6 (1988) 615–623.
- [24] Korhonen, P.J., and Wallenius, J., "A multiple objective linear programming decision support system", *Decision Support Systems* 6/3 (1990) 243–251.
- [25] Lazimy, R., "Interactive relaxation method for a broad class of integer and continuous nonlinear multiple criteria problems", *Journal of Mathematical Analysis and Applications* 116/2 (1986) 553–573.
- [26] Lewandowski, A., and Grauer, M., "The reference point optimization approach – Methods of efficient implementation", in: M. Grauer, A. Lewandowski and A.P. Wierzbicki (eds.), *Multiobjective and Stochastic Optimization*, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1982, 353–376.
- [27] Lewandowski, A., Kreglewski, T., Rogowski, T., and Wierzbicki, A.P., "Decision support systems of DIDAS family (Dynamic Interactive Decision Analysis and Support)", *Archiwum Automatyki i Telemekhaniki* 32/4 (1987) 221–246.
- [28] Lewandowski, A., and Wierzbicki, A.P., "Decision support systems using reference point optimization", in: A. Lewandowski and A.P. Wierzbicki (eds.), *Aspiration Based Decision Support Systems, Lecture Notes in Economics and Mathematical Systems*, Vol. 331, (1989) 3–20.
- [29] Liou, F.H., "A routine for generating grid point defined weighting vectors", Masters Thesis, Department of Management Science and Information Technology, University of Georgia, Athens, GA, 1984.
- [30] Loganathan, G.V. and Sherali, H.D., "A convergent interactive cutting-plane algorithm for multiobjective optimization", *Operations Research* 35/3 (1987) 365–377.
- [31] Malakooti, B., "An exact interactive method for exploring the efficient facets of multiple objective linear programming problems with quasi-concave utility functions", *IEEE Transactions on Systems, Man, and Cybernetics* 18/5 (1988) 787–801.
- [32] Marcotte, O., and Soland, R.M., "An interactive branch-and-bound algorithm for multiple criteria optimization", *Management Science* 32/1 (1986) 61–75.
- [33] Michalowski, W., and Szapiro, T., "A procedure for worst outcomes displacement in multiple criteria decision making", *Computers and Operations Research* 16/3 (1989) 195–206.
- [34] Nakayama, H., "Interactive multi-objective programming and its applications", in: A. Lewandowski, Serafini, P. and Speranza, M.G. (eds.), *Methodology, Implementation and Applications of Decision Support Systems*, Springer-Verlag, Vienna, 1991, forthcoming.
- [35] Nakayama, H., and Sawaragi, Y., "Satisficing trade-off method for multiobjective programming", *Lecture Notes in Economics and Mathematical Systems* 229 (1984) 114–122.
- [36] Pomerol, J.C., and Trabelsi, T., "An adaptation of PRIAM to multiobjective linear programming", *European Journal of Operational Research* 31/3 (1987) 335–341.
- [37] Reeves, G.R., and Franz, L.S., "A simplified interactive multiple objective linear programming procedure", *Computers & Operations Research* 12/6 (1985) 589–601.
- [38] Sakawa, M., and Yano, H., "An interactive fuzzy satisficing method for generalized multiobjective programming problems with fuzzy parameters", *Fuzzy Sets and Systems* 35/2 (1990) 125–142.
- [39] Shin, W.A., and Ravidran, A., "Interactive multiple objective optimization: Survey I – Continuous case", *Computers & Operations Research* 18/1 (1991) 97–114.
- [40] Slowinski, R., and Warczynski, J., "Application of the ellipsoid method in an interactive procedure for multicriteria linear programming", *Zeitschrift für Operations Research* 28/3 (1984) 89–100.

- [41] Spronk, J., *Interactive Multiple Goal Programming*, Martinus Nijhoff, Boston, 1981.
- [42] Steuer, R.E., *Multiple Criteria Optimization: Theory, Computation, and Application*, Wiley, New York, 1986.
- [43] Steuer, R.E., "The Tchebycheff procedure of interactive multiple objective programming", in: B. Karpak, and S. Zionts (eds.), *Multiple Criteria Decision Making and Risk Analysis Using Microcomputers*, NATO ASI Series F, Vol. 56, Springer-Verlag, Berlin, 1989, 235–249.
- [44] Steuer, R.E., "Manual for the ADBASE multiple objective linear programming package", Department of Management Science and Information Technology, University of Georgia, Athens, GA, 1992.
- [45] Steuer, R.E., and Choo, E.-U. "An interactive weighted Tchebycheff procedure for multiple objective programming", *Mathematical Programming* 26/1 (1983) 326–344.
- [46] Steuer, R.E., and Gardiner, L.R., "Toward a unified theory of multiple objective programming", ORSA/TIMS Joint National Meeting, October, 1986.
- [47] Stewart, T.J., "An interactive multiple objective linear programming method based on piecewise-linear additive value functions", *IEEE Transactions on Systems, Man, and Cybernetics* 17/5 (1987) 799–805.
- [48] Teghem, Jr., J., Dufrane, D., Thauvoye, M., and Kunsch, P., "STRANGE: An interactive method for multi-objective linear programming under uncertainty", *European Journal of Operational Research* 26/1 (1986) 65–82.
- [49] Vincke, P., *L'Aide Multicritère à la Décision*, Éditions de l'Université de Bruxelles, Brussels, 1988.
- [50] White, D.J., "Multi-objective interactive programming", *Journal of the Operational Research Society* 31/6 (1980) 517–523.
- [51] White, D.J., "The maximal dispersion problem and the 'First point outside the neighborhood' heuristic", *Computers & Operations Research* 18/1 (1991) 43–50.
- [52] Wierzbicki, A.P., "Basic properties of scalarizing functionals for multiobjective optimization", *Mathematische Operationsforschung und Statistik – Series Optimization* 8/1 (1977) 55–60.
- [53] Wierzbicki, A.P., "The use of reference objectives in multiobjective optimization", in: G. Fandel and T. Gal (eds.), *Multiple Criteria Decision Making: Theory and Application, Lecture Notes in Economics and Mathematical Systems*, Vol. 177, 1980, 468–486.
- [54] Wierzbicki, A.P., "On the completeness and constructiveness of parametric characterizations to vector optimization problems", *OR Spektrum* 8/2 (1986) 73–87.
- [55] Winkels, H.-M., and Meika, M., "An integration of efficiency projections into the Geoffrion approach for multi-objective linear programming", *European Journal of Operational Research* 16/1 (1984) 113–127.
- [56] Zionts, S., "Multiple criteria decision making: An updated overview and several approaches", in: B. Karpak and S. Zionts (eds.), *Multiple Criteria Decision Making and Risk Analysis Using Microcomputers*, NATO ASI Series F, Vol. 56, Springer-Verlag, Berlin, 1989, 7–60.
- [57] Zionts, S., and Wallenius, J., "An interactive programming method for solving the multiple criteria problem", *Management Science* 22/6 (1976) 652–663.
- [58] Zionts, S., and Wallenius, J., "An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions", *Management Science* 29/5 (1983) 519–529.