Approximating Nondominated Sets in Continuous Multiobjective Optimization Problems

Jacinto Martín,[1] **Concha Bielza,**[2] **David Ríos Insua**[3]

[1] *Mathematics Departament, Univ. of Extremadura, Carretera de Trujillo, 10071 Cáceres, Spain*

[2] *Artificial Intelligence Dpt., Madrid Technical Univ., Campus de Montegancedo, Boadilla del Monte, 28660 Madrid, Spain*

[3] *Statistics and Decision Sciences Group, Rey Juan Carlos University, Tulipán s/n, Móstoles, 28933 Madrid, Spain*

Received 1 August 2002; revised 14 March 2004; accepted 31 March 2005
DOI 10.1002/nav.20090
Published online 9 May 2005 in Wiley InterScience (www.interscience.wiley.com).

Abstract: Many important problems in Operations Research and Statistics require the computation of nondominated (or Pareto or efficient) sets. This task may be currently undertaken efficiently for discrete sets of alternatives or for continuous sets under special and fairly tight structural conditions. Under more general continuous settings, parametric characterisations of the nondominated set, for example through convex combinations of the objective functions or ε-constrained problems, or discretizations-based approaches, pose several problems. In this paper, the lack of a general approach to approximate the nondominated set in continuous multiobjective problems is addressed. Our simulation-based procedure only requires to sample from the set of alternatives and check whether an alternative dominates another. Stopping rules, efficient sampling schemes, and procedures to check for dominance are proposed. A continuous approximation to the nondominated set is obtained by fitting a surface through the points of a discrete approximation, using a local (robust) regression method. Other actions like clustering and projecting points onto the frontier are required in nonconvex feasible regions and nonconnected Pareto sets. In a sense, our method may be seen as an evolutionary algorithm with a variable population size. © 2005 Wiley Periodicals, Inc. Naval Research Logistics 52: 469–480, 2005.

Keywords: multiple objectives; efficient solutions; stochastic optimization; evolutionary algorithms; nonlinear regression

1. INTRODUCTION

Many problems in Operations Research and Statistics lead to comparisons of alternatives in terms of a class of objective functions, typical reasons being the existence of (1) multiple conflicting objectives or (2) imprecision in the decision maker's beliefs and/or preferences. Important examples include: *multiobjective programming problems* where alternatives are ranked, in a Pareto sense, with respect to several objective functions (see, e.g., [38, 35, 36, 30, 16]); *robust Bayesian analysis problems,* where alternatives are ranked according to a class of posterior expected utilities, with respect to a class of utility functions and a class of prior distributions (see [27] and references therein); *stochastic dominance problems,* where options are ranked according to their expected utilities with respect to a class of utility functions (see, e.g., [17]); *experimental design problems,* where the utility function is difficult to assess and designs are compared with respect to several utility functions (see, e.g., [7]). Our main concern here is with problem (1). In that context, we say that an alternative is better than another one (or dominates it) if it is better for each of the objective functions. The natural solution concept is that of nondominated alternative and our computational aim should be the determination of the nondominated (or Pareto, or efficient) set.

Solutions for this problem are well known under fairly tight structural conditions. Two examples are the computation of the Pareto frontier via the multiobjective simplex method in multiobjective linear programming (see, e.g., [38]) or the nondominated set in discrete decision analytic problems (see, e.g., [25]). Under more general settings, we may appeal to a characterization of the Pareto set, for example, through convex combinations of the objective

Correspondence to: C. Bielza (mcbielza@fi.upm.es); J. Martín (jrmartin@unex.es); D. Ríos Insua (david.rios@urjc.es) functions or ε-constrained problems (see, e.g., [34]). Based on such parametric characterizations, we may undertake a systematic or random exploration of the parametric space, and suggest the associated set of solutions as an approximation to the efficient set. This approach poses several problems, however. For example, for the case of optimisation of convex combinations of the objective functions (see [35, 37]), if there are no preconditions, we may not be able to generate enough solutions to obtain a good approximation of the nondominated set. Moreover, unless appropriate convexity conditions for the objective functions and the set of alternatives hold, current methods cannot solve the corresponding optimisation problem. Therefore, a good idea would be to convert the problem into a form where relevant algorithms do exist which may benefit from the problem structure. In that sense, White [37] proposes convex combinations of finite powers of the objective functions to generate all efficient solutions. This method can be considered as a means of generating solutions additional to those generated from convex combinations of the objective functions. Considerable computational difficulties arise when some of the objective functions are convex and others are concave and a transformation to retain concavity is often needed, which is far from easy. To sum up, there is not a sufficiently general approach to the problem, especially in continuous settings.

A way to tackle the general problem would be through discretization (see [2]). Several recent papers have extended the results obtained with genetic algorithms to multiobjective problems (see, e.g., [13, 8, 5]). Coello and Christiansen [9] place the origins in the work by Rosenberg [28], while Hanne [15] provides convergence results for these methods. However, their convergence depends heavily on the initial population generated and there may even exist unreachable points. Sarker, Liang, and Newton [29] avoid this problem by means of modifications of the population size. Moreover, most papers focus on obtaining some nondominated points, rather than an approximation to the efficient set. Benson and Sayin [3] propose a method to represent this set generating a finite set of solutions. They also provide an excellent bibliography on different solution methods. Their proposal works on the image set instead of the feasible set, with the advantage of reducing the dimension. Other approximation methods to the efficient set may be found in [14, 21]. Teghem, Tuyttens, and Ulungu [31] use simulated annealing for discrete multiobjective problems.

In this paper, we shall provide a general approach to approximate the nondominated set in continuous problems. We only require procedures to sample from the set of alternatives and check whether an alternative dominates another. Then, we may define a discrete approximation of the nondominated set by taking a sample and finding the nondominated subsample. We also provide an efficient set representation. In Section 2 we show that, under appropriate conditions, we truly approximate the nondominated set. Several examples in Section 3 allow us to point out potential shortcomings of our basic approach, which we address in Section 4. Specifically, we study continuous approximations, stopping rules, efficient sampling schemes and how we may deal with the problem of checking dominance. Finally, through an example, we show how our approach may be seen as an evolutionary algorithm with a variable population size. We end up with some conclusions.

2. PROBLEM DESCRIPTION AND BASIC STRATEGY

Suppose we have to choose among a set $\mathscr{A} \subset \mathbf{R}^m$ of alternatives according to a preference relation $\preceq$ (a is at most as preferred as b)

$$a \preceq b \Leftrightarrow (\Psi(a, w) \leq \Psi(b, w),\ \forall\, w \in S).$$

$\Psi(\cdot, w)$ may be interpreted as an objective function to be maximized.

Recall, that the standard multiobjective programming problem with k objective functions v_i,

$$\begin{array}{ll} \max & (v_1(a), \ldots, v_k(a)) \\ \text{s.t.} & a \in \mathscr{A} \end{array}$$

can be described as

$$a \preceq b \Leftrightarrow (v_i(a) \leq v_i(b),\ i = 1, \ldots, k). \tag{1}$$

We could define $\Psi(a, w) = v_w(a)$, $w \in S = \{1, 2, \ldots, k\}$. Examples in Section 3 illustrate this formulation.

The natural solution concept in our problem is the set $N(\mathscr{A})$ of nondominated alternatives in $\mathscr{A}$:

DEFINITION 1: $a \in \mathscr{A}$ is nondominated if there is no $b \in \mathscr{A}$ such that $a \prec b$ (that is, $a \preceq b$ and $b \preceq a$).

Equivalently, we may say that a is nondominated if for any $b \in \mathscr{A}$, $\Psi(a, w) \leq \Psi(b, w)$, $\forall w \in S \Rightarrow \Psi(a, w) = \Psi(b, w)$, $\forall w \in S$. We aim at computing the set $N(\mathscr{A})$.

We shall assume here that S is finite, $\mathscr{A}$ is a nonempty, compact set without isolated points, and functions $\Psi(a, w)$ are continuous with respect to a in a topology appropriate to ensure the existence of the nondominated set. For reasons outlined in the Introduction, in general cases we may not appeal to standard methods based on auxiliary problems with convex combinations of objectives and/or ε-constraints.

Our basic strategy may be seen as a *pure random search* method extended to a case with many objectives. The basic idea is to sample randomly the set of alternatives and apply a pairwise comparison procedure, as in, for example, [26]. We shall make two assumptions:

a. We may sample from $\mathscr{A}$ in such a way that all points in $\mathscr{A}$ are in the support of the underlying sampling distribution. We could use the uniform distribution on $\mathscr{A}$, which can be sampled with the rejection method (see, e.g., [23]), though this is not always efficient. Occasionally, we may have additional information which may be reflected in the distribution used.

b. Given two alternatives a, b, we have a procedure to determine whether $a \preceq b$, or vice versa. This would need discussions for specific cases, but has been the object of extensive work (see, e.g., [24]). The general approach would be to solve problem

$$\min_{w \in S} (\Psi(b, w) - \Psi(a, w)) = \Delta_{ba}. \tag{2}$$

Then,

(a) If $\Delta_{ba} > 0$, $a \prec b$.
(b) If $\Delta_{ba} = 0$,
  i. If $\Delta_{ab} < 0$, $a \prec b$
  ii. If $\Delta_{ab} = 0$, $b \sim a$
(c) If $\Delta_{ba} < 0$, if $\Delta_{ab} \geq 0$, $b \prec a$.
(d) If $\Delta_{ba} < 0$, if $\Delta_{ab} < 0$, $b \not\prec a$ and $a \not\prec b$.

and we may discard alternatives correspondingly. For example, in case 2(a) we would discard alternative a, as it is dominated by b. Note though that problems (2) may be difficult to deal with if, for example, we lack an explicit formula for $\Psi(\,\cdot\,, w)$ or is a high dimension integral, for example, in stochastic optimization problems. In these cases, we can use simulation methods with criteria provided in [20].

The basic approach proposed is as follows:

```
1. Sample $A_n = \{a_1, \ldots, a_n\} \subset \mathscr{A}$.
  2. Choose $w \in S$ and relabel the $a$'s as $(e_1, \ldots, e_n)$ so that $\Psi(e_{i+1}, w) \leq \Psi(e_i, w)$, $\forall i$.
  3. Let $d(i) = 0$, $\forall i$.
  4. For $i = 1$ to $n - 1$
        If $d(i) = 0$
           For $j = i + 1$ to $n$
              If $d(j) = 0$
                 If $e_j \preceq e_i$ then $d(j) = 1$
                             Else, if $(\Psi(e_i, w) = \Psi(e_j, w)$ and $e_i \preceq e_j)$
                                   then $d(i) = 1$ and next $i$
  5. Set $N_n = \{e_i : d(i) = 0\}$.
```

Clearly, we have that N_n is the nondominated set of the discrete approximation A_n, that is, $N_n = N(A_n)$. We can prove that the procedure converges to the true nondominated set N as n increases, under our assumption **a**, i.e., N_n is a discrete approximation to N.

LEMMA 1: Let $\mathscr{A}$ be a nonempty, compact set without isolated points. $\forall a \in \mathscr{A}$, a.s. there is a subsequence (a_{n_j}), obtained with our basic approach, convergent to a.

PROOF: Let p denote the underlying sampling distribution. Fix $a \in \mathscr{A}$ and suppose there is not such sequence. Then, as all the points in $\mathscr{A}$ are in the support of p, there are n_0 and a neighborhood $B(a)$ of a, with $0 < p(B(a)) < 1$ such that $\forall n > n_0$, $a_n \notin B(a)$. But $p(a_n \notin B(a), \forall n > n_0) = \lim_n p(a_i \notin B(a), i = n_0, \ldots, n) = \lim_n \prod_{i=n_0}^{n} p(a_i \notin B(a)) = \lim_n (1 - p(B(a)))^{n-n_0+1} = 0$. □

In particular, this holds $\forall a \in N(\mathscr{A})$.

PROPOSITION 1: Under the conditions of Lemma 1, $\forall a \in N(\mathscr{A})$, a.s. there is a convergent subsequence (a_{n_j}), with $a_{n_j} \in N_{n_j}$ such that $a_{n_j} \to b$ and $b \sim a$.

PROOF: Suppose this does not hold. We know (Lemma 1) that a.s. there is a subsequence $a_{n_j} \to a$. Then, for almost all j, $a_{n_j} \notin N_{n_j}$, and, consequently, there is $b_{n_j} \in N_{n_j}$, such that $a_{n_j} \prec b_{n_j}$. Since $\mathscr{A}$ is compact, there is a convergent subsequence, say $b_{n_{j_k}} \to b$, $b \in \mathscr{A}$. But now, taking into account $a_{n_{j_k}} \to a$, $a_{n_{j_k}} \prec b_{n_{j_k}}$ and the continuity of the evaluation functions, we have $a \preceq b$. But a was nondominated, and, consequently, $a \sim b$. □

The result implies that, for the b found, $\Psi(a, w) = \Psi(b, w)$, $\forall w \in S$. Then

COROLLARY 1: Under the conditions of Lemma 1 and identifiability condition, i.e., when $\Psi(a, w) = \Psi(b, w)$, $\forall w \in S \Rightarrow a = b$, we have that $\forall a \in N(\mathscr{A})$, a.s. there is a sequence (a_{n_j}), with $a_{n_j} \in N_{n_j}$ *such that* $a_{n_j} \to a$.

The above results suggest using N_n as a discrete approximation to the nondominated set.

Note that the proof is simpler and the result is more general than that given in [15]. The reason lies mainly in our increasing population size at each step of the algorithm, and also, in effectively sampling from the whole feasible set and, consequently, from the whole efficient set. This removes dependence from the initial population in the evolutionary algorithms. Obviously, this may be at the expense of a higher computational cost, although we must emphasize our interest in the approximation of the *whole* efficient set.

Two examples describe several subtleties that need to be dealt with before this strategy may be used routinely in applications.

3. EXAMPLES

We consider two simple examples which will allow us to motivate various implementation issues. The first one is a standard biobjective linear programming problem. The second example has a nonconnected nondominated set.

3.1. Example 1: Multiobjective Linear Programming

Consider the multiobjective programming problem, with feasible region $\mathcal{A}$ of alternatives (x_1, x_2) defined by

$$2.7x_1 + x_2 \le 3,$$

$$x_1 + x_2 \le 1.5,$$

$$\frac{x_1}{3} + x_2 \le 1.1,$$

$$0 \le x_1, x_2 \le 1,$$

and objective functions $v_1(x_1, x_2) = x_1$, $v_2(x_1, x_2) = x_2$. Clearly, in this case, $x \preceq y \Leftrightarrow x_1 \le y_1, x_2 \le y_2$ which is checked easily. Using the notation of Section 2, we have $S = \{1, 2\}$, and for $a = (x_1, x_2)$, we have $\Psi(a, w) = v_w(a)$, $w \in S$. The nondominated set is also easily obtained, e.g., graphically. Suppose we sample uniformly from $\mathcal{A}$, e.g. with the rejection method (see [23]). Panel (a) in Figure 1 shows the initial sample (200 points), whereas panel (b) shows the approximation to the nondominated set based on the 13 points in the nondominated set of the sample. Panel (d) shows the approximation when a bigger sample is taken. Note how the approximation improves as the sample size grows (25 nondominated points from a sample with 500 points). Other panels will be described later.

3.2. Example 2: Multiobjective Nonlinear Programming

Suppose now that the feasible region is the set $\mathcal{A}$ in $\mathbf{R}^2$ given by

$$x_2 - \sqrt{16 - x_1^2} \le 6,$$

$$x_2 + \sqrt{1 - (x_1 - 5)^2} \le 4,$$

$$x_2 - \sqrt{9 - (x_1 - 5)^2} \le 0,$$

$$0 \le x_1 \le 8,$$

$$0 \le x_2 \le 10,$$

with preferences defined by

$$(x_1, x_2) \preceq (y_1, y_2) \Leftrightarrow (x_1 \le y_1, x_1 + 5x_2 \le y_1 + 5y_2).$$

Now, $S = \{1, 2\}$ and for $a = (x_1, x_2)$, $\Psi(a, 1) = x_1$, $\Psi(a, 2) = x_1 + 5x_2$. This is a case in which conventional methods would have problems because of the nonconvexity of the feasible region and, as we shall see, the nonconnectedness of the nondominated set. Again, we sample uniformly, see panel (a) of Figure 2. Panels (b) and (d) show two approximations, for nested samples. Similar comments to Example 1 would follow. Note that the approximations seem to suggest the mentioned nonconnectedness of the nondominated set.

4. IMPLEMENTATION DETAILS

The previous examples raise several issues that need to be addressed before our basic strategy becomes effective. The first one is that since we are especially interested in continuous problems, we should probably look for continuous approximations to the non-dominated set. The second one refers to stopping rules: We expect approximations to improve as the sample size grows, but we must introduce rules to stop sampling. Other issues concern sampling schemes other than uniform. In the examples, we used this one, that is, all points in the feasible set have the same likelihood to be in the sample. In general problems, we may need to use rejection sampling (see Section 4.3). This may be wasteful in two senses: If we use rejection sampling, for some feasible regions, we might generate many nonfeasible solutions; by uniformly sampling, we might be generating far

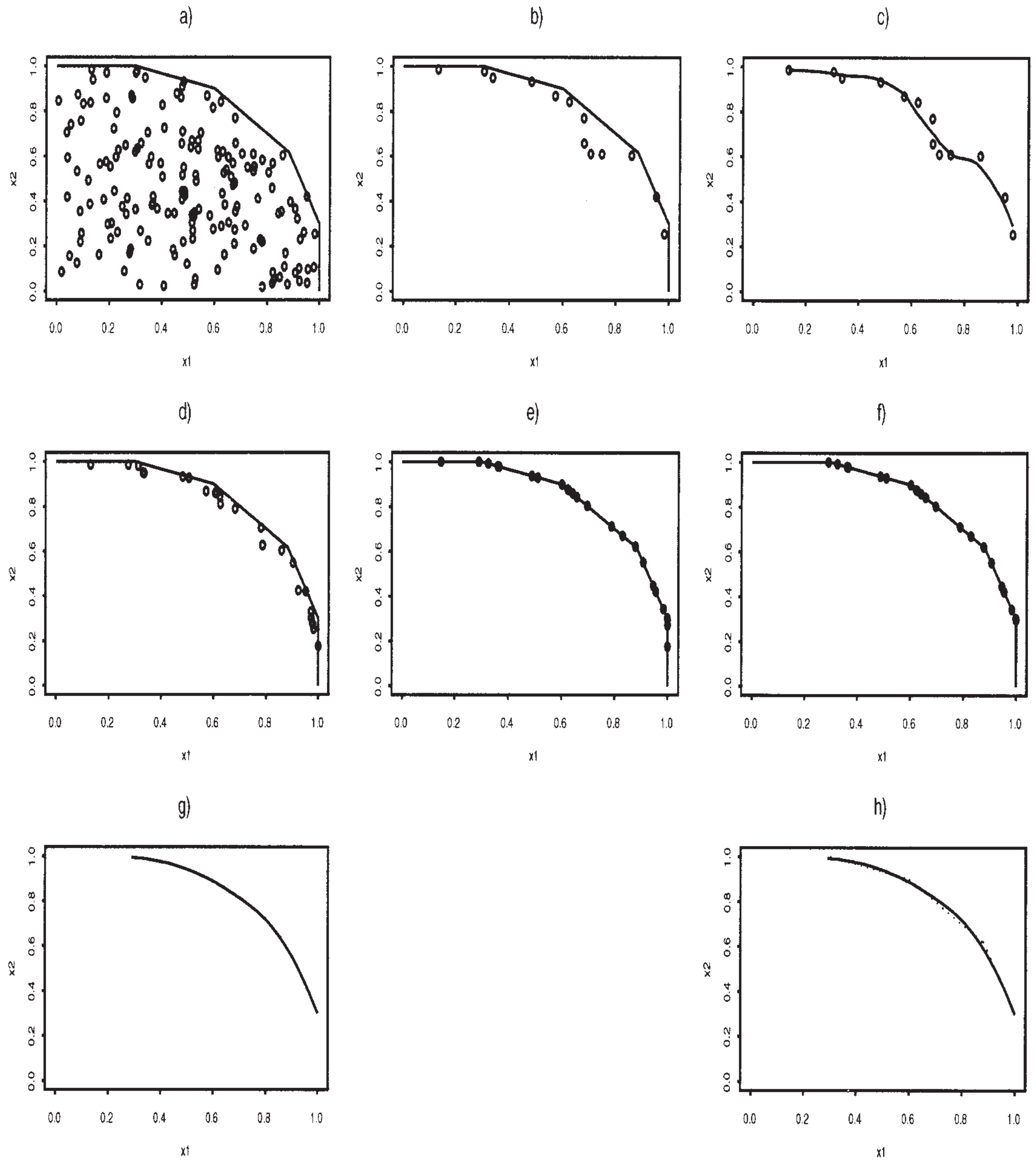

Figure 1. Example 1.

apart from the nondominated set. Finally, we should look for procedures to check for dominance. We address these issues now.

4.1. A Continuous Approximation

We have suggested using N_n as a discrete approximation to the nondominated set. When $\mathcal{A}$ is continuous, we may be interested in having a continuous approximation. We could pose the approximation problem of fitting a surface $\Phi_n(a) = 0$ through the points of N_n. We may use one of the coordinates, say the mth one, of the approximate alternatives as a response variable, the other $m - 1$ coordinates as explanatory variables and the points of N_n as data.

In general, this will be a nonlinear regression problem, possibly in high dimensions and with little structural information. This begs for a robust regression method. In our implementation, we use a local regression method, called *loess* [6], as implemented in S-Plus [33]. Local regression models provide nonparametric regression methods, with the underlying assumption that there is a neighborhood containing the explanatory variables in which the regression sur-

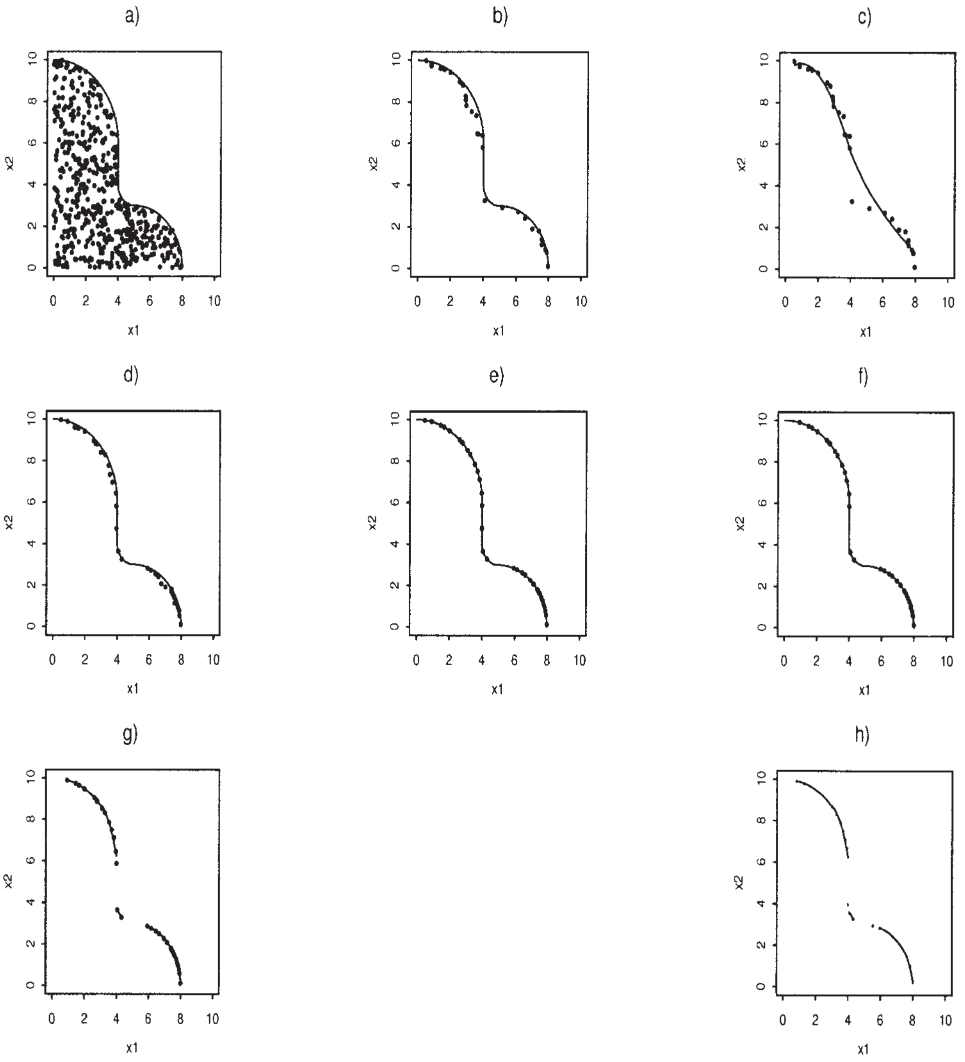

Figure 2. Example 2.

face is well approximated by a function from a specific parametric class. Linear and quadratic polynomials are the functions generally used. The sizes of the neighborhoods, defined via the Euclidean distance, are specified by a parameter, which determines the smoothness of the local regression fit. There are many other possibilities including interpolation methods from Numerical Analysis or neural networks.

Then, once we have fitted the curve, we suggest the set $\mathcal{A} \cap \{a : \Phi_n(a) = 0\}$ as the approximation to the nondominated set. This set is computed using the information from the local regression model and the nondominated points from the last sample. Panel (c) in Figure 1 shows our continuous approximation in Example 1 based on the discrete approximation in Figure 1(b).

We may gain further computational efficiency, when the evaluation functions Ψ are increasing in a, possibly after a convenient transformation (as when working on the image set). In such case, $a \le b \Rightarrow \Psi(a, w) \le \Psi(b, w)$, $\forall w \in S \Rightarrow a \preceq b$. Loess (and other regression methods) adapt easily to that case, producing a function Φ_n such that $a \le b$ implies $\Phi_n(a) \le \Phi_n(b)$. In this monotonic case, we know that the efficient set will be in the frontier of $\mathcal{A}$. Hence, we could think of sampling uniformly from that set. However, this problem is extremely involved, even in the linear case (see, e.g., [4]). Since we know that efficient points are in the frontier, we could project the points of N_n onto the frontier, by searching in the direction of the gradient of Φ_n, if available, or simply in the direction $(1, 1, \ldots, 1)$; eliminate, if any, the dominated projections, and smooth through the remaining solutions. Figure 1(e) shows the projections from solutions in (d); (f) has filtered the dominated solutions, and (g) shows the continuous approximation, which we compare with the actual nondominated set in Figure 1(h). Note that there are only some differences near the extreme points, but this is not surprising, since there is a small probability of sampling near them.

The proposed approach would however fail in Example 2, because the nondominated set is not connected. In this case, prior to undertaking the smoothing, we could proceed by clustering the solutions in N_n and then applying a smoother in each cluster. Figure 2(c) shows the smoother fitted without clustering; Figures 2(e) and 2(f) represent the projection of the discrete approximation in 2(d) and the filtered projections, respectively. Figure 2(g) includes the smoothed approximations based on three clusters detected, which is compared in Figure 2(h) with the actual nondominated set. We used an agglomerative hierarchical clustering method with the S-Plus program, via the *hclust* function. The dissimilarity matrix was built with the Euclidean distance, and the between-cluster dissimilarity was defined by using the nearest neighbor method.

In more general cases, we could fit a surface $\Phi_n(a) = 0$ through the pairs $(a, b(a))$ with $b(a) = 1$ if $a \in N_n$, and $b(a) = 0$ if $a \notin N_n$ or is infeasible. Logistic regression models, (see, e.g., [1]) are relevant in this context. This method has a number of advantages over the previous one, at the price of requiring more statistical sophistication. For example, an interesting point is that we may view $\Phi_n(a)$ as the probability of a belonging to the nondominated set. However, this criterion must be used cautiously as close points will tend to have similar Φ_n values, due to the continuity of this function. Consequently, it would be better viewed as an indicator of the closeness to a nondominated solution.

4.2. Stopping Rules

We have shown how the set N_n approximates the nondominated set as the sample size grows, and suggested a continuous approximation based on N_n. Clearly, we need criteria to determine when the approximation is good enough and stop sampling. All conceivable rules should take into account the effort wasted in generating new solutions and discovering new nondominated solutions. Per se, this is an extremely interesting problem, and there is related work in the area of stopping rules for stochastic global optimization methods (see, e.g., [22]).

We propose two types of stopping rules:

1. For problems with evaluation functions increasing in a (possibly after some transformation), we have the following scheme:
   (a) Fit a smoother Φ_n to N_n.
   (b) Generate another sample from $\mathscr{A}$. If only a small enough proportion (say 5%) of the points is above the smoother, stop. Else, compute a new approximation.

   Then, we are discarding the alternatives a with $\Phi_n(a) < 0$ (geometrically, those that are below the smoother). Examples 1 and 2 used this rule.

2. In the more general case in which we fit Φ_n through the pairs $(a, b(a))$, we suggest:
   (a) Fit a smoother Φ_n to the set $\{(a, b(a)) : a \in A_n\}$.
   (b) Generate a sample A_{n+1} from $\mathscr{A}$.
   (c) Compute $\Phi_n(a)$, $\forall a \in \mathscr{A}_{n+1}$.
   (d) Compute N_{n+1}.
   (e) Check the discrepancy between the fitted values $\Phi_n(a)$ and the observed values $b(a)$, by

$$D = \sum_{a \in A_{n+1}} \frac{(\Phi_n(a) - b(a))^2}{|A_{n+1}|},$$

   where $|\cdot|$ denotes the cardinal of a set.
   (f) Stop, either, if

- D is very small (say, less than 0.15), or
- D has not decreased sufficiently during the last steps.

Consider Example 1. We apply this approach to samples of sizes 100, 500, and 1000. The value of D for the second sample (with respect to the logistic regression applied to the first sample) is 0.2913. For the third sample with respect to the second one, D is 0.1824. We may consider it small enough and stop or we may continue the sampling process. Figure 3(a) shows the relationship between actual values $b(a)$ and fitted values $\Phi_{n(a)}$ for the second sample. Figure 3(b) shows the same feature for the third sample. Observe a higher concentration of points near vertices (0, 0) and (1, 1) in Figure 3(b) since both are the best prediction points with the logistic regression model [predict 1 when $b(a) = 1$ and predict 0 when $b(a) = 0$].

4.3. Sampling Scheme and Checking Dominance

To sum up, the proposed procedure is:

```
1. Initialization: $n = 1$, $N_0 = \emptyset$
2. Generate a sample $A_n$ from $\mathscr{A}$
3. Compute $N_n = N(A_n \cup N_{n-1})$
4. Until stopping rule holds,
    Cluster $N_n$
    Fit $\Phi_n$ to $N_n$
    $n = n + 1$
    Go to 2
5. Project $N_n$ onto the frontier
6. Compute nondominated alternatives
7. Cluster
8. Fit a surface to each cluster
```

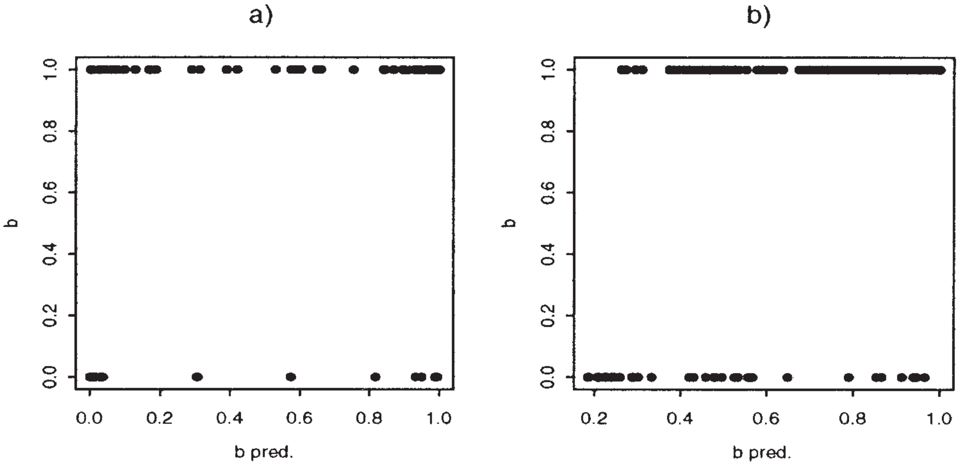

Figure 3. Relationship between $\Phi_n(a)$ (b pred.) and $b(a)$ for Example 1.

Obviously, steps 5–8 will be appropriate for increasing objective functions. Steps 6–8 will improve the approximation even more, as seen above.

We now provide further insights about steps 2 and 3. For step 2 (sampling), we propose two strategies:

a. *Rejection.* We can draw points from the feasible region with the rejection method. We first bound the region. For that, we solve $2n$ optimization problems:

$$x_i^* = \sup\{x_i, x \in \mathcal{A}\}, \qquad x_{i*} = \inf\{x_i, x \in \mathcal{A}\},$$
$$i = 1, 2, \ldots, n.$$

Note that we only search for bounds for the different values rather than accurately solving the optimization problems. Thus, $\mathcal{A} \subseteq \Pi_{i=1}^n [x_{i*}, x_i^*]$. Now we draw points from this region and reject those which do not belong to $\mathcal{A}$. Sampling from intervals like $[x_{i*}, x_i^*]$ is straightforward.

The region is better fitted if we use stratified sampling, that is, we divide the region into subsets (strata) and a sample is generated from each stratum. For instance, we can divide $[x_{1*}, x_1^*]$ into a partition with m regions $X_1^1, \ldots, X_m^1$ and, for each one, solve the problems:

$$x_{ij}^* = \sup\{x_i, x \in \mathcal{A} \cap \{x : x_1 \in X_j^1\}\},$$
$$x_{ij*} = \inf\{x_i, x \in \mathcal{A} \cap \{x : x_1 \in X_j^1\}\}, i = 2, \ldots, n.$$

We get

$$\mathcal{A} \subseteq \bigcup_{j=1}^{m} \{X_j^1 \times \prod_{i=2}^{n} [x_{ij*}, x_{ij}^*]\}.$$

Sampling from $\cup_{j=1}^m \{X_j^1 \times \Pi_{i=2}^n [x_{ij*}, x_{ij}^*]\}$ may be undertaken by *composition* (see [23]).

b. *Adaptive sampling.* The following technique is based on the ideas of *adaptive sampling* [32]. We build a grid which grows until it contains the whole feasible region. Suppose $\mathcal{A}$ is connected:

1. Find an initial feasible solution.
2. Generate a grid around it with s^m cells (s is the number of partitions made at each of the m dimensions).
3. Generate $t > 1$ points at each cell of the grid.
4. Check the feasibility of each point.
5. Add to the grid every cell that adjoins any cell with feasible points.
6. Generate points on the new cells.
7. If stopping rule does not hold, go to 5. Otherwise, stop.

The generation of points at each cell will be carried out whenever the cell has not been visited before. The algorithm will stop when there is no new cell with feasible points. Note that the grid has to be constructed once and it can be reused later.

As far as the computation of nondominated alternatives is concerned, some improvements follow:

- *Use of nadir point.* Mateos, Ríos-Insua, and Nevado [19] provide approximation methods to the nondominated set in multicriteria decision-making. One of these methods, based on a nadir point generalisation, removes feasible points which are dominated. By taking advantage of this technique, we can reduce the region to be explored. It can be also used as a pretest. For example, suppose $\mathcal{A} \subset \mathbf{R}^2$ and we have two criteria Ψ_1, Ψ_2 increasing in both components. We compute the optimal solutions x_1^*, x_2^* for both criteria and evaluate $\Psi_{1*} = \Psi_1(x_2^*)$ and $\Psi_{2*} = \Psi_2(x_1^*)$. Then, any generated point z is removed from the sample if $\Psi_1(z) \leq \Psi_{1*}$ and $\Psi_2(z) \leq \Psi_{2*}$.

Another alternative is to find, if possible, $y = (y_1, y_2)$ such that $\Psi(y) = \Psi_{1*}$ and $\Psi(y) = \Psi_{2*}$. All the points (z_1, z_2) such that $z_1 < y_1$ and $z_2 < y_2$ are dominated, reducing the region of points to be generated.

- *Information from the logistic regression.* If we use a grid to generate from the feasible region and a logistic regression model, as described above, we can estimate the probability, p_j, of being nondominated at each mean point of each cell j. This quantity p_j gives us information to generate additional points at that grid: At each cell, we generate a number of points proportional to the probability of being nondominated. Thus, in order to have s points in all, we generate at each cell j, $R[s \cdot p_j/\Sigma_k p_k]$ points, where k ranges over all the cells and $R[x]$ denotes the roundoff of x. Therefore, the total number of generated points, without taking into account roundoffs, is $\Sigma_j \; s \cdot p_j/\Sigma_k \; p_k = s$.
- *Evolutionary algorithm.* We can employ evolutionary algorithm techniques in combination with our proposal. Given the nondominated set, the new population could consist of: nondominated points of the current population, points uniformly sampled from $\mathcal{A}$, and points generated by crossover or mutation from the last population nondominated points. Note these changes modify neither Lemma 1 nor Proposition 1 since points of each population are generated from $\mathcal{A}$ independently. The next subsection illustrates the ideas with an example taken from [3].

4.4. Example 3

Benson and Sayin [3] propose working on the image set instead of the alternative set, because the dimension is usually smaller. Moreover, many efficient points may have the same point as image, and we make unnecessary computations. Also, for decision-makers, it is easier to choose a solution to the outcome space than on the feasible set. These and other arguments may be seen in [10, 11, 12].

Example 1 would be considered as a problem solved in the image space since the objective functions coincide with the coordinates. For that reason, let us analyze the example proposed in [3], working on the alternative set, rather than on the image set, to show the power of our method.

Consider the problem

$$\mathcal{A} = [0,\, 1]^{10}$$

and

$$v_1(x) = -\sum_{i=1}^{4} x_i + \sum_{i=5}^{8} 0.667x_i - \sum_{i=9}^{10} 0.75x_i,$$

$$v_2(x) = \sum_{i=1}^{4} x_i - \sum_{i=5}^{8} 0.333x_i + \sum_{i=9}^{10} 0.25x_i.$$

We apply our method with the evolutionary algorithm variation.

Figure 4 shows the algorithm evolution. Initially, we generate a population of size 200 in $\mathcal{A}$. We compute the nondominated points and undertake the smoothing. We proceed to obtain the next populations (with size 1.3 × last population size) by using the last nondominated points, the points generated from them, and points uniformly sampled from $\mathcal{A}$. Specifically, for each nondominated point, we generate 1 point from a 10-dimensional normal distribution, with mean equal to the nondominated point and variance–covariance matrix $0.1I$, where I is the identity matrix. This may be interpreted as a mutation. A point that is not in the feasible region is rejected, and another point is drawn. Moreover, for each parent pair randomly chosen, we generate four offspring by crossover as follows: Given two parents $x = (x_1, x_2, \ldots, x_{10})$ and $y = (y_1, y_2, \ldots, y_{10})$, a random number between 1 and 5 is chosen, say 3. The children would then be $(x_1, x_2, x_3, y_4, y_5, \ldots, y_{10})$ and $(y_1, y_2, y_3, x_4, x_5, \ldots, x_{10})$. For the same parents, the same process is carried out generating a number between 6 and 9. Since all the variables belong to [0, 1], all the children belong to the feasible set. Note that, due to the increase in the population size, we can have two children per parent and, also, a parent mutation. Obviously, other alternative ways to proceed might be devised: mutate children, define more parent pairs, ... Panels 1–12 in Figure 4 show the fitted surface in the image set through the different generations. We have considered the first type of stopping rule given in Section 4.2, stopping when 90% of the generated points are below the smoother. Although many panels seem equal, note that the number of nondominated points is, respectively, 22, 28, 45, 72, 75, 110, 153, 166, 200, 243, 298, and 351. Panel 13 shows the resulting points on the image set after performing the projection onto the feasible set frontier. Note that we do not project onto the image set frontier, because we do not work on this set. Panel 15 shows our final approximation to the efficient set. Both the sampling and the projection have been carried out in the alternative set, whereas the stopping rule has been defined in the image set.

We must emphasize the algorithm good performance in a problem with 34 extreme points, 68 efficient edges, one 2-dimensional efficient face, and two 4-dimensional efficient faces (see [3]).

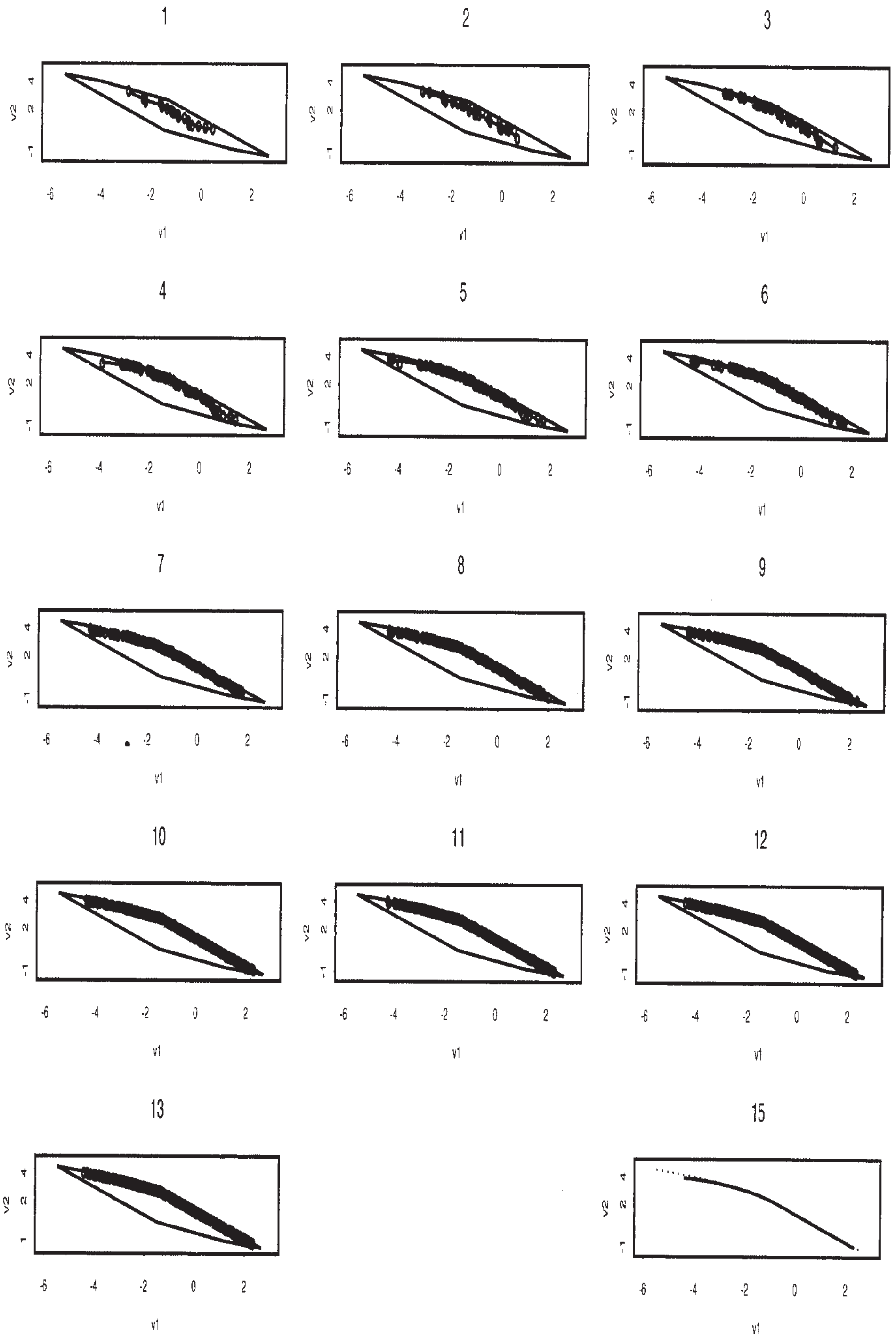

Figure 4. Example 3.

4.5. Computational Discussion

The proposed approach is specially useful when the objective functions are monotonic. However, one of its main strengths is that it is possible to apply the approach to other functions by suitably changing the stopping rule, e.g., using rule 2 of Section 4.2 and removing the projection onto the frontier. It can be used also by performing the local regression on the image of the sample nondominated points.

Mean CPU times for the examples (50 executions) have been: Example 1, 4.1 s; Example 2, 14.6 s; Example 3, 464 s. The costliest step is pairwise dominance checking. It is important to do the relabeling (step 2) since it substantially reduces this cost: in Examples 1 and 3, mean CPU times increase to 4.9 and 19.1 s, respectively, if relabeling is not used. In order to apply clustering and regression techniques, it is convenient to use statistical software. Our examples have been run with the S-Plus program for Windows 98, in a Pentium IV 1.8 GHz PC.

As far as parameters are concerned, we recommend an initial population size of 100. The size increases multiplying this value by a number in [1.2, 2]. Smaller values may yield a lack of points above the smoother and problems with stopping rule 1 from Section 4.2. Higher values involve a high computational burden. The 1.3 value seems to be especially effective.

5. CONCLUSIONS

This research has provided a stochastic approach to approximate the nondominated set in general problems. It is general enough to cope with fairly unstructured contexts, yet it permits the efficient incorporation of partial information, allowing us therefore to take advantage of any special structure available. In this sense, it may be competitive even in cases in which there is a structured method like a multiobjective simplex procedure, given the enormous computational complexity of those ones. Our method may be viewed as an evolutionary algorithm in which the population size is variable, and the objective is not to obtain a point from the efficient set but a representation of it.

Moreover, many problems of interest in areas such as multiobjective programming, robust Bayesian analysis, stochastic dominance, and experimental design, may be described with our formulation, opening up new fields of application. In many of these cases, S will not be finite. For example, in stochastic dominance problems, alternatives are ordered through their expected utilities, but there is imprecision in both the utility function u and the probability distribution p (see [27]). Then,

$$a \preceq b \Leftrightarrow \left(\int u(a,\theta)\,dp(\theta) \le \int u(b,\theta)\,dp(\theta),\ \forall\, u \in U,\ \forall\, p \in P \right),$$

where U is a class of utility functions and P is a class of probability distributions. Therefore, $w = (u, p)$, $\Psi(a, w) = \int u(a, \theta)\, dp(\theta)$ and $S = U \times P$.

Finally, ongoing research is directed towards extending some of the ideas suggested throughout the paper, like using a neural network as a smoother or trying other stopping rules.

ACKNOWLEDGMENTS

Research supported by projects from CAM, MEC (TSI2004-06801-C04-01, -03 and -04), URJC, MCYT (DPI2001-3731), the European Science Foundation TED programme, and the DMR Consulting Foundation Decision Engineering Lab. We are very grateful for the referee's and associate editor's valuable suggestions.

REFERENCES

[1] A. Agresti, Categorical data analysis, Wiley, New York, 1990.

[2] R. Armann, Solving multiobjective programming problems by discrete representations, Optimization 20 (1989), 483–492.

[3] H.P. Benson and S. Sayin, Towards finding global representations of the efficient set in multiple objective Mathematical Programming, Naval Res Logist 44 (1997), 47–67.

[4] H. Berbee, C. Boender, A. Rinnooy Kan, C. Scheffer, R. Smith, and J. Telgen, Hit-and-run algorithms for the identification of non-redundant linear inequalities, Math Program 37 (1987), 184–207.

[5] J. Branke, T. Kaußler, and H. Schmeck, Guidance in evolutionary multi-objective optimization, Adv Eng Software 32 (2001), 499–507.

[6] W.S. Cleveland, E. Grosse, and W. Shyu, "Local regression models," Statistical models in S, J.M. Chambers and T.J. Hastie (Editors), Wadsworth & Brooks, Pacific Grove, CA, 1992, pp. 309–376.

[7] M. Clyde and K. Chaloner, The equivalence of constrained and weighted designs in multiple objective design problems, J Amer Statist Assoc 91 (1996), 1236–1244.

[8] C.A. Coello, A comprehensive survey of evolutionary-based multiobjective optimization techniques, Knowledge Inform Syst 1 (1999), 129–156.

[9] C.A. Coello and A.D. Christiansen, Multiobjective optimization of trusses using genetic algorithms, Comput & Structures 75 (2000), 647–660.

[10] J.P. Dauer, Analysis of the objective space in multiple objective linear programming, J Math Anal Appl 126 (1987), 579–593.

[11] J.P. Dauer and Y. Liu, Solving multiple objective linear programs in objective space, European J Oper Res 46 (1990), 350–357.

[12] J.P. Dauer and O.A. Saleh, Constructing the set of efficient objectives values in multiple objective linear programs, European J Oper Res 46 (1990), 358–365.

[13] C.M. Fonseca and P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evolutionary Comput 3 (1995), 1–16.

[14] W.B. Gearhart, Comprise solutions and estimation of the noninferior set, J Optim Theory Appl 28 (1979), 29–47.

[15] T. Hanne, On the convergence of multiobjective evolutionary algorithms, European J Oper Res 117 (2000), 553–564.

[16] J.P. Ignizio and T.M. Cavalier, Linear programming, Prentice Hall, Englewood Cliffs, NJ, 1994.

[17] H. Levy and M. Robinson, Stochastic dominance: Investment decision making under uncertainty, Kluwer, Boston, 1998.

[18] D.T. Luc, Theory of vector optimization, Springer, Berlin, 1989.

[19] A. Mateos, S. Ríos-Insua, and G. Nevado, Solving discrete multiobjective optimization problems based on approximation, OPSEARCH 35 (1998), 93–105.

[20] M. Mebarki and P. Castagna, An approach based on Hotelling's test for multicriteria stochastic simulation-optimization, Simulation Practice Theory 8 (2000), 341–355.

[21] V.D. Noghin, Estimation of the set of nondominated solutions, Numer Funct Anal Optim 12 (1991), 507–515.

[22] A.H.G. Rinnooy Kan and G. Timmer, "Global optimization," Optimization, G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd (Editors), North-Holland, Amsterdam, 1988, Vol. 1, pp. 631–662.

[23] B. Ripley, Stochastic simulation, Wiley, New York, 1987.

[24] D. Ríos Insua, Sensitivity analysis in multiobjective decision making, Lecture Notes in Economical and Mathematical Systems 347, Springer, Berlin, 1990.

[25] D. Ríos Insua and S. French, A framework for sensitivity analysis in discrete multiobjective decision making, European J Oper Res 54 (1991), 176–190.

[26] D. Ríos Insua and J. Martín, Robustness issues under imprecise beliefs and preferences, J Stat Plan Inference 40 (1994), 383–389.

[27] D. Ríos Insua and F. Ruggeri (Editors), Robust Bayesian analysis, Lecture Notes in Statistics 152, Springer, Berlin, 2000.

[28] R.S. Rosenberg, Simulation of genetic populations with biochemical properties, Ph.D. thesis, University of Michigan, Ann Harbor, MI, 1967.

[29] R. Sarker, K. Liang, and C. Newton, A new multiobjective evolutionary algorithm, European J Oper Res 140 (2002), 12–23.

[30] R.E. Steuer, Multiple criteria optimization: Theory, computation and application, Wiley, New York, 1986.

[31] J. Teghem, D. Tuyttens, and E.L. Ulungu, An interactive heuristic method for multiobjective combinatorial optimization, Comput Oper Res 27 (2000), 621–634.

[32] S.K. Thompson, Sampling, Wiley, New York, 1992.

[33] W. Venables and B. Ripley, Modern applied Statistics with S-Plus, Springer, New York, 1994.

[34] D.J. White, Optimality and efficiency, Wiley, Chichester, 1982.

[35] D.J. White, Weighting factor extensions for finite multiple objective vector minimisation problems, European J Oper Res 36 (1988), 256–265.

[36] D.J. White, A bibliography on the applications of mathematical programming multiple-objective methods, J Oper Res Soc 41 (1990), 669–691.

[37] D.J. White, Multiple-objective weighting factor auxiliary optimization problems, J Multi-Criteria Decision Anal 4 (1995), 122–132.

[38] P.L. Yu, Multiple-criteria decision making: Concepts, techniques and extensions, Plenum, New York, 1986.