Norm-Based Approximation in Bicriteria Programming[1]

Bernd Schandl[2] Kathrin Klamroth[3] Margaret M. Wiecek[4]

Abstract

An algorithm to approximate the nondominated set of continuous and discrete bicriteria programs is proposed. The algorithm employs block norms to find an approximation and evaluate its quality. By automatically adapting to the problem's structure and scaling, the approximation is constructed objectively without interaction with the decision maker. Examples and case studies are included.

Key Words. Approximation, nondominated points, bicriteria programs, block norms.

1 Introduction

In view of increased computational power and enhanced graphic capabilities of computers, approximation of the solution set for bicriteria programming has been a research topic of special interest. Since bicriteria programs feature only two criteria, their solution set can be visualized graphically which significantly faciliates decision making. In this vein, researchers have given

[1]This work was partially supported by ONR Grant N00014-97-1-0784.

[2]Research Assistant, Department of Mathematical Sciences, Clemson University, Clemson, SC.

[3]Associate Professor, Department of Computer Science and Mathematics, University of Applied Sciences Dresden, Dresden, Germany.

[4]Associate Professor, Department of Mathematical Sciences, Clemson University, Clemson, SC.

special attention to developing approximation methods that yield a representation or description of the solution set rather than to further studying scalarization approaches that had been extensively examined earlier.

Below we review approximation approaches specifically developed for bicriteria programs. We focus on methods that are based on exact algorithms for the solution of scalarization problems and were applied to example problems.

Cohon (1978) and Poliščuk (1979) independently develop similar approximation approaches for linear and convex bicriteria problems, respectively. The weighted-sum scalarization is employed to find nondominated points and the l_2 norm is used as an estimate of the accuracy of the approximation. Fruhwirth et al. (1989) propose a sandwich algorithm to approximate a convex curve in $\mathbb{R}^2$ and apply it to the bicriteria minimum cost flow problem. The curve is approximated by two piecewise linear functions, one above and one below the curve. The curve's derivative is used to partition a coordinate axis. Yang and Goh (1997) use the derivative of the upper approximation instead. For both algorithms the approximation error decreases quadratically with the number of approximation points. Jahn and Merkel (1992) propose a reference-point-approach for general bicriteria programs and give attention to avoid finding local optima. The approach produces a piecewise linear approximation of the nondominated set. Payne (1993) proposes to approximate the nondominated set of a general bicriteria problem by rectangles, each defined by two nondominated points. Das (1999) briefly discusses an approach based on the Normal-Boundary Intersection technique. A direction orthogonal to a line defined by two nondominated points is used to find a new nondominated point. The identified point has the maximal l_∞ distance from the approximation in the considered region.

The following two approaches are, to our knowledge, the only ones that give a closed-form formula for an approximating function of the nondominated set rather than a set of approximating points or a piecewise linear approximation of the nondominated set. Approximating the nondominated set of a convex bicriteria problem by a hyper-ellipse is proposed in Li et al. (1998) and Li (1999). The technique requires three nondominated points and their choice affects the quality of approximation. In Chen et al. (1999) and Zhang et al. (1999), quadratic functions are used to locally approximate the nondominated set of a general bicriteria problem in a neighborhood of a nondominated point of interest. By performing the procedure for several nondominated points, a piecewise quadratic approximation of the whole nondominated set can be generated.

In this paper, we propose to approximate the solution set of bicriteria programs by means of block norms. Using block norms to generate nondominated points has several implications: the norm's unit ball approximates the nondominated set and, at the same time, the norm evaluates the feasible points as well as the quality of the current approximation.

Let $x \leqq y$ denote $x_i \leq y_i$ for $i = 1, 2$, and $x \leq y$ denote $x \leqq y$ and $x \neq y$. We consider the following general bicriteria program:

$$\begin{array}{ll} \min & \{z_1 = f_1(x)\} \\ \min & \{z_2 = f_2(x)\} \\ \text{s.t.} & x \in X, \end{array} \tag{1}$$

where $X \subseteq \mathbb{R}^m$ is the *feasible set* and $f_1(x)$ and $f_2(x)$ are real-valued functions. We define the *set of all feasible criterion vectors* Z and the *set of all globally nondominated criterion vectors* N of (1) as follows

$$\begin{aligned} Z &= \{z \in \mathbb{R}^2 : z = f(x), x \in X\} = f(X) \\ N &= \{z \in Z : \nexists \tilde{z} \in Z \text{ s.t. } \tilde{z} \leq z\}, \end{aligned}$$

where $f(x) = \big(f_1(x), f_2(x)\big)^T$. We assume that the set Z is closed and nonempty, and that there exists a point $u \in \mathbb{R}^2$ so that $Z \subseteq u + \mathbb{R}^2_{\geqq}$ where $\mathbb{R}^2_{\geqq} := \{x \in \mathbb{R}^2 : x \geqq 0\}$. It follows that the set N is nonempty, see, for example, Sawaragi et al. (1985, pp. 50–51).

The point $z^* \in \mathbb{R}^2$ with

$$z_i^* = \min\{f_i(x) : x \in X\} - \epsilon_i \qquad i = 1, 2$$

is called the *utopia (ideal) criterion vector* where the components of $\epsilon =$

$(\epsilon_1, \epsilon_2) \in \mathbb{R}^2$ are small positive numbers.

The point $z^\times \in \mathbb{R}^2$ with

$$z_i^\times = \min\Big\{ f_i(\bar{x}) : f_j(\bar{x}) = \min_{x \in X} f_j(x), j \neq i \Big\} \qquad i = 1, 2$$

is called the *nadir point*.

In Section 2, we present the methodological tools we use to construct the approximation. Section 3 contains the approximation algorithm featuring specific procedures depending on the structure of the problem. Examples and case studies illustrating the performance of the algorithm are presented in Section 4, and Section 5 concludes the paper.

2 Methodological Tools

In this section, we discuss approaches to generating nondominated points used in the proposed approximation algorithm. Furthermore, the algorithm relies on the usage of block norms, a well-known concept in convex analysis. Block norms are norms with a polyhedral unit ball. A cone generated by two neighboring extreme points of a unit ball is called a fundamental cone. The partition of the unit ball into fundamental cones is used extensively in our methodology.

Let $\gamma : \mathbb{R}^2 \to \mathbb{R}$ be an oblique norm[5] with the unit ball B. Given a reference point z^0 (without loss of generality $z^0 = 0 \in Z + \mathbb{R}^2_{\geqq}$), the following program yields a globally nondominated point, see Schandl (1999):

$$\begin{array}{ll} \max & \gamma(z) \\ \text{s.t.} & z \in Z \cap (z^0 - \mathbb{R}^2_{\geqq}). \end{array} \tag{2}$$

In the algorithm, the norm γ with the center at z^0, as used in (2), is being constructed and used to generate new nondominated points.

Solving (2) requires a calculation of the norm γ. As shown in Hamacher and Klamroth (1997), it is sufficient to know in which fundamental cone

[5]An oblique norm is a block norm where no facet of the unit ball is parallel to any coordinate axis. For details see Schandl (1999).

a point z is located to calculate its norm $\gamma(z)$. Let γ be a polyhedral norm with the unit ball $B \subseteq \mathbb{R}^2$. Let $z \in C$ where $C = \text{cone}(v^i, v^j)$ is a fundamental cone, that is, C is generated by the two extreme points v^i and v^j. If $z = \lambda_i v^i + \lambda_j v^j$, where $\lambda_i, \lambda_j \geq 0$, is the unique representation of z in terms of v^i and v^j then

$$\gamma(z) = \lambda_i + \lambda_j. \tag{3}$$

Let z^i and z^j be two nondominated points in $z^0 - \mathbb{R}^2_{\geqq}$. To guarantee that a point z is in the cone $C = \text{cone}(z^i, z^j)$, it is sufficient to require $z = \lambda_i z^i + \lambda_j z^j$ where $\lambda_i, \lambda_j \geq 0$.

Using (3), the general norm problem (2) restricted to a cone can be formulated as:

$$\begin{array}{ll} \max & \lambda_i + \lambda_j \\ \text{s.t.} & z = \lambda_i z^i + \lambda_j z^j \\ & \lambda_i, \lambda_j \geq 0 \\ & z \in Z. \end{array} \tag{4}$$

Given an optimal solution $(\bar{\lambda}, \bar{z})$ of (4), $\bar{z}$ is globally nondominated. Observe that problem (4) generates a nondominated point independently of the existence of a norm.

Besides the norm-based approach described above, we use two other techniques to generate globally nondominated solutions. Following Steuer and Choo (1983), we reformulate the lexicographic Tchebycheff method for the cone $C = \text{cone}(z^i, z^j)$:

$$\begin{array}{ll} \text{lex min} & \left(\|z - \tilde{z}^*\|_\infty^w, \|z - \tilde{z}^*\|_1 \right) \\ \text{s.t.} & z = \lambda_i z^i + \lambda_j z^j \\ & \lambda_i, \lambda_j \geq 0 \\ & z \in Z, \end{array} \tag{5}$$

where $\tilde{z}^*$ is the local utopia point for the cone, see Section 3.4. We first minimize the weighted Tchebycheff norm between the local utopia point and a feasible point. If there is no unique solution in this first step, we minimize the l_1 distance among all the solutions of the first step. Given an optimal solution $(\bar{\lambda}, \bar{z})$ of (5), $\bar{z}$ is globally nondominated, see Schandl (1999).

A direction method introduced in Pascoletti and Serafini (1984) is modified in Schandl (1999). We use this method to search for globally nondominated points in the entire set Z. Let $z^0 \in Z + \mathbb{R}^2_{\geqq}$, $d \in \mathbb{R}^2 \setminus \mathbb{R}^2_{\geqq}$ and $1 \leq p < \infty$. Then the problem

$$\begin{array}{ll} \operatorname{lex\,max} & (\alpha, \|q\|_p) \\ \text{s.t.} & z = z^0 + \alpha d + q \\ & q \in -\mathbb{R}^n_{\geqq} \\ & z \in Z, \end{array} \tag{6}$$

has a finite solution $(\bar{\alpha}, \bar{z}, \bar{q})$, where $\bar{z}$ is a globally nondominated point.

3 Approximation Algorithm

In this section, the algorithms for an $\mathbb{R}^2_{\geqq}$-convex[6], $\mathbb{R}^2_{\geqq}$-nonconvex and discrete feasible set Z are proposed. The algorithms in all three cases are very similar, so we first present the general algorithm and then point out special features of the different cases.

3.1 General Strategy

The approximation algorithm is based on the successive generation of nondominated points using the methods described in Section 2. The basic idea is to generate points in the areas where the nondominated set is not yet well approximated. The approximation quality is evaluated using the approximation itself by interpreting it as part of the unit ball of a block norm.

We explain the algorithm for the $\mathbb{R}^2_{\geqq}$-convex case using Figure 1. To start, we need a reference point $z^0 \in Z + \mathbb{R}^2_{\geqq}$. This might be a currently implemented (not nondominated) solution or just a (not necessarily feasible)

[6] A set Z is called $\mathbb{R}^2_{\geqq}$-convex if $Z + \mathbb{R}^2_{\geqq}$ is convex.

guess. Without loss of generality, we assume throughout the section that the reference point is located at the origin.

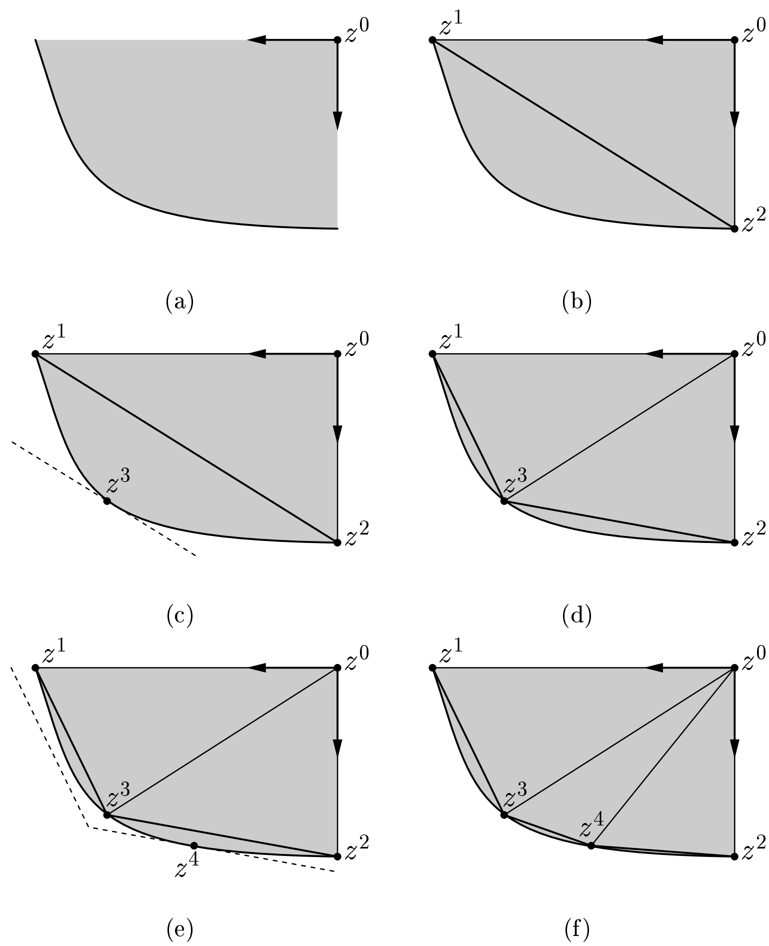


Figure 1: The steps of the approximation algorithm

To approximate the nondominated set in $z^0 - \mathbb{R}^2_{\geqq}$, we first explore the feasible set along the directions $(-1, 0)$ and $(0, -1)$ to find z^1 and z^2 using the direction method (6). These two points together with the reference point z^0 are used to define a cone and a first approximation, see Figure 1(b).

In a cone we search for a new candidate point to add to the approximation. Constructing new cones within the first cone, we get a finer approximation of the nondominated set while generating nondominated points and updating the norm. Depending on the structure of the feasible set Z ($\mathbb{R}^2_{\geqq}$-convex, $\mathbb{R}^2_{\geqq}$-nonconvex, discrete), we use the norm method (4) and/or the lexicographic Tchebycheff method (5). For more details see Sections 3.3, 3.4 and 3.5. Interpreting the approximation as the lower left part of the unit ball of a norm with z^0 as its center, we can calculate the distance of a point $\bar{z}$ from the current approximation as $\mathrm{dev}(\bar{z}) := |\gamma(\bar{z}) - 1|$, which we call the deviation of $\bar{z}$. Whenever possible, we add a point of worst approximation by substituting two new cones for the cone in which this new point is located.

3.2 Description of the Algorithm

The algorithm accepts the following input.

1. A reference point $z^0 \in Z + \mathbb{R}^2_{\geqq}$ can be specified. If it is not given then $z^0 := z^\times$ is used as a default.

2. Initial search directions can be given. There are three possibilities:

   (a) At least two directions $d^i \in -\mathbb{R}^2_{\geqq}$ are given.

   (b) An integer `randDirNo` ≥ 2 is given, which defines the number of random directions in $-\mathbb{R}^2_{\geqq}$ which are generated.

   (c) No directions are given; then the default directions $d^1 := (-1, 0)$ and $d^2 := (0, -1)$ are used.

   The directions are sorted in counterclockwise order. Let the number of directions be $k \geq 2$.

3. There are two possible stopping criteria; usually, at least one of them must be given. The first one is an upper bound $\epsilon > 0$ on the maximal deviation. As soon as we get $\mathrm{dev}(\bar{z}) < \epsilon$ for a point that should be added next, the algorithm stops. The other possibility is to give an integer `maxConeNo` ≥ 1, which specifies the maximum number of cones to be generated.

The algorithm starts by solving the direction method (6) for all directions d^i and defining l initial cones. Note that l is not necessarily equal to $k-1$ because two directions may generate the same nondominated point.

Now we find a candidate to add in each cone, each having a deviation from the current approximation associated with it. How this candidate is found differs for the three types of problems and is described in the subsections below.

Finally the main loop of the algorithm starts. If the maximum number of cones `maxConeNo` was already constructed, the loop stops. Otherwise, the candidate $\bar{z}$ with the maximum deviation is considered. If this deviation is smaller than ϵ, the loop stops. Otherwise, two new cones are constructed in place of the cone containing $\bar{z}$, candidate points for the new cones together with their deviations are calculated and the points are added to the list of candidates.

At the end of the loop the sorted list of r nondominated points is printed and can be used to visualize the approximated nondominated set. In the $\mathbb{R}^2_{\geqq}$-convex case, the approximation is in the form of an oblique norm's unit ball with an algebraic description $Az \leqq e$, where A is an $(r-1) \times n$ matrix and e is the vector of ones.

The algorithm is summarized in Figure 2. The procedure Calculate Candidate depends on the structure of the feasible set. Suitable procedures for $\mathbb{R}^2_{\geqq}$-convex, $\mathbb{R}^2_{\geqq}$-nonconvex and discrete feasible sets are given in Figures 3 and 5.

3.3 Convex Case

For the $\mathbb{R}^2_{\geqq}$-convex case, the candidate in a cone is found by the norm method (4). By taking the candidate with the maximal deviation, we globally maximize the norm and the resulting point is guaranteed to be globally nondominated. Note that the deviation is implicitly given by the solution of (4) because, due to (3), the optimal objective value of (4) is equal to the candidate's norm, that is, $\gamma(\bar{z}) = \bar{\lambda}_i + \bar{\lambda}_j$ where $\bar{z} = \bar{\lambda}_i z^i + \bar{\lambda}_j z^j$.

Given the set of extreme points of the approximation, we can easily find

```
PROCEDURE: BICRITERIA APPROXIMATION
  Read/generate z^0, d^i, ε, maxConeNo
  for all d^i do
    Solve direction method
  end for
  Construct cones
  for all cones do
    Call CALCULATE CANDIDATE
  end for
  while #cones < maxConeNo and dev(next point) ≥ ε do
    Add next point
    Construct new cones
    for all new cones do
      Call CALCULATE CANDIDATE
    end for
  end while
  Output approximation
```

Figure 2: Pseudo code of the approximation algorithm

a representation of the approximation in the form $Az \leqq e$. Since $z^0 = 0$, no line connecting two neighboring extreme points includes the origin. Given two points z^i and z^{i+1}, we calculate row i of the matrix A as follows:

$$a_{i1} = \frac{z_2^{i+1} - z_2^i}{z_1^i z_2^{i+1} - z_2^i z_1^{i+1}} \qquad \text{and} \qquad a_{i2} = \frac{z_1^i - z_1^{i+1}}{z_1^i z_2^{i+1} - z_2^i z_1^{i+1}}.$$

The procedure to calculate a candidate is summarized in Figure 3.

```
PROCEDURE: CALCULATE CANDIDATE
  Solve norm method to find z̄ and dev(z̄)
```

Figure 3: Finding a candidate in a cone for an $\mathbb{R}^2_{\geqq}$-convex problem

Setting the stopping criteria to $\epsilon = 0$ and `maxConeNo` $= \infty$ can lead to an infinite running time for a general $\mathbb{R}^2_{\geqq}$-convex set (not considering numerical problems). On the other hand, these settings can be useful for the special case of a polyhedral set Z, since in this case our algorithm is able to find the exact nondominated set.

Consider a polyhedral feasible set Z. There are two cases for the location of the points z^i and z^j when solving (4). Either both extreme points of the approximation are on the same facet or they are on different (not necessarily neighboring) facets.

Since (4) is a linear program if Z is polyhedral, its optimal solution is an extreme point or a facet of the feasible set. We thus either find a new point to add to the approximation or the identified point has a deviation of 0 in which case the cone is not considered anymore. The necessary number of iterations is $O(k)$ where k is the number of extreme points of the nondominated set because in each iteration we either find an extreme point or we eliminate a cone from further consideration.

3.4 Nonconvex Case

Finding a candidate in a cone for an $\mathbb{R}^2_{\geqq}$-nonconvex feasible set is a two-stage procedure. We first try to find a candidate "outside" the approximation; if this fails, we look for a candidate "inside". Thus we give a priority to constructing the convex hull of the nondominated set before we further investigate nonconvex areas.

Finding a candidate "outside" is done with the same method as for $\mathbb{R}^2_{\geqq}$-convex sets, that is, we use problem (4) exercising its applicability in the absence of a norm. If the deviation of the candidate found by this method is too small, that is, smaller than ϵ, we switch to a method using the Tchebycheff norm in order to investigate whether the nondominated set is $\mathbb{R}^2_{\geqq}$-convex in this cone and its approximation is already good enough or whether the nondominated set is $\mathbb{R}^2_{\geqq}$-nonconvex and a candidate has to be found in the interior of the approximation. For a cone defined by the two points z^i and z^{i+1}, we first calculate the local utopia and the local nadir point:

$$\tilde{z}^* = (z_1^i, z_2^{i+1}) \qquad \text{and} \qquad \tilde{z}^\times = (z_1^{i+1}, z_2^i).$$

Using these two points, we calculate the weights for a Tchebycheff norm whose unit ball's center is $\tilde{z}^*$ and whose upper right corner is $\tilde{z}^\times$, see Figure 4. The weights thus are

$$w_i = \frac{1}{\tilde{z}_i^\times - \tilde{z}_i^*} \qquad i = 1, 2.$$

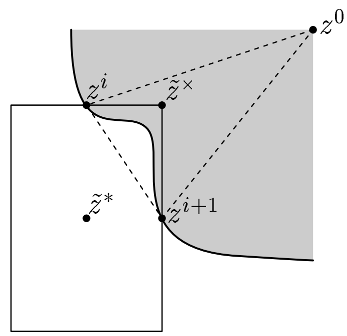


Figure 4: The Tchebycheff norm for a nonconvex area

We then use the lexicographic Tchebycheff method (5) to find a candidate for this cone. Having found a candidate $\bar{z}$, its deviation is calculated using (3). The norm can also be calculated using the equality constraint of (5) because $\gamma(\bar{z}) = \bar{\lambda}_i + \bar{\lambda}_j$ where $\bar{z} = \bar{\lambda}_i z^i + \bar{\lambda}_j z^j$.

Note that the candidate found using this two-stage procedure is not necessarily the point of worst approximation. If the candidate has already been found using program (4), it is the point of worst approximation among all points "outside" the current approximation in this cone. Finding a point with the lexicographic Tchebycheff method—that is, in the second stage—does not imply anything about how well this point is currently approximated in comparison with other points. So it might happen that we miss a point with a larger deviation than the candidate $\bar{z}$ we are considering. But unless the deviation of $\bar{z}$ is so small that the cone is not further considered, there is a good chance that the point with the larger deviation is found in a later iteration. The procedure to calculate a candidate is summarized in Figure 5.

```
PROCEDURE: CALCULATE CANDIDATE
  Solve norm method to find z̄ and dev(z̄)
  if dev(z̄) < ϵ then
    Calculate z̃* and z̃×
    Use lexicographic Tchebycheff method to find z̄
    Calculate dev(z̄)
  end if
```

Figure 5: Finding a candidate in a cone for an $\mathbb{R}^2_{\geqq}$-nonconvex problem

3.5 Discrete Case

The approach for the discrete case is exactly the same as for the $\mathbb{R}^2_{\geqq}$-nonconvex case, that is, we first use the norm method to search for a candidate "outside" the approximation; if we find none (or only one with a small deviation), we search "inside" using the Tchebycheff method.

Since using the Tchebycheff method might lead to NP-hard problems, see, for example, Warburton (1987) or Murthy and Her (1992), we develop an alternative approach for the discrete case that uses cutting planes and does not need two stages. Since this approach is not used in our implementation, we only give a brief outline and refer the reader to Schandl (1999) for more details. This approach might lead to NP-hard problems as well but there are cases where the Tchebycheff method leads to NP-hard problems while the approach based on cutting planes does not.

The idea of the cutting-plane-approach is to restrict the feasible region to an open rectangle defined by the two generators of the cone because this is the only area within this cone where nondominated points can be located. Then the norm method (4) is used to identify a candidate for this cone. If we find a candidate "outside" the current approximation, that is, a candidate with a deviation large enough, we have a point of worst approximation and a suitable point to add to the approximation. But if a point is found "inside" the approximation, it is actually a point of *best* approximation. Therefore it may happen quite often that a cone is excluded from further consideration too early.

Independently of the choice of an approach to examine the interior of the approximation, the algorithm enumerates all nondominated points if we use the stopping criteria $\epsilon = 0$ and $\texttt{maxConeNo} = \infty$.

The procedure to calculate a candidate using the Tchebycheff method is the same as for the $\mathbb{R}^2_{\geqq}$-nonconvex case, see Figure 5.

3.6 A Note on Connectedness

While the nondominated set of an $\mathbb{R}^2_{\geqq}$-convex set is always connected, see Bitran and Magnanti (1979) or Luc (1989), the nondominated set of an $\mathbb{R}^2_{\geqq}$-nonconvex problem might be disconnected. An indicator for disconnectedness is the fact that we do not find any new nondominated point in a cone, neither in the interior nor in the exterior of the approximation. Since we are able to identify disconnectedness in this way, we can remove such a cone so that the resulting final approximation is a disconnected set as well. Thus our approximation is suitable for problems with connected and with disconnected nondominated sets.

3.7 Properties of the Algorithm

The approximation algorithm for general bicriteria problems presented in this section has many desirable properties some of which are, to our knowledge, not available in any other approximation approach.

In each iteration, the subproblems (4) and/or (5) are only solved in two new cones. Thus results from previous iterations are reused and no optimization over the whole approximated region is necessary. Instead of adding an arbitrary point in each iteration, our goal is to add the point of worst approximation and to maximize the improvement in each iteration. While this property does not always hold in the $\mathbb{R}^2_{\geqq}$-nonconvex and discrete cases, it always holds in the $\mathbb{R}^2_{\geqq}$-convex case. If the algorithm is interrupted or stopped at a particular point (for example because the maximum allowed number of cones has been constructed), the approximation has a similar quality for the whole nondominated set.

While the points of the approximation are in general not nondominated or even not feasible, all *extreme* points of the approximation are nondominated. Even in the $\mathbb{R}^2_{\geqq}$-convex case, points of the approximation may be infeasible if the feasible set Z is "very thin" or even only a line. If all points of the approximation are feasible though, we have constructed an inner approximation of the nondominated set.

Using a norm induced by the problem (or, more precisely, by the approximation of its nondominated set) avoids the necessity to choose, for example, an appropriate norm, weights or directions to evaluate or estimate the quality of the current approximation. The induced norm evaluates the approximation quality and simultaneously generates suitable additional points to improve the approximation. Since the quality of the approximation is evaluated by the norm, the stopping criterion ϵ for the maximal deviation is independent of the scaling. Indeed, the norm automatically adapts to the given problem and yields a scaling-independent approximation.

Additionally, the constructed norm can be used to evaluate and compare feasible points in $z^0 - \mathbb{R}^2_{\geqq}$. A nondominated point has a norm greater or equal 1 while a norm between 0 and 1 for a point $\bar{z}$ indicates that there is a "better" point in the direction from z^0 to $\bar{z}$. The norm of a point $\bar{z}$ can be interpreted as a measure of quality relative to the maximal achievable quality in the direction of $\bar{z}$.

While it is often convenient to have the reference point generated by the algorithm, which is then the nadir point, choosing a specific reference point can be used to closely explore a particular region of the nondominated set. The automatically generated reference point can be used to construct a global approximation of the entire nondominated set while a manually chosen reference point helps to examine the structure and trade-offs of the nondominated set in a specific region. Thus the choice of the reference point can be used to "zoom into" regions of interest. For examples see Section 4.

Finally, the algorithm works essentially in the same way for $\mathbb{R}^2_{\geqq}$-convex and $\mathbb{R}^2_{\geqq}$-nonconvex problems. If the structure of the feasible set Z is unknown, we can apply the algorithm described in Section 3.4. However, if the problem is in fact $\mathbb{R}^2_{\geqq}$-convex, some additional (unnecessary) computation have to be performed. Not finding a candidate with a large enough deviation in the exterior of the approximation in the $\mathbb{R}^2_{\geqq}$-convex case is an indicator that the approximation is already good enough in the corresponding cone. In the $\mathbb{R}^2_{\geqq}$-nonconvex case though, the Tchebycheff method is used to search for a candidate in the interior of the approximation which is unnecessary in the $\mathbb{R}^2_{\geqq}$-convex case because there cannot be a nondominated point in the interior of the approximation. But the disadvantage of performing some additional calculations is clearly outweighed by the fact that no information concerning the structure of the feasible set Z is necessary. If the information that the feasible set is $\mathbb{R}^2_{\geqq}$-convex is available, the specialized algorithm presented in Section 3.3 should be used of course.

4 Examples and Case Studies

The approximation algorithm presented in Section 3 was implemented using C++, AMPL, CPLEX, MINOS and gnuplot. The C++ program keeps lists of points and cones and formulates mathematical programs which are solved by AMPL, CPLEX and MINOS. Finally, the results are written to text files which gnuplot uses to create two-dimensional plots.

4.1 Convex Example

Consider the following $\mathbb{R}^2_{\geqq}$-convex example:

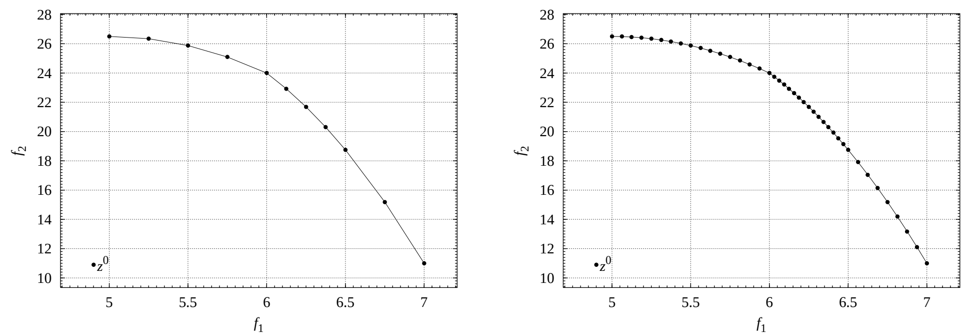
$$\begin{aligned} \max \quad & x_1 + x_2 \\ \max \quad & 10x_1 - x_1^2 + 4x_2 - x_2^2 \\ \text{s.t.} \quad & 3x_1 + x_2 - 12 \leq 0 \\ & 2x_1 + x_2 - 9 \leq 0 \\ & x_1 + 2x_2 - 12 \leq 0 \\ & x \in \mathbb{R}^2_{\geqq}. \end{aligned} \tag{7}$$

The solutions for 10 and 40 cones are shown in Figure 6. The approximation is already very good for 10 cones and improves only slightly for 40 cones. A small cusp can be seen at $f(3,3) = (6,24)$ in both figures. At this point, the first two constraints hold with equality. The first constraint defines the nondominated set to the left of the cusp, the second one to the right of the cusp.

The corresponding matrix for 10 cones, rounded to two decimals, looks as follows:

$$A^T = \begin{pmatrix} 0.49 & 0.48 & 0.463 & 0.45 & 0.42 & 0.39 & 0.25 & 0.19 & 0.12 & 0.04 \\ 0.03 & 0.03 & 0.04 & 0.04 & 0.04 & 0.05 & 0.06 & 0.06 & 0.06 & 0.06 \end{pmatrix}.$$

All entries are positive, because A defines the oblique norm in the quadrant $\mathbb{R}^2_{\geqq}$. The rows of A define the facets of the approximation from the right to the left.



(a) 10 cones

(b) 40 cones

Figure 6: Approximation of (7)

4.2 Nonconvex Example

We present an $\mathbb{R}^2_{\geqq}$-nonconvex example taken from Zhang (1999):

$$\begin{aligned} \min \quad & 10(x_1-2)^4 + 10(x_1-2)^3 + 10(x_2-2)^4 + 10(x_2-2)^3 + 10 \\ \min \quad & (x_1-3)^2 + (x_2-3)^2 + 10 \\ \text{s.t.} \quad & -x_1 - x_2 + 0.1 \leq 0 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \\ & x \in \mathbb{R}^2. \end{aligned} \tag{8}$$

Two interesting properties of our approximation algorithm can be seen in Figure 7, depicting the approximation for different numbers of cones.

The approximation first constructed by the algorithm is similar to the convex hull of the nondominated set. Even when using 30 cones for the approximation it is not yet apparent that the problem is $\mathbb{R}^2_{\geqq}$-nonconvex. The reason is that the algorithm uses only the norm method as long as it finds candidates with a deviation larger than ϵ (which was set to 0.0001 in this example). When it does not find such a candidate in a cone, it switches to the Tchebycheff method to examine the interior of the approximation and "discovers" the nonconvexity in the big cone, see Figures 7(c) and 7(d). This illustrates that the choice of ϵ can influence the approximation process.

As in the $\mathbb{R}^2_{\geqq}$-convex example above, we see that areas with a big curvature induce numerous cones so that the linear approximation adapts to the nonlinear nondominated set. Our results agree with those obtained by Zhang (1999) using the Tchebycheff scalarization.

4.3 Case Study: Evaluation of Aircraft Technologies

We now present a bicriteria model to evaluate aircraft technologies for a new aircraft. The model was proposed in Mavris and Kirby (1999) and the data was provided by the Aerospace Systems Design Laboratory at Georgia Institute of Technology. They can be found in Schandl (1999).

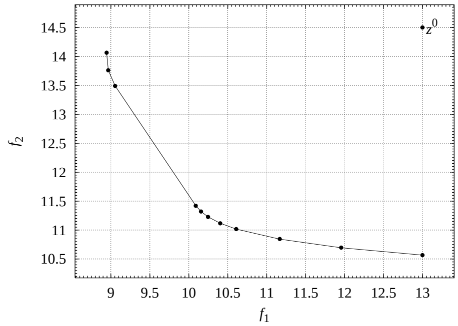
The model is a bicriteria problem of the form

$$\begin{aligned} \min \quad & f_1(x) \\ \min \quad & -f_2(x) \\ \text{s.t.} \quad & -1 \leq x_i \leq 1 \qquad i = 1, \ldots, 9. \end{aligned} \tag{9}$$

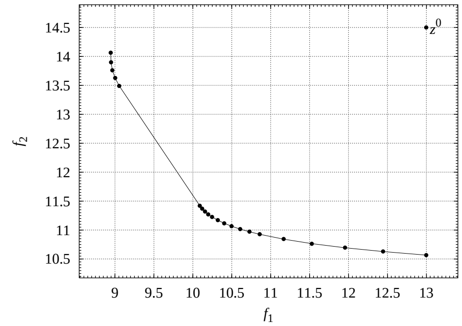
The functions $f_1(x)$ and $f_2(x)$ are modeled as Response Surface Equations:

$$b_0 + \sum_{i=1}^{9} b_i x_i + \sum_{i=1}^{9} b_{ii} x_i^2 + \sum_{i=1}^{8} \sum_{j=i+1}^{9} b_{ij} x_i x_j,$$

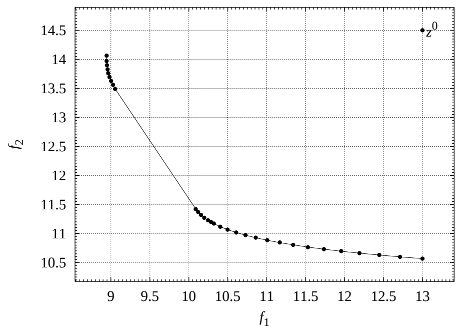
where the coefficients b_i and b_{ij} are found by regression. The Hessian of



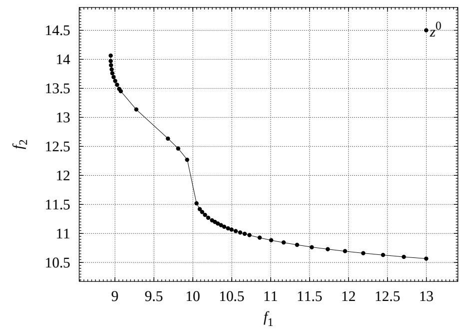
(a) 10 cones



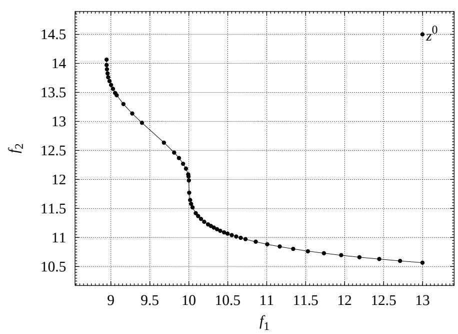
(b) 20 cones



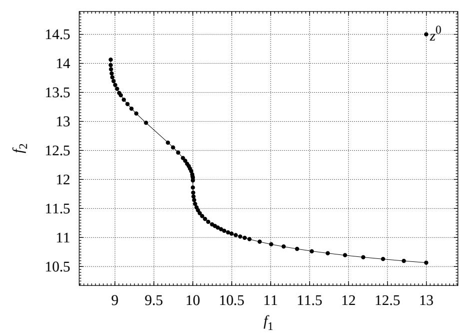
(c) 30 cones



(d) 40 cones



(e) 50 cones



(f) 60 cones

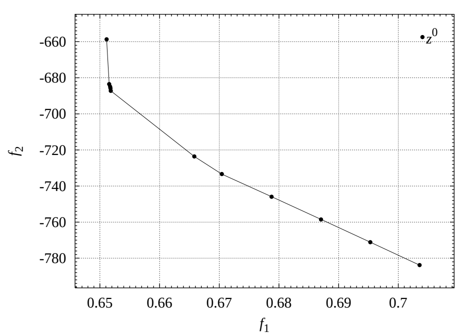
Figure 7: Approximation of (8)

neither of the functions $f_1(x)$ and $f_2(x)$ is positive or negative (semi)definite.

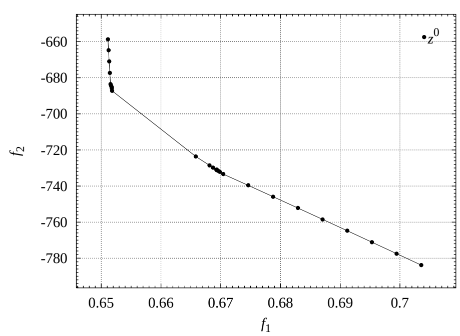
The decision variable in the problem is a vector of nine so-called "k" factors. The impact of a technology is mapped to such a vector, so every technology has a specific vector assigned to it. Not all technologies affect all components of the vector. While the problem is thus discrete, the goal of this model is to identify the values of "k" factors that are beneficial for the objective functions. Then technologies with corresponding vectors can be further investigated. All "k" factors are normalized to the range $[-1, 1]$ and represent a change from the value of the currently used technologies.

The two criteria are the life cycle cost (including research cost, production cost, and support cost) to be minimized and the specific express power (measure of maneuverability) to be maximized.

The results of the approximation algorithm for 10 and 29 cones are shown in Figure 8. Our approximation agrees with the simulation results obtained at the Aerospace Systems Design Laboratory, see Schandl (1999).



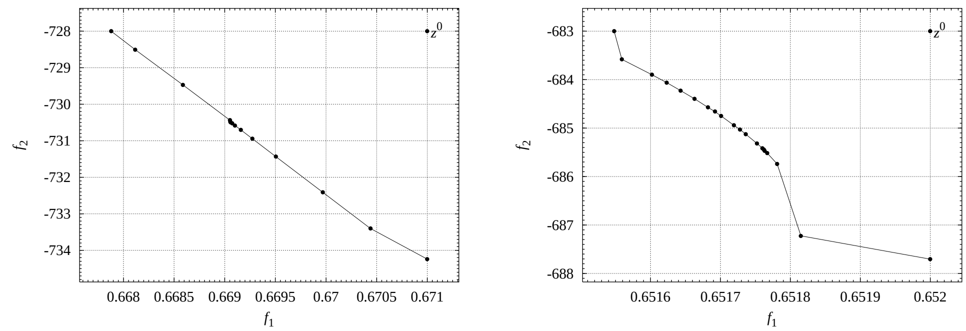
(a) 10 cones



(b) 29 cones

Figure 8: Approximation of (9)

There are two areas with an accumulation of constructed points in Figure 8(b). We examine these areas more closely by manually setting the reference point to $(0.671, -728)$ and $(0.652, -683)$, respectively. The corresponding approximations are shown in Figures 9(a) and 9(b). In Figure 9(a), no reason for the accumulation of constructed points is apparent. Figure 9(b) on the other hand shows a small nonconvex area of the nondominated set.



(a) 13 cones, $z^0 = (0.671, -728)$ (b) 19 cones, $z^0 = (0.652, -683)$

Figure 9: Approximation of (9)

Being able to choose the reference point in this way demonstrates a strength of our approximation approach. By simply resetting this point, we are able to closely examine "suspicious" areas or areas of special interest. Thus the approximation approach can be used both to get a general impression of the entire nondominated set and to "zoom into" areas of interest without changing the underlying algorithm.

An extended model includes a constraint and is discussed in Schandl (1999).

4.4 Case Study: Choosing Affordable Projects

We now consider the problem of selecting the most affordable portfolio of projects so that two criteria are maximized subject to a budgetary constraint. The model and data were taken from Adams et al. (1998) and Hartman (1999).

There are 24 projects in which the decision maker can invest. Depending on the model, the decision maker can invest in each project exactly once (binary variables) or a positive number of times (integer variables). The goal is to maximize the net present value (NPV) of investment and to maximize the joint application or dual use (JA/DU) potential of the chosen projects. The latter is a score assigned to each project by an expert. The investment has to be made with respect to a budgetary constraint. The problem is formulated as a bicriteria knapsack problem:

$$\begin{aligned} \max \quad & \sum_{i=1}^{24} c_{1i} x_i \\ \max \quad & \sum_{i=1}^{24} c_{2i} x_i \\ \text{s.t.} \quad & \sum_{i=1}^{24} a_i x_i \leq b \\ & x \text{ binary or nonnegative integer,} \end{aligned} \tag{10}$$

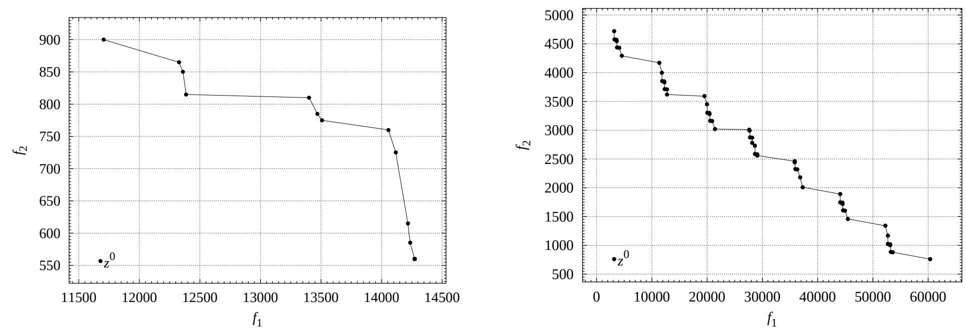
where the parameters are explained in Table 1. The values of the parameters are given in Schandl (1999).

Parameter	Explanation
c_{1i}	NPV of investment for project i in millions of dollars, $i = 1, \ldots, 24$
c_{2i}	JA/DU score for project i, $i = 1, \ldots, 24$
a_i	Total cost of project i over three years in hundreds of thousands dollars, $i = 1, \ldots, 24$
b	Total budget in hundreds of thousands of dollars

Table 1: Explanation of parameters in (10)

The approximation for the binary variable x is shown in Figure 10(a). Our approximation algorithm finds all twelve nondominated solutions given in Hartman (1999).

Allowing the variable x to be a nonnegative integer instead of binary yields many more solutions. The approximation of (10) for the nonnegative integer variable x is shown in Figure 10(b). In fact, the approximation finds all 54 nondominated solutions that, according to personal communications, Hartman found using her implementation of a dynamic-programming-based algorithm generating all nondominated points.



(a) Binary variables

(b) Integer variables

Figure 10: Approximation of (10)

5 Conclusions

In this paper we introduced a new approximation approach for bicriteria programs. Block norms are used to construct the approximation and evaluate its quality.

The algorithm combines several desirable properties. Whenever possible, the approximation is improved in the area where "it is needed most" because in each iteration, a point of worst approximation is added. The algorithm is applicable even if the structure and convexity of the feasible set is unknown. Given this knowledge though, more efficient versions can be applied. Using the approximation or a norm induced by it to improve the approximation releases the decision maker from specifying preferences (in the form of weights, norms, or directions) to evaluate the quality of the approximation.

The algorithm yields a global piecewise linear approximation of the nondominated set which can easily be visualized. For $\mathbb{R}^2_{\geqq}$-convex problems, a closed-form description of the approximation can be calculated. For all problems, the trade-off information provided by the approximation can be used in the decision-making process. While the approximation is carried out objectively, the subjective preferences must be (and should be) applied to single out one (or several) final result(s).

In the future, we plan to employ global optimization techniques for the single objective subproblems in order to handle problems with disconnected nondominated set and/or local minima.

References

Adams, Warren; Coffin, Marie; Kiessler, Peter; Kulasekera, K. B.; Saltzman, Matthew J.; Wiecek, Margaret (1998). An Optimization-Based Framework for Affordable Science and Technology Decisions at the Office of Naval Research. Clemson Report, October 16, 1998. Available at `http://www.math.clemson.edu/affordability/reports.html` (15.12.1999).

Bitran, G. R.; Magnanti, T. L. (1979). The Structure of Admissible Points with Respect to Cone Dominance. *Journal of Optimization Theory and Applications*, 29:573–614.

Chen, Wei; Wiecek, Margaret M.; Zhang, Jinhuan (1999). Quality Utility—A Compromise Programming Approach to Robust Design. *ASME Journal of Mechanical Design*, 121:179–187.

Cohon, Jared L. (1978). *Multiobjective Programming and Planning.* Academic Press, New York.

Das, Indraneel (1999). An Improved Technique for Choosing Parameters for Pareto Surface Generation Using Normal-Boundary Intersection. In *Short Paper Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization*, vol. 2, pages 411–413.

Fruhwirth, B.; Burkard, R. E.; Rote, G. (1989). Approximation of convex curves with application to the bicriterial minimum cost flow problem. *European Journal of Operational Research*, 42:326–338.

Hamacher, Horst W.; Klamroth, Kathrin (1997). Planar Location Problems with Barriers under Polyhedral Gauges. Report in Wirtschaftsmathematik 21, Universität Kaiserslautern, Department of Mathematics. To be published in *Annals of Operations Research*.

Hartman, Rebecca L. (1999). *Implementation of Multiple Criteria Dynamic Programming Procedures for Affordable Decision Making*. Master's thesis, Clemson University, Clemson, SC. Available at `http://www.math.clemson.edu/affordability/publications.html` (15.12.1999).

Jahn, J.; Merkel, A. (1992). Reference Point Approximation Method for the Solution of Bicriterial Nonlinear Optimization Problems. *Journal of Optimization Theory and Applications*, 74:87–103.

Li, Yusheng (1999). *Approximating the Pareto Set of Convex Bi-Criteria Optimization Problems to Aid Decision Making in Design*. Ph.D. thesis, Clemson University, Clemson, SC.

Li, Yusheng; Fadel, Georges M.; Wiecek, Margaret M. (1998). Approximating Pareto Curves Using the Hyper-Ellipse. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Symposium, Part 3*, pages 1990–2002. Paper No. 98-4961.

Luc, Dinh The (1989). *Theory of Vector Optimization*. No. 319 in Lecture Notes in Economics and Mathematical Systems. Springer Verlag, Berlin.

Mavris, Dimitri N.; Kirby, Michelle R. (1999). Technology Identification, Evaluation, and Selection for Commercial Transport Aircraft. Presented at the 58^{th} Annual Conference of the Society of Allied Weight Engineers in May 1999 in San Jose, CA. Available at `http://alpha.cad.gatech.edu/pubs/` (15.12.1999).

Murthy, Ishwar; Her, Shenq-Shyong (1992). Solving Min-Max Shortest-Path Problems on a Network. *Naval Research Logistics*, 39:669–683.

Pascoletti, A.; Serafini, P. (1984). Scalarizing Vector Optimization Problems. *Journal of Optimization Theory and Applications*, 42:499–524.

Payne, Anthony N. (1993). Efficient Approximate Representation of Biobjective Tradeoff Sets. *Journal of the Franklin Institute*, 330(6):1219–1233.

Poliščuk, L. I. (1979). Piecewise-linear approximation of the Pareto boundary of convex two-criteria problems. In *Models and Methods of Investigating Economic Systems*, pages 108–116. "Nauka" Sibirsk. Otdel., Novosibirsk. Russian.

Sawaragi, Yoshikazu; Nakayama, Hirotaka; Tanino, Tetsuzo (1985). *Theory of Multiobjective Optimization*, vol. 176 of *Mathematics in Science and Engineering*. Academic Press, Inc., Orlando, FL.

Schandl, Bernd (1999). *Norm-Based Evaluation and Approximation in Multicriteria Programming*. Ph.D. thesis, Clemson University, Clemson, SC. Available at `http://www.math.clemson.edu/affordability/publications.html` (15.12.1999).

Steuer, Ralph E.; Choo, E. U. (1983). An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming. *Mathematical Programming*, 26:326–344.

Warburton, Arthur (1987). Approximation of Pareto Optima in Multiple-Objective, Shortest-Path Problems. *Operations Research*, 35:70–79.

Yang, X. Q.; Goh, C. J. (1997). A method for convex curve approximation. *European Journal of Operational Research*, 97:205–212.

Zhang, Jinhuan (1999). *An Interactive Multiobjective Robust Design Procedure*. Master's thesis, Clemson University, Clemson, SC.

Zhang, Jinhuan; Wiecek, Margaret M.; Chen, Wei (1999). Local Approximation of the Efficient Frontier in Robust Design. To appear in *ASME Journal of Mechanical Design*.