8-2009

Quality Representation in Multiobjective Programming

Stacey Faulkenberg
Clemson University, sfaulke@clemson.edu

Quality Representation in Multiobjective Programming

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mathematical Sciences

by
Stacey L. Faulkenberg
August 2009

Accepted by:
Dr. Margaret M. Wiecek, Committee Chair
Dr. Perino M. Dearing
Dr. Georges M. Fadel
Dr. Matthew J. Saltzman

Abstract

In recent years, emphasis has been placed on generating quality representations of the nondominated set of multiobjective programming problems. This manuscript presents two methods for generating discrete representations with equidistant points for multiobjective programs with solution sets determined by convex cones. The Bilevel Controlled Spacing (BCS) method has a bilevel structure with the lower-level generating the nondominated points and the upper-level controlling the spacing. The Constraint Controlled Spacing (CCS) method is based on the epsilon-constraint method with an additional constraint to control the spacing of generated points. Both methods (under certain assumptions) are proven to produce (weakly) nondominated points. Along the way, several interesting results about obtuse, simplicial cones are also proved.

Both the BCS and CCS methods are tested and show promise on a variety of problems: linear, convex, nonconvex (CCS only), two-dimensional, and three-dimensional. Sample Matlab code for two of these examples can be found in the appendices as well as tables containing the generated solution points. The manuscript closes with conclusions and ideas for further research in this field.

Acknowledgments

This manuscript would not have been possible without the contributions of numerous individuals. I would first like to thank my advisor Dr. Margaret Wiecek for her support and encouragement, and for knowing when I needed a little (or maybe, big) push. In rough times, she was often an "advisor" in the broadest sense of the word, and I will always remember and appreciate that. I would also like to thank my other committee members, Dr. Perino Dearing, Dr. Georges Fadel, and Dr. Matthew Saltzman, for taking the time to read my work and offering valuable insight.

Next, I would like to thank all of my wonderful friends without whom I never would have made it through this sometimes arduous journey. Thank you for listening to me cry and for convincing me not to give up. You know who you are, and I'm glad we're friends.

Lastly, although really they probably deserved to be thanked first, is my family. Without them, I would be nothing. My parents are the most loving, giving, selfless people I have ever known, and I owe them lifetimes of gratitude for all that they have done for me. Mom, Dad, this dissertation is dedicated to you. And I can't forget my little brother who I'm pleased to say is now also one of my best friends. Thanks, Brad, la la la. The final member of my family (and I will fight anyone who says she's not) who I would like to thank is Patches. My sweet little bean, snuggle buddy, stress reliever, constant companion: sometimes you were the only thing that could make me smile. And because of that, you will always be more than a pet to me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Decisions are a part of life and can range from the mundane ("What should I have for breakfast?") to the life-changing ("Who should I spend the rest of my life with?"). With so many different options and opinions bombarding us on a daily basis, it is easy to become overwhelmed. Herein lies the attraction of mathematical programming: its ability to be an unbiased aid in the decision making process.

An important subfield of mathematical programming which in the past decade has "found its legs" (so to speak) in the research community is multiple-objective optimization. The increased research efforts in this field are due in large part to the ubiquity of multiple-objective problems in the real world. Multiple-objective problems can be found in a wide array of applications: in [18], the authors formulate an multiple-objective program (MOP) to determine optimal routes for the transportation of nuclear waste; the authors of [62] use multiple-objective optimization to assist in developing a well managed paper recycling logistics system; and in [59], the authors apply a multiple-objective framework to the problem of allocating the bandwidth of a computer server in both a fair and an efficient manner. Of course, these are only a few examples of multiple-objective optimization "in action" and many others can be found. Hopefully, however, these applications give a sense of the widespread applicability and usefulness of the field.

Unlike single-objective optimization problems which typically have one unique optimum, anmul MOP usually has infinitely many solution points due to conflict among the objective functions. Mathematically, all of the solutions are equivalent, so the selection of the final “best” solution depends upon the preferences and experiences of the decision maker (DM). Techniques for assisting the DM in the actual choice of this final solution are studied in the field of multiple criteria decision analysis (MCDA). Together, multiple criteria optimization and MCDA form the field of multiple criteria decision making (MCDM). The role of multiple-objective programming, then, when considered in the larger framework of MCDM, is to present the DM with as much information as possible about the solution set to facilitate the next step in the decision making process.

Unfortunately, this is a difficult task: the set of solution points of an MOP can rarely be defined by a closed-form formula and generating the entire set of solutions is almost always too time consuming and costly to be feasible. Thus, the majority of the research in this field has focused on generating representations or approximations of the solution sets; for MOPs with continuous variables, many methods are proposed (see [28], [66] for reviews). In the literature, the terms “representation” and “approximation” seem to be used almost interchangeably. However, in this manuscript, we differentiate between (discrete) representations and approximations of the solution set of an MOP as suggested in [36] and [72]. Henceforth, when we speak of a discrete representation, or simply a representation, we mean a finite subset of individual (true) solution points; when we use the term approximation, we are referring to a collection of (perhaps, not exact) solution points *along with* some sort of approximating structure (e.g., a piece-wise linear or interpolated curve).

It is our opinion that discrete representations of solution sets are preferable to approximated solution sets for three reasons. First, discrete representations present a finite, manageable number of solutions to the DM, whereas approximations do not limit the number of solutions. Second, discrete representations explicitly provide the DM with solution points while solutions are only implicitly available through an approximating structure. Lastly, all the solutions points in a discrete representation are optimal for the MOP; this is not necessarily true for solution points inferred from an approximated solution set.

With respect to discrete representations, several authors (e.g., [2, 10, 33]) suggest that more emphasis should be placed on finding globally representative subsets of the solution sets of MOPs, instead of contenting ourselves with simply finding solutions. That is, solution procedures should be placed into a larger framework where both the relationship among solution points and the relationship between the approximated solution set and the true solution set are considered. Further, Lotov et al. [49] and Berezkin et al. [11] discuss the importance of the visualization of solution sets as a tool in the decision making process. Generating representative subsets of the solution sets of MOPs both provides complete information to the DM as well as aids in her visualization of the solution set (in two and three dimensions).

In this light, many researchers have proposed measures for determining the quality of solution sets of MOPs. Others have begun integrating those and other measures into algorithms for generating discrete representations which meet a prespecified quality criterion. In Chapter 3, we review and classify the measures and (exact) algorithms that have been published in the literature. We present measures that have been proposed independently of an algorithm or that are used in conjunction with a heuristic algorithm but are applicable to exact algorithms. Sayin [67] suggests that measures of the quality of a solution set fall into three main categories: cardinality, coverage, and spacing. The measures are sorted according to this scheme. The published algorithms are sorted according to whether a measure is integrated before generation of a solution point (*a priori*), after generation of a solution point (*a posteriori*), or not at all. We include the last category because we found several algorithms that were developed to produce quality representations of the solution set but do not integrate any measure per se. Again, note that we include only exact algorithms which produce discrete representations satisfying a stated quality measure or which improve on a certain quality characteristic of a previous method. Additionally, because of this, we do not include any measure that quantifies the “error” of a discrete representation (the distance between the representation and the true solution set) because in the case of exact algorithms, the representation is always be a subset of the true solution set (i.e., there is no error).

Based on our literature review, the goal of our research is threefold. First, we would like to generate discrete representations of the Pareto set with equidistant spacing of Pareto points and complete coverage. Second, we would like the method to be applicable to both convex and nonconvex continuous MOPs. Third, we would like to be able to generalize the method to notions of optimality defined by convex, polyhedral cones. This final aspect of our research has been studied recently both theoretically in [39], [31], and [80], and in applications in [15] and [78], as a means of integrating the DM's preferences into the optimization process. Of the *a priori* techniques we reviewed, only [30] presents a method for controlling the spacing of Pareto points. Only [30] and [50] are applicable to general MOPs. Lastly, only [30] presents an approach that can be extended to notions of optimality defined by general cones. Our work offers alternatives to the method presented in [30] which we believe will be more easily understood and implemented by operations research practitioners. Additionally, we introduce the idea of controlling the tradeoffs of generated solution points, in order to further integrate the DM into the optimization process.

Finally, we would like to emphasize that in our work, we seek to produce quality representations of the solution set in the *objective space*. This decision is supported by discussions in several papers such as [10], [65], and [67]. In particular, all of these papers give three main reasons for focusing on the objective space instead of the decision space. First, the dimension of the objective space (i.e., the number of objective functions) tends to be smaller than the dimension of the decision space (i.e., the number of decision variables) making the objective space much more attractive computationally in terms of size and complexity. Second, there is no guarantee of a one-to-one mapping between the decision space and the objective space. Thus, if we generate a representation in the decision space, it is possible that all of the decision points could map to the same solution in the outcome space. Third, DMs seem to base their decisions primarily on the values of the objective functions, not on the values of decision variables.

The organization of this manuscript is as follows. In Chapter 2, we present terminology and notation that we use throughout the remainder of the dissertation. As discussed above, Chapter 3 is a literature review of measures of the quality of the solution set and algorithms that have been proposed to generate quality representations. In Chapters 4 and 5, we give our two methods for generating equidistant representations. The Bilevel Controlled Spacing method is discussed in Chapter 4, and the Constraint Controlled Spacing method is discussed in Chapter 5. Chapter 6 contains our numerical experiments for both methods. Our conclusions and directions for further research are presented in Chapter 7. Appendices A and B contain sample Matlab code from the numerical experiments, and Appendix C contains the solution points for selected problems from Chapter 6.

Chapter 2

Terminology and Notation

The general form of a multiple-objective program (MOP) is as follows:

$$\begin{aligned} &\text{minimize} && f(x) = [f_1(x),\ f_2(x), \ldots, f_p(x)] \\ &\text{subject to} && x \in X \end{aligned} \tag{2.1}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, p$, and $X \subseteq \mathbb{R}^n$. We assume that f_i, $i = 1, \ldots, p$, are continuous, and we let $Y = f(X) = \{f(x) : x \in X\}$ be the feasible region in the objective space.

When comparing two vectors $a, b \in \mathbb{R}^p$, we use the following notations: $a \leqq b$ implies that $a_i \leq b_i$ for $i = 1, \ldots, p$, while $a \leq b$ implies that $a_i \leq b_i$ for $i = 1, \ldots, p$ and $a \neq b$.

The optimality of a solution $y \in Y$ to (2.1) can be defined by a pointed, convex, polyhedral cone C which we call a *preference cone*.

Definition 2.0.1. A set $C \subset \mathbb{R}^p$ is said to be a *cone* if

$$\lambda C \subseteq C \text{ for all } \lambda > 0.$$

Definition 2.0.2. A cone $C \subset \mathbb{R}^p$ is said to be *convex* if

$$C + C \subseteq C.$$

Definition 2.0.3. A convex cone $C \subset \mathbb{R}^p$ is said to be *pointed* if

$$C \cap -C = \{0\}$$

Definition 2.0.4. A convex cone $C \subset \mathbb{R}^p$ is said to be *polyhedral* if

$$C = \{y \in \mathbb{R}^p : Ay \geqq 0\}$$

where A is a $m \times p$ matrix.

Additionally, we assume that C is simplicial.

Definition 2.0.5. [57] Let $C = \{y \in \mathbb{R}^p : Ay \geqq 0\}$ be a cone in $\mathbb{R}^p$. Then, C is *simplicial* if A is a $p \times p$ invertible matrix.

Note that a simplicial cone is always a polyhedral cone.

More details on the types of cones which can be used to define optimality in multiple-objective programming can be found in [27], among others.

Definition 2.0.6. Given a preference cone C, we say a solution $y^* = f(x^*)$ to (2.1) is *nondominated* if $(y^* - C) \cap Y = \{y^*\}$ or is *weakly nondominated* if $(y^* - \text{int } C) \cap Y = \emptyset$.

The set of all (weakly) nondominated points is called the *(weakly) nondominated set* and is denoted by $N(Y, C)$ $(N_W(Y, C))$. The pre-image of $N(Y, C)$ $(N_W(Y, C))$, in the decision space, is known as the *(weakly) efficient set*, $E(X, f, C)$ $(E_W(X, f, C))$.

Often, preferences are modeled by the *Pareto cone* $\mathbb{R}^p_{\geqq} := \{y \in \mathbb{R}^p : y_i \geq 0, i = 1, \ldots, p\}$.

Definition 2.0.7. A point $y^* = f(x^*)$ is called a *Pareto optimal* point if there does not exist $x \in X$ such that $f(x) \leq f(x^*)$. Further, a solution is called a *weak Pareto optimal* point if there does not exist $x \in X$ such that $f(x) < f(x^*)$.

The set of all (weak) Pareto optimal points is called the *(weak) Pareto set* and is denoted by Y_N (Y_{WN}). In this case, we denote the (weakly) efficient set by X_E (X_{WE}). In our work, we assume that $N_W(Y, C)$ and Y_{WN} are nonempty, bounded sets.

The relationship between Pareto sets and general nondominated sets given a cone C has been investigated by [82] and [58], among others. This relationship is given in the following proposition.

Proposition 2.0.8. *Let C be a convex, pointed cone such that $C = \{y \in \mathbb{R}^p : Ay \geqq 0\}$ where A is a $m \times p$ matrix. Further, let $A \cdot Y = \{Ay : y \in Y\}$. Then,*

(i) $y \in N(Y, C)$ if and only if $Ay \in N(A \cdot Y, \mathbb{R}^p_{\geqq})$ [82], and

(ii) $E(X, f, C) = E(X, Af, \mathbb{R}^p_{\geqq})$ [58].

In other words, to obtain the nondominated set $N(Y, C)$ for the MOP in (2.1), we may equivalently determine the Pareto set of the transformed MOP in (2.2) and then use the elements in the efficient set of (2.2) to recover $N(Y, C)$.

$$\begin{array}{ll} \text{minimize} & g(x) = [g_1(x),\ g_2(x), \ldots, g_p(x)] \\ \text{where} & g(x) = A \cdot f(x) \\ \text{subject to} & x \in X \end{array} \tag{2.2}$$

This result simplifies the problem of finding the nondominated set with respect to a cone C and is important later in our work.

In Chapter 5, the structure of the matrix A which defines a preference cone C is also important.

Definition 2.0.9. Given a $p \times p$ matrix A, a *minor* of A is obtained by deleting a specified (and equal) number of rows and columns of A.

Definition 2.0.10. Given a $p \times p$ matrix A, a *principal minor* of A is obtained by deleting the same rows and columns of A.

For example, given the matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

we obtain a minor of A by deleting row one and column three:

$$\begin{bmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}.$$

To obtain a principal minor, however, we delete the same row(s) and column(s), say row one and column one:

$$\begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix}.$$

An important class of square matrices is the class of positive definite matrices.

Definition 2.0.11. A $p \times p$ matrix A is said to be *positive definite* if

$$x^T Ax > 0 \text{ for all } x \in \mathbb{R}^p \text{ with } x \neq 0.$$

Definition 2.0.11 is difficult to apply in practice, so we present the following proposition.

Proposition 2.0.12. *[12, 63] A $p \times p$ matrix A is positive definite if all its principal minors are positive.*

Lastly, we define the notions of cone convexity and connectivity.

Definition 2.0.13. Let $C \subseteq \mathbb{R}^p$ be a cone. Then a set $Y \subseteq \mathbb{R}^p$ is said to be *C-convex* if $Y + C$ is convex, where $Y + C := \{y + c : y \in Y \text{ and } c \in C\}$.

With respect to connectivity, intuitively, the notion is simple: if Y_N is disconnected, we should be able to separate it into two or more disjoint parts.

Definition 2.0.14. [13] A set $S \subset \mathbb{R}^p$ is *disconnected* if there exist closed sets H_1 and H_2 such that

(i) $S \subseteq H_1 \cup H_2$,

(ii) $S \cap H_1 \neq \emptyset$,

(iii) $S \cap H_2 \neq \emptyset$, and

(iv) $S \cap (H_1 \cap H_2) = \emptyset$.

The application of this definition to the nondominated set is straightforward.

Chapter 3

Literature Review

In this chapter, we review and classify the measures and (exact) algorithms that have been published in the literature. In Section 3.1, we present measures that have been proposed independently of an algorithm or that are used in conjunction with a heuristic algorithm but are applicable to exact algorithms. We follow the suggestion of Sayin [67] and sort the measures of quality into three main categories: cardinality (Section 3.1.1), coverage (Section 3.1.2), and spacing (Section 3.1.3). Section 3.1.4 contains hybrid measures which overlap the three categories. In Section 3.2, the published algorithms are sorted according to whether a measure is integrated before generation (*a priori*) of a solution point (Section 3.2.1), after generation (*a posteriori*) of a solution point (Section 3.2.2), or not at all (Section 3.2.3). We include the last category because we found several algorithms that were developed to produce quality representations of the solution set but do not integrate any measure per se. Again, note that we include only exact algorithms which produce discrete representations satisfying a stated quality measure or which improve on a certain quality characteristic of a previous method. Additionally, because of this, we do not include any measure in Section 3.1 that quantifies the "error" of a discrete representation (the distance between the representation and the true solution set) because in the case of exact algorithms, the representation is always a subset of the true solution set (i.e., there is no error).

Throughout the paper, we denote discrete representations of the true efficient set and nondominated set by $\overline{X}_E$ and $\overline{Y}_N$, respectively. Moreover, for ease of notation, the same symbol may denote slighty different concepts when used in different contexts, but, in such cases, adequate explanation will be given. In particular, unless otherwise stated, $d(\cdot, \cdot)$ denotes a metric for measuring the distance between two points in $\mathbb{R}^n$; when not specifically defined, the DM should choose the appropriate metric for her situation. Lastly, when reviewing papers, we have maintained the authors' notations as much as possible.

3.1 Review of Measures

In general, we would like to provide the DM with a "good" representation of the nondominated set. The meaning of "good", here, is ambiguous because no definite consensus has been reached in the mathematical and operations research community on what qualities a good representation of the nondominated set should possess. However, within the past ten years, many authors have suggested quality measures which may be useful to this end. In this section, we present and classify the measures proposed in the literature.

We sort the measures into three main groups as suggested by Sayin [67]: measures of *cardinality*, *coverage*, and *spacing*. Cardinality refers to the number of points in a representation. In general, we desire enough points to fully represent the outcome set, but not so many that the DM is overwhelmed with choices. Measures of coverage seek to ensure that all regions of the outcome set are represented. That is, we do not want any portion of the outcome set to be neglected. Measures of spacing quantify the distance between points in the representation. Typically, we would like a representation to have uniform, or equidistant, spacing, so that all portions of the outcome set are represented to an equal degree. Although quality measures defined in the literature may allow for the use of alternate notions of optimality, they have predominantly been applied to problems governed by the Pareto notion. Further, in some papers, measures were defined for arbitrary sets. In these cases, we present the measures in the context of the Pareto set.

3.1.1 Measures of Cardinality

Measures of cardinality essentially boil down to the same idea: count the number of Pareto points in the representation. Clearly, two straightforward candidates for measures of cardinality are the size of the generated Pareto set (i.e., $|\overline{Y}_N|$) and the size of the generated efficient set (i.e., $|\overline{X}_E|$). Van Veldhuizen [79] proposes the former measure as "overall nondominated vector generation", while Sayin [67] proposes the latter.

On the other hand, in this group we also have the measure proposed by Wu and Azarm [81] called the "number of distinct choices". This measure is similar to the previous two but takes the DM's preferences into account: Pareto outcomes within a certain distance of each other (a choice made by the DM) are counted as a single point. Thus, the number of distinct choices of a discrete representation is always less than or equal to its cardinality. To calculate this measure, let μ $(0 < \mu < 1)$ be chosen so that the DM is indifferent between any two outcomes whose difference in each (normalized) criterion is less than or equal to μ. Next, divide the objective space into $\frac{1}{\mu^p}$ hypercubes. These hypercubes are indifference regions. Let $T_\mu(q)$ denote the hypercube with reference vertex q, and let $NT_\mu(q, \overline{Y}_N)$ be defined as follows:

$$NT_\mu(q, \overline{Y}_N) = \begin{cases} 1, & \exists\, y \in \overline{Y}_N \text{ such that } y \in T_\mu(q) \\ 0, & \text{else} \end{cases} .$$

Then, given the above, the number of distinct choices of a nondominated set, $\overline{Y}_N$, is

$$NDC_\mu(\overline{Y}_N) = \sum_{l_p=0}^{\frac{1}{\mu}-1} \cdots \sum_{l_2=0}^{\frac{1}{\mu}-1} \sum_{l_1=0}^{\frac{1}{\mu}-1} NT_\mu(q, \overline{Y}_N) \tag{3.1}$$

where $q = (q_1, \ldots, q_p)$ with $q_i = l_i \mu$. Thus, the number of distinct choices of a set $\overline{Y}_N$ is the number of hypercubes containing at least one Pareto point.

Since the cardinality of a discrete representation of the Pareto set is easily controlled, this category of measures is less important than the following two. In general, the cardinality of a representation should be minimized while still maintaining good coverage and spacing.

3.1.2 Measures of Coverage

Measures of coverage face the challenge of trying to assess something that is unknown because, in general, the true Pareto set is not known *a priori*. However, we must try to ensure that no region of Y_N is neglected. Because of this, we always seek to maximize the coverage of a discrete representation, and unless otherwise noted, the following measures should be maximized as well.

Czyżak and Jaszkiewicz [21] introduce two measures to determine the extent to which a discrete representation covers the true Pareto set. The measures, "D1" and "D2", are defined as follows:

$$D1(Y_N, \overline{Y}_N) = \frac{1}{|Y_N|} \sum_{y \in Y_N} \min_{\tilde{y} \in \overline{Y}_N} d(y, \tilde{y}), \tag{3.2}$$

and

$$D2(Y_N, \overline{Y}_N) = \max_{y \in Y_N} \min_{\tilde{y} \in \overline{Y}_N} d(y, \tilde{y}), \tag{3.3}$$

where $d(y, \tilde{y}) = \max_{i=1,\ldots,p}\{w_i|f_i(y) - f_i(\tilde{y})|\}$. The weights used in the definition of d are defined as $w_i = 1/R_i$ where R_i is the range of f_i in the true Pareto set. Measure D1 gives a (weighted) average of the distances between a point in the Pareto set and the closest point in the representation, while measure D2 gives the largest (weighted) distance between a point in the Pareto set and the closest point in the representation. We would like these distances to be as small as possible, so both of these measures should be minimized.

Zitzler and Thiele [86] propose a measure to determine the size of the region dominated by $\overline{Y}_N$. In his dissertation, Ziztler [84] calls this measure the "S-measure". Each point $y \in \overline{Y}_N$ dominates a (hyper)cube with one corner at y and another at $y^{\max}$ where $y^{\max} = (f_1^{\max}, \ldots, f_p^{\max})$ and $f_i^{\max} = \max_{x \in X} f_i(x)$. Thus, the region dominated by $\overline{Y}_N$, which we denote by $D(\overline{Y}_N)$, is found by taking the union of these cubes for all $y \in \overline{Y}_N$.

The value of the S-measure is the volume of this union.

$$S(\overline{Y}_N) = \text{volume}(D(\overline{Y}_N)). \tag{3.4}$$

Zitzler [84] also suggests a measure, called "M_3", to determine the overall range of the representation:

$$M_3(\overline{Y}_N) = \sqrt{\sum_{i=1}^{p} \max\{|\, y_i - \tilde{y}_i| : y, \tilde{y} \in \overline{Y}_N\}}. \tag{3.5}$$

This measure calculates an average of the ranges of the criteria.

Sayin [67] suggests the measure "coverage" which determines the maximum distance, ϵ, between a point in the true Pareto set and its closest neighbor in the representation:

$$\epsilon(Y_N, \overline{Y}_N) = \max_{y \in Y_N} \min_{\tilde{y} \in \overline{Y}_N} d(y, \tilde{y}). \tag{3.6}$$

Because we would like every point in the true Pareto set to be represented in our discrete representation, we want this measure to be minimized rather than maximized.

Wu and Azarm [81] propose two measures, "overall Pareto spread" and "i^{th} Pareto spread", which calculate the range of the entire representation and of each individual criterion, respectively. These measures are defined as follows:

$$OS(\overline{Y}_N) = \prod_i (\max_{y \in \overline{Y}_N} y_i - \min_{y \in \overline{Y}_N} y_i) \tag{3.7}$$

and

$$OS_i(\overline{Y}_N) = \max_{y \in \overline{Y}_N} y_i - \min_{y \in \overline{Y}_N} y_i. \tag{3.8}$$

Wu and Azarm [81] also propose a measure called "hyperarea difference" which is a slight variation on the S-measure (3.4) of Ziztler and Thiele. This measure calculates the difference (in terms of volume) between the portions of the objective space which are dominated by the true Pareto set and a given representation of the Pareto set. To overcome the difficulty of not knowing the true Pareto set, they suggest normalizing the objective space so that the volume of the region dominated by the true Pareto set can be estimated as one. Given this, the hyperarea difference is computed as follows:

$$HD(\overline{Y}_N) = 1 - \text{volume}(D(\overline{Y}_N)) \tag{3.9}$$

where $D(\overline{Y}_N)$ denotes the region of the objective space which is dominated by the set $\overline{Y}_N$, and can be found as discussed in the paragraph preceding (3.4). Note that if $\overline{Y}_N = Y_N$ then $\text{vol}(D(\overline{Y}_N)) = 1$ and thus, $HD(\overline{Y}_N) = 0$. In general, a small hyperarea difference value is desired.

Meng et al. [52] propose a measure of coverage called "extension". Let

$$U_i = \max_{x \in X_E} f_i(x)$$

and

$$L_i = \min_{x \in X_E} f_i(x),$$

then we have a set of reference outcomes, $\{y^1, \ldots, y^p\}$, where

$$y^i = (L_1, \ldots, L_{i-1}, U_i, L_{i+1}, \ldots, L_p).$$

Meng et al. suggest that U_i and L_i can be approximated if the true values are not readily available. Given a discrete representation $\overline{Y}_N$, we calculate the distance between each reference outcome and $\overline{Y}_N$ as follows:

$$d(y^i, \overline{Y}_N) = \min\{d(y^i, y) \,|\, y \in \overline{Y}_N\}.$$

Finally, the extension is calculated as follows:

$$EX(\overline{Y}_N) = \frac{\sqrt{\sum_{i=1}^{p} (d(y^i, \overline{Y}_N))^2}}{p}. \tag{3.10}$$

Thus, the extension measures an average distance of a representation from its reference outcomes. For this measure, a small value is preferred to a larger value because the latter could indicate that the representation is mainly in the center of the true Pareto set with the outskirts being neglected.

Zitzler et al. [85] suggest a measure of the "outer diameter" of a discrete representation which is defined as shown below:

$$OD(\overline{Y}_N) = \max_{i=1,\ldots,p} w_i(\max_{y \in \overline{Y}_N} y_i - \min_{y \in \overline{Y}_N} y_i), \tag{3.11}$$

where $w_i > 0$. The outer diameter measures the maximum (weighted) range over all the objective functions.

3.1.3 Measures of Spacing

Measures of spacing are abundant in the literature. In general, we desire a discrete representation of the Pareto set with equally spaced Pareto points, so that each region of the true Pareto set is represented to an equal degree. Note that having equidistant Pareto points, however, does not guarantee that we have good coverage as well, so measures of spacing should always be used in conjunction with a coverage measure.

Schott [71] proposes a measure for bicriteria problems called "spacing" which takes the standard deviation of the distances between nearest-neighbor points:

$$f_{spacing}(\overline{Y}_N) = \sqrt{\frac{1}{|\overline{Y}_N| - 1} \sum_{i=1}^{|\overline{Y}_N|} (\bar{d} - d_i)^2} \tag{3.12}$$

where each d_i is measured with the l_1-norm and $\bar{d}$ is the average of the d_i. Because we want the spacing of Pareto points to be equidistant, small values for this measure are desired. Note that Schott's measure can be extended to higher dimensions by changing the definition of d_i.

Ziztler [84] proposes the "M_2" measure which calculates the average cardinality of the set of points which are greater than a fixed distance, σ, from a Pareto point in the representation:

$$M_2(\overline{Y}_N) = \frac{1}{|\overline{Y}_N| - 1} \sum_{y \in \overline{Y}_N} |\{\tilde{y} \in \overline{Y}_N : d(y, \tilde{y}) > \sigma\}|. \tag{3.13}$$

This measure gives us a sense of the number of redundancies (with respect to the chosen value of σ) which are contained in our representation. Ideally, for a chosen σ, $M_2(\overline{Y}_N) = |\overline{Y}_N|$, indicating no redundancies.

Sayin [67] proposes the measure "uniformity" which is defined as the minimum distance, δ, between any two distinct points in the discrete representation of the Pareto set:

$$\delta(\overline{Y}_N) = \min_{y,\, \tilde{y} \in \overline{Y}_N,\, y \neq \tilde{y}} d(y, \tilde{y}). \tag{3.14}$$

Wu and Azarm [81] suggest a measure called "cluster" which measures the average size of a redundant cluster of points (with respect to the parameter μ) in the representation. To compute the cluster value, the number of points in the representation is divided by the number of distinct choices, $NDC_\mu(\overline{Y}_N)$ (see (3.1)): namely,

$$Cluster(\overline{Y}_N) = \frac{|\overline{Y}_N|}{NDC_\mu(\overline{Y}_N)}. \tag{3.15}$$

We desire no redundancies, so ideally, $|\overline{Y}_N| = NDC_\mu(\overline{Y}_N)$ which gives a cluster value of one. Otherwise, the cluster value is greater than one.

Messac and Mattson [55] present a measure of spacing called "evenness". For each point, y_i, in the discrete representation, two (hyper)spheres are constructed: the smallest and the largest spheres that can be formed between y_i and any other point in the set such that no other points are within the spheres. The diameters of the two spheres are denoted by d_l^i and d_u^i, respectively. The evenness, ξ, of a representation is then calculated with the following formula:

$$\xi(\overline{Y}_N) = \frac{\sigma_d}{\bar{d}} \tag{3.16}$$

where $\bar{d}$ and σ_d are, respectively, the mean and standard deviation of the set of minimum and maximum diameters for each point in the representation. A discrete representation with all points spaced equidistantly has $\xi = 0$ because the d_l^i and d_u^i are all equal (i.e., $\sigma_d = 0$).

Meng et al. [52] propose a measure called "uniformity" which was inspired by wavelet analysis. This measure was developed for comparing two different representations of Y_N. Let $\overline{Y}_N^1$ and $\overline{Y}_N^2$ be two distinct discrete representations of the Pareto set, and suppose that $|\overline{Y}_N^1| = N$ and $|\overline{Y}_N^2| = M$. Set $l = 1$. For each point $y^i \in \overline{Y}_N^1$ and $y^k \in \overline{Y}_N^2$, we calculate the distance to its nearest neighbor:

$$d_i^1 = d(y^i, \overline{Y}_N^1) = \min_{y^j \in \overline{Y}_N^1,\, y^j \neq y^i} d(y^i, y^j), \qquad i = 1, \ldots, N$$

and

$$d_k^2 = d(y^k, \overline{Y}_N^2) = \min_{y^j \in \overline{Y}_N^2,\, y^j \neq y^k} d(y^k, y^j), \qquad k = 1, \ldots, M.$$

Next we calculate the average distance between nearest neighbor points for both sets:

$$\bar{d}_l^1 = \frac{\sum_{i=1}^N d_i^1}{N} \qquad \text{and} \qquad \bar{d}_l^2 = \frac{\sum_{k=1}^M d_k^1}{M}.$$

Finally, we calculate the spacing measures as follows:

$$SP_l^1(\overline{Y}_N^1) = \sqrt{\frac{\sum_{i=1}^N (1 - F(d_i^1, \bar{d}_l^1))^2}{N-1}} \qquad \text{and} \qquad SP_l^2(\overline{Y}_N^2) = \sqrt{\frac{\sum_{k=1}^M (1 - F(d_k^2, \bar{d}_l^2))^2}{M-1}} \tag{3.17}$$

where

$$F(a,b) = \begin{cases} \frac{a}{b}, & \text{if } a > b \\ \frac{b}{a}, & \text{else} \end{cases} .$$

If $SP_l^1 < SP_l^2$, then $\overline{Y}_N^1$ has better uniformity, and vice versa. If $SP_l^1 = SP_l^2$ and $l \geq \min(N-1, M-1)$, then $\overline{Y}_N^1$ is the same as $\overline{Y}_N^2$. Else, if $SP_l^1 = SP_l^2$ and $l < \min(N - 1, M-1)$, then increment l by one and decrement N and M by one, and recalculate the spacing measure for both sets, ignoring the smallest d_i^1 and d_k^2, respectively. Note that this measure is binary because it is used to compare two different discrete representations of the Pareto set; the value of the spacing measure by itself does not have a clear interpretation.

Collette and Siarry [20] propose two different spacing measures for bicriteria problems: "spacing", which is a modification of Schott's measure (3.12), and the "hole relative size" measure. Both measures require that the generated Pareto points be put in ascending order with respect to the first objective function. Their spacing measure is computed as follows:

$$Spacing(\overline{Y}_N) = \sqrt{\frac{1}{|\overline{Y}_N| - 1} \sum_{i=1}^{|\overline{Y}_N|-1} \left(1 - \frac{d_i}{\bar{d}}\right)^2} \tag{3.18}$$

where $d_i = \sqrt{(f_1(x_i) - f_1(x_{i+1}))^2 + (f_2(x_i) - f_2(x_{i+1}))^2}$ and $\bar{d}$ is the average of all the d_i. Their hole relative size measure gives the ratio of the largest gap between two adjacent points to the average gap:

$$HRS(\overline{Y}_N) = \frac{\max_i d_i}{\bar{d}} \tag{3.19}$$

where d_i and $\bar{d}$ are as defined previously. The authors note that the hole relative size measure would not be appropriate for use on a problem with a disconnected Pareto set.

3.1.4 Hybrid Measures

Several authors propose measures which overlap the above three categories. Deb et al. [23, 26] suggest the "Δ" measure for bicriteria problems which takes into account both the spacing between generated Pareto points and the coverage of the true Pareto set by the generated representation. This measure calculates the distance between each point and its nearest neighbors (a spacing-type measure) as well as the distance between the individual objective minima and their respective single nearest neighbor (a coverage-type measure). Including the second part of the measure ensures that there is not a group of equally spaced points in the center of the set, for example, with the outer portions neglected. Deb's measure for bicriteria problems is as follows:

$$\Delta(\overline{Y}_N) = \frac{d_f + d_l + \sum_{i=1}^{|\overline{Y}_N|-1} |d_i - \bar{d}\,|}{d_f + d_l + (|\overline{Y}_N| - 1)\bar{d}} \tag{3.20}$$

where d_f and d_l are the Euclidean distances between the individual objective minima and the nearest points in the representation, the d_i are the Euclidean distances between each pair of consecutive Pareto points, and $\bar{d}$ is the average of all the d_i. A small value for Δ is desired with the ideal value being $\Delta = 0$: $d_i = \bar{d}$ for all i and $d_f = d_l = 0$, indicating that the individual objective minima are included in the representation. The authors note that this measure can be extended to three or more dimensions, but the formula would change slightly.

Leung and Wang [48] suggest the "U-measure" which measures both coverage and spacing, similar to Deb's measure (3.20). First, we determine the nearest neighbors of each Pareto point with respect to each axis, as well as the nearest neighbor of each reference point (i.e., the individual objective minima or other points chosen by the DM). Let χ be the set of distances between nearest neighbor outcomes and let $\bar{\chi}$ be the set of distances between a reference point and its nearest neighbor outcome. For good spacing, we would like the distances in χ to be roughly the same. For good coverage, we would like the distances in $\bar{\chi}$ to be close to zero. For ease of calculation, we combine the sets into one by computing the average of the distances in χ and incrementing each element in $\bar{\chi}$ by this number. We denote this new set by $\bar{\chi}'$. Now, we need only to check that the elements in $\bar{\chi}'$ are close to each other. Given this, we compute the U-measure as follows:

$$U(\overline{Y}_N) = \frac{1}{D}\sum_{i=1}^{D}\left|\frac{d_i'}{d_{ideal}} - 1\right| \tag{3.21}$$

where $d_i' \in \bar{\chi}'$, $d_{ideal} = \sum_{i=1}^{D} d_i'/D$, and $D = |\bar{\chi}'|$. If the points are equally spaced over the entire set, then $d_i' = d_{ideal}$ for each i resulting in $U = 0$. This measure calculates the average deviation from the ideal so that a small U-measure indicates a representation that is close to equidistant and covers the entire Pareto set.

Farhang-Mehr and Azarm [32] propose a measure called "entropy" using ideas from the field of information theory. Entropy assesses all three of the quality categories: cardinality, coverage, and spacing. For each Pareto point $\bar{y}^i$ in a discrete representation $\overline{Y}_N$, we define a scalar-valued influence function Ω_i which is decreasing in the distance to $\bar{y}^i$ (e.g., a Gaussian distribution centered on $\bar{y}^i$). Then, the density function, defined for any point in the objective space, is given by the following:

$$D(y) = \sum_{i=1}^{|\overline{Y}_N|} \Omega_i(y), \qquad y \in Y.$$

Next, we create an $a_1 \times a_2 \times \ldots \times a_p$ grid in the objective space so that the DM is indifferent between outcomes which share the same (hyper)cube (where a_i is the number of indifference regions with respect to the i^{th} axis). Let $y_{i_1,i_2,\ldots,i_p}$ denote the center point of the cube having grid position $(i_1, i_2, \ldots, i_p)$. Given this, we evaluate $D(y)$ for each center point and then normalize the result as follows:

$$\rho_{i_1,i_2,\ldots,i_p} = \frac{D(y_{i_1,i_2,\ldots,i_p})}{\sum_{k_1=1}^{a_1} \sum_{k_2=1}^{a_2} \cdots \sum_{k_p=1}^{a_p} D(y_{k_1,k_2,\ldots,k_p})}.$$

Finally, the entropy of $\overline{Y}_N$ is given by

$$H(\overline{Y}_N) = -\sum_{i_1=1}^{a_1} \sum_{i_2=1}^{a_2} \cdots \sum_{i_p=1}^{a_p} \rho_{i_1,i_2,\ldots,i_p} \ln(\rho_{i_1,i_2,\ldots,i_p}). \tag{3.22}$$

A high entropy value is desired because a set with high entropy maximizes coverage and minimizes redundancies for a given cardinality.

3.2 Review of Methods

In this section, we review articles which present exact methods for generating a discrete representation of the nondominated set. Recall that we use the term discrete representation to mean a subset of outcome points from the nondominated set, while an approximation uses some additional structure. We classify these papers according to whether a measure is incorporated into the method *a priori* (before generation of nondominated points), *a posteriori* (after generation of nondominated points), or not at all. For methods with measures, we also indicate which type of measure is used. Finally, within each category, we present the papers chronologically according to their publication dates.

3.2.1 Methods With *A Priori* Measures

Despite the prevalence of quality measures in the literature, only a few authors have integrated these measures into algorithms to produce representations of the Pareto set satisfying some prespecified quality criterion. Additionally, a majority of the algorithms in this section are only applicable to specific classes of problems.

Helbig [37] suggests an approach for producing a discrete representation of the Pareto set with good coverage which is applicable to biobjective programs (BOPs) with connected Pareto sets. The convex hull of the individual objective minima is discretized and these points are used as the reference points in the max-ordering method. Helbig presents a method for choosing the discretized points so that the maximum Euclidean distance between a point in the true Pareto set and a point in the representation is at most a prespecified value (chosen by the DM).

Churkina [19] investigates the Chebyshev method [74] as a method of producing a discrete representation of the Pareto set for convex MOPs. The notion of a delta-grid is used as a measure of coverage. A finite delta-grid, in terms of the Pareto set, is a finite subset in which the maximum distance between a point in the true Pareto set and a point in the representation is at most delta where the distance is measured with the Chebyshev norm. The author proves that for any chosen delta, it is possible to find an epsilon so that a finite epsilon-grid of reference points produces a delta-grid representation of the Pareto set. However, no method is presented for finding the value of epsilon or the set of reference points.

Sayin [68] proposes a method for multiple-objective linear problems (MOLPs) to produce a representation with a given target coverage value or the maximum coverage possible given a target cardinality when the set of efficient faces is known *a priori*. Here, the coverage of the representation is calculated using the coverage measure (3.6) from [67]. At each iteration, the point in the true Pareto set which has the maximum Chebyshev distance from the current representation is selected.

Sayin and Kouvelis [69] and Kouvelis and Sayin [46] give a two-stage method called Algorithm Robust for generating representations of the Pareto set for discrete BOPs. The coverage, using the measure (3.6) in [67], is controlled by continuing to refine an interval between two previously generated Pareto points until its length falls below a prespecified value.

Eichfelder [29, 29] is, to the best of our knowledge, the first author to attempt to control the spacing of generated nondominated points. Her method is based on the Pascoletti and Serafini scalarization [61], making it applicable to general MOPs and notions of optimality defined by general cones. She derives sensitivity information in a neighborhood about a nondominated point and uses this information to determine input parameters for the scalarization so that the produced nondominated point is a prespecified distance from the previous point. Although theoretically sound, issues arise when the method is applied to problems with three or more objective functions.

Ruzika [65] and Hamacher et al. [36] present two box algorithms for producing representations of the Pareto set for discrete BOPs. The algorithms use the lexicographical epsilon-constraint scalarization to generate Pareto points. Boxes are formed with consecutive Pareto points as the upper left and lower right corner points. The "accuracy" (a coverage measure) of the current representation is calculated as the area of the largest of these rectangles. The representation is refined until the accuracy has met a prespecified value. Alternatively, the authors point out that a specific cardinality may be used as the stopping criterion for the algorithms instead of a desired accuracy, and that the resulting accuracy is a function of this given cardinality. Filtering is also mentioned as a way to reduce redundancies and, thus, improve the spacing of the representation *a posteriori*.

Sylva and Crema [76] propose an algorithm for mixed-integer linear MOPs which generates representations of the Pareto set with good coverage. At each iteration, their algorithm finds the Pareto point which maximizes the infinity-norm distance from the set already dominated by previous outcomes. The cardinality of the representation can also be used as a stopping criterion.

Most recently, Masin and Bukchin [50] present the Diversity Maximization Approach to produce representations with good coverage for general MOPs. At each iteration, the most "diverse" outcome is added to the representation where the most diverse outcome is defined as the one that maximizes the minimum coordinate-wise distance between the new point and all the points already in the representation. The authors note that although this method is applicable to general MOPs, it is recommended predominantly for mixed-integer and combinatorial problems.

The methods discussed in this section which integrate some sort of quality measure of the Pareto set *a priori* have the disadvantage of perhaps increasing the complexity of the optimization procedure: not only do we want to produce a Pareto point, we want to produce a specific type of Pareto point. However, we feel that this possible drawback is outweighed by the fact that discrete representations produced by *a priori* methods wholly represent the true Pareto set, and thus provide the best possible information to the DM.

3.2.2 Methods With *A Posteriori* Measures

A posteriori methods are the simplest of the three classes of methods that we present here. In general, these methods consist of generating a discrete representation of the nondominated set and then removing certain points so that the resulting representation satisfies some quality criterion.

As early as 1980, filtering techniques were being proposed to produce discrete representative subsets of the outcome sets of MOPs. Steuer and Harris [75] suggest using a forward and reverse interactive filtering scheme to produce a representative subset of the

Pareto extreme points of an MOLP. Forward filtering consists of using a weighted l_p-norm to discard Pareto extreme points that are too close together and thus, produce a diverse set of extreme point outcomes. Once the DM has chosen her most preferred outcome from this set, the reverse filtering is performed. In this process, the closest Pareto extreme points to the preferred outcome are reintroduced and Pareto extreme points that are further away are discarded. In this way, the DM is now presented with outcomes that are most similar to her preferred outcome, allowing her to refine her outcome even further.

Morse [56] suggests a filtering method involving cluster analysis for reducing redundancy in the Pareto sets of MOLPs. The DM sets a minimum redundancy level below which she is indifferent between two Pareto points. The author experiments with several clustering methods among which he found the most useful in this context to be Ward's Method, the Group Average Method, and the Centroid Method. Using the desired method and the minimum redundancy level, clusters of Pareto points are formed. The DM is then presented with a representative Pareto point from each cluster.

More recently, Mattson et al. [51] suggest a Smart Pareto filter to produce representations with good cardinality and complete coverage which emphasize areas with high tradeoffs more than areas with low or insignificant tradeoffs. First, an "even" representation of the Pareto set is produced. The authors define an even representation as one where all areas of the Pareto set are represented to an equal degree and suggest methods proposed in [22], [54], or [55] to produce such a set. Next, a Pareto point is selected and the tradeoffs between it and all other Pareto points are calculated. If the tradeoff between the chosen point and another point falls below a prespecified level, the second point is removed. Otherwise, the second point is retained. This process is performed on each point in the representation. The authors denote the resulting representation as the "smart" Pareto set.

As mentioned previously, *a posteriori* methods tend to be simple to understand and straightforward to implement. However, one major drawback of this class of methods is that redundant work is performed. Resources are used to generate a large representation of the Pareto set only to have a portion of the produced Pareto points discarded at a later stage.

3.2.3 Methods Without Measures

This class includes a variety of methods which aim to produce quality discrete representations but which do not include any measure of quality of the Pareto set. Many authors improve upon the coverage or spacing of existing methods with simple variations. A few authors produce quality representations of other sets (i.e., weights, reference points) and project these sets onto the Pareto set. However, without a quality measure, the degree to which the quality of the method improves can be neither quantified nor guaranteed.

Steuer and Harris [75] propose an intra-set point generation method to go along with their filtering method which is discussed in the previous section. In cases where the Pareto extreme points do not sufficiently describe the Pareto set of a MOLP, the authors suggest generating Pareto points within the set (i.e., not extreme points) using intelligently chosen convex combinations of the Pareto extreme points so that the generated points provide good coverage of the entire Pareto set. They present empirical evidence which shows that choosing half of the weights from the uniform distribution and half of the weights from the Weibull distribution results in a well-distributed set.

Armann [2] develops a method for choosing the epsilon parameters in the hybrid weighted-sum, epsilon-constraint scalarization for general MOPs (proposed by Guddat et al. [35], among others). Given the desired number of points in the representation, he solves an integer program to determine the values of epsilon to use in the hybrid scalarization so that in the resulting representation, the distance between neighboring points is maximized. This improves the coverage and the spacing of the hybrid scalarization as compared to using equally spaced values of the epsilon parameter.

Benson and Sayin [10] propose a global shooting procedure to produce a representation of the nondominated set of a general MOP. This method seeks to cover the entire nondominated set without many redundancies. The method begins by constructing a simplex which contains the feasible region in the objective space. Then a subsimplex is chosen and a discrete sample of points from this subsimplex is taken. Finally, the representation is obtained by shooting from each of these points in a specific direction toward the nondominated set. The authors stress that it is the method by which the points are sampled from the subsimplex which determines whether the representation has good coverage.

Das and Dennis [22] introduce the Normal Boundary Intersection method for producing representations of the Pareto set with complete coverage. The convex hull of the individual objective minima is discretized with equally spaced points. Then a series of minimization problems is solved to determine the intersection between the boundary of the feasible region in the objective space and the normal vector emanating from each of these points respectively. The Normal Boundary Intersection method may produce non-Pareto points and not all Pareto points are obtainable using this method. However, if the Pareto set is sufficiently well-behaved the Normal Boundary Intersection method produces a representation with both good coverage and good uniformity.

Buchanan and Gardiner [16] perform a comparative study of two versions of the weighted Chebyshev method [74], one using the ideal point as a reference point and the other using the nadir point. The authors found that when choosing weights from the uniform distribution, discrete representations produced using the nadir outcome as the reference point had better coverage than those produced with the ideal point.

Messac et al. [53] introduce the Normal Constraint method for producing representations with good coverage. The convex hull of the individual objective minima is discretized with equally spaced points, and then for each discretized point, a single objective optimization problem is solved over a reduced feasible region which is determined using the current point; this produces a Pareto point. In [55], the authors point out that the original Normal Constraint method neglects certains regions of the Pareto set. They remedy this by slightly enlarging the convex hull of the individual objective minima so that it covers the entire feasible region. With this refinement, the Normal Constraint method produces representations with complete coverage of the Pareto set. Also, similar to the Normal Boundary Intersection method [22], for well-behaved Pareto sets, the Normal Constraint method produces representations with good uniformity as well because of the equally spaced points on the enlarged convex hull.

Fu and Diwekar [34] present a variation of the epsilon-constraint method for general MOPs. In their method, the parameter epsilon is chosen in a pseudo-random manner. Empirical evidence is given to show that representations produced using this technique have more complete coverage (measured in terms of the mean and variance of the set) than those produced by the traditional method of using uniformly spaced epsilon values.

Kim and de Weck [45] propose an algorithm based on the weighted-sum method which seeks to produce complete coverage. The Adaptive Weighted-Sum method begins by using the usual weighted-sum method to approximate the shape of the Pareto set. A piecewise linear mesh is formed using the Pareto points just found. Any mesh "patch" that is too large is refined with additional points chosen through interpolation. These points are then projected onto the Pareto set in the direction of a pseudo-nadir point. The authors note that non-Pareto points may be produced and a Pareto filter should be applied at the end of each iteration. The weighted-sum method can only produce nondominated points along $\mathbb{R}^p_{\geqq}$-convex regions of the Pareto set, resulting in large gaps in the representation if the nondominated set is not entirely convex. However, by refining the feasible region as discussed, the Adaptive Weighted-Sum method is able to generate Pareto points in nonconvex regions, thus improving the coverage of the weighted-sum method as well as making it applicable to nonconvex MOPs.

Zhang and Gao [83] present a method for adaptively choosing the weights and the reference point in the min-max method so as to produce a discrete representation of the Pareto set with approximately equidistant Pareto points. Beginning from a known Pareto point, a single objective optimization problem is solved to determine the optimal weights for the weighted-sum method which produces the specified point. The vector of weights is then used to determine the tangent descent direction at the current point. The weighting vector is translated along the tangential direction a distance of α, where α is the desired distance between Pareto points. This weighting vector now becomes the weighting vector of the min-max method and the reference point is chosen to be "small enough" on this vector (i.e., a point which clearly is outside the feasible region). The min-max problem is solved with this new weighting vector and reference point to obtain the next Pareto point. The method is an improvement over the Normal Boundary Intersection method [22] and the Normal Constraint method [53] because the reference plane (i.e., the tangent to each point) is updated at each point and more closely reflects the curvature of the Pareto set. However, this method still only produces approximately equidistant points because the α distance is measured between points on the tangent directions and not between the Pareto points themselves. This method is applicable to general MOPs, although it has only been tested on bicriteria problems.

Shao and Ehrgott [72] combine the global shooting procedure of Benson and Sayin [10] and the Normal Boundary Intersection method of Das and Dennis [22] to produce a revised Normal Boundary Intersection method for use on MOLPs. Instead of using the convex hull of the individual minima as the reference plane as in the Normal Boundary Intersection method, the revised method uses the subsimplex described in the global shooting procedure. This is done to overcome the inability of the Normal Boundary Intersection method to produce certain Pareto points. As in the original Normal Boundary Intersection method, equidistant reference points are chosen on the subsimplex and an optimization problem is solved for each reference point to determine the intersection between the boundary of the feasible region in the objective space and the normal vector emanating from that point. Shao and Ehrgott prove that the distance between any two Pareto points produced by the revised method is between d and $d\sqrt{p}$, where p is the number of objectives and d is the spacing between the reference points. Thus, for small problems, the new revised method is guaranteed to produce representations with complete coverage and good uniformity.

Karasakal and Köksalan [43] suggest a method for producing discrete representations of the Pareto set with good coverage and spacing as measured by (3.6) and (3.14), respectively. First, a weighted l_p-surface is used to approximate the Pareto set. Then, this surface is discretized with equidistant points which are projected onto the Pareto set in the direction of the gradient of the surface at each point. These Pareto points form the discrete representation. Although the method is proposed for general MOPs, the authors emphasize that it is best suited for convex problems.

The methods discussed in this section are a compromise between *a priori* methods and *a posteriori* methods. These methods eliminate the redundancy of the latter class and move toward the integrated quality of the former class. However, since no measure of quality of the Pareto points themselves is utilized, the quality of the produced representations usually cannot be guaranteed and is sometimes highly dependent on the structure (e.g., the curvature) of the Pareto set.

Chapter 4

Bilevel Controlled Spacing

In this chapter, we present the Bilevel Controlled Spacing (BCS) approach for generating discrete representations of the nondominated set. The chapter is divided into two main sections, Section 4.1 details the BCS approach in the context of biobjective problems and Section 4.2 extends the approach to multiple-objective problems. In the biobjective section, we first discuss the approach with respect to the Pareto cone and then generalize to polyhedral, convex cones. At the end of the biobjective section, we suggest a modification of the BCS approach that allows us to control the tradeoffs of the generated nondominated points. In the multiple-objective section, we present two different implementations of the BCS approach: the center method (Section 4.2.1) and the slicing method (Section 4.2.2). Again, we discuss both first with respect to the Pareto cone and then with respect to general cones.

4.1 Biobjective Approach

Pareto Cone The idea behind the Bilevel Controlled Spacing approach is to generate a Pareto point with the lower-level problem and to control the spacing of that point with the upper-level problem. Previously produced Pareto points are used as reference points for the placement of new points: we input two Pareto points and use a min-max formulation so that the generated point is equidistant between the two input points. Because of this, the DM is restricted somewhat in her choice for the cardinality of the representation: N must satisfy $N = 2^n + 1$ for some $n \in \mathbb{N}$. Further, we consider this approach only for convex MOPs since, as discussed below, we transform the bilevel optimization problem into a single-level problem using the Karush-Kuhn-Tucker (KKT) optimality conditions.

A similar bilevel formulation has been used in the context of MOLPs in [5, 6, 7, 8, 9]. In these papers, an additional linear function (i.e., not necessarily a criterion function of the MOLP) is optimized over the (weakly) efficient solutions of the problem. However, the purpose is not to generate a discrete representation as is our goal, but to to choose a single optimum from among the set of efficient points.

The Bilevel Controlled Spacing formulation (BCS(x^{1*}, x^{2*})) for a biobjective program is given below where $y^{1*} = f(x^{1*})$ and $y^{2*} = f(x^{2*})$ are previously produced Pareto points.

$$\begin{aligned} &\text{minimize} \quad \max\{||y^{1*} - f(x)||, ||y^{2*} - f(x)||\} \\ &\text{subject to} \quad x \in X_{WE} \end{aligned} \tag{4.1}$$

In the first iteration, the reference points are chosen to be

$$y^1 = f(x^1) := \operatorname{lex\,min}\{[f_1(x), f_2(x)] : x \in X\}, \tag{4.2}$$

and

$$y^2 = f(x^2) := \operatorname{lex\,min}\{[f_2(x), f_1(x)] : x \in X\}. \tag{4.3}$$

Recall from the discussion at the beginning of Section 3.1 that $x \in X_{WE}$ implies that there exists an ϵ such that x is a solution to the corresponding ϵ-constraint problem.

In other words, we can rewrite (4.1) in terms of the ϵ-constraint method as follows:

$$\begin{aligned}
\text{minimize} \quad & \max\{||y^{1*} - f(x)||, ||y^{2*} - f(x)||\} \\
\text{subject to} \quad & x = \arg\min \quad f_1(\tilde{x}) \\
& \qquad\quad \text{subject to} \;\; f_2(\tilde{x}) \leq \epsilon \\
& \qquad\qquad\qquad\quad\;\; \tilde{x} \in X
\end{aligned} \tag{4.4}$$

The first order KKT optimality conditions are both necessary and sufficient for convex programs (see for example, [4]). Thus, we may transform (4.4) into a single-level optimization problem by replacing the lower-level problem with its KKT conditions:

$$\begin{aligned}
\text{minimize} \quad & \max\{||y^{1*} - f(x)||, ||y^{2*} - f(x)||\} \\
\text{subject to} \quad & \nabla_x f_1(x) + u \nabla_x (f_2(x) - \epsilon) = 0 \\
& f_2(x) \leq \epsilon \\
& u(f_2(x) - \epsilon) = 0 \\
& u \geq 0 \\
& x \in X
\end{aligned} \tag{4.5}$$

where $u \in \mathbb{R}$ is a dual multiplier. In general, in the ϵ-constraint method, ϵ is fixed prior to optimization. However, note that fixing ϵ in (4.4) also fixes $f(x)$ so that no minimization actually occurs at the upper-level. Our approach is to allow ϵ to vary throughout the optimization, but not as an optimization variable. Rather, we optimize with respect to x, and ϵ is adjusted as needed to maintain feasibility.

General Cones The BCS method can be generalized to arbitrary convex, polyhedral preference cones. Just as in the CCS method, we use the results given in Proposition 2.0.8. Given that a cone C is represented by the matrix A, we make the transformation $g(x) = Af(x)$ and derive the KKT conditions for this new problem. The single-level optimization problem for a general cone C, BCS(x^{1*}, x^{2*}, A), is then given by

$$\begin{aligned} \text{minimize} \quad & \max\{||y^{1*} - f(x)||, ||y^{2*} - f(x)||\} \\ \text{subject to} \quad & \nabla_x g_1(x) + v \nabla_x (g_2(x) - \epsilon) = 0 \\ & g_2(x) \leq \epsilon \\ & v(g_2(x) - \epsilon) = 0 \\ & v \geq 0 \\ & x \in X \end{aligned} \tag{4.6}$$

where $v \in \mathbb{R}$ is a dual multiplier associated with the transformed functions. Notice that our objective function, however, is the same as in (4.5). This is because we seek equidistant spacing in the original objective space, not the transformed objective space.

Proposition 4.1.1. *Let C be a convex cone defined by $C = \{y \in \mathbb{R}^2 : Ay \geqq 0\}$. If $\hat{x}$ is an optimal solution to BCS(x^{1*}, x^{2*}, A), then $f(\hat{x}) \in N_W(Y, C)$.*

Proof. Let $\hat{x}$ be an optimal solution to BCS(x^{1*}, x^{2*}, A). Since the KKT optimality conditions are both necessary and sufficient for optimality and the ϵ-constraint method guarantees at least a weakly efficient point, $\hat{x} \in E_W(X, g, \mathbb{R}^p_{\geqq}) = E_W(X, Af, \mathbb{R}^p_{\geqq})$. By Proposition 2.0.8, $\hat{x} \in E_W(X, f, C)$ which implies $f(\hat{x}) \in N_W(Y, C)$. This completes the proof. □

The single level formulation in (4.6) can now be solved using a nonlinear solver. Unfortunately, we found through the literature (e.g., [73]) and through test runs of our own that the highly nonlinear complementary slackness conditions make this problem very difficult to solve. In [3] and [73], a branch-and-bound algorithm for solving bilevel problems is presented. The problem is first formulated as a single level problem as we have done and then solved without the complementary slackness conditions. If the complementary slackness conditions are satisfied for the relaxed problem, then we have found a solution. Otherwise, we branch on the most violated complementary slackness constraint, first setting the corresponding dual multiplier to zero then setting the primal constraint equal to zero. We solve each relaxation, checking to see if complementary slackness is satisfied. We continue in this way until we have found a solution. Note that usually we do not have to enumerate all the possible combinations because some paths lead to infeasibility or can be fathomed.

Pseudocode for the Bilevel Controlled Spacing Algorithm (BCSA_N) for general cones is shown in Figure 4.1. The inputs for this algorithm are N, the number of points desired in the representation, and two nondominated points, x^1 and x^2. As mentioned earlier, N must satisfy $N = 2^n + 1$ for some $n \in \mathbb{N}$. If x^1 and x^2 are chosen to be the individual objective minima for the transformed problem, we determine their values using procedure BCSA Initialization (Figure 4.2). Otherwise, we may use two points chosen by the DM. In either case, these points are used as the reference points in the first iteration. We next solve problem (4.6) to obtain the center point between our reference points and we insert this point between the reference points in the list L. If the length of L is shorter than the desired length N, then we solve (4.6) with the first and second points in the list as reference points and, again, with the second and third points as reference points. Each time, we insert the newly generated point into the list L between the two points which produced it. We continue in this manner until the desired cardinality is reached. The algorithm outputs the list L containing the N efficient points that were produced.

We can slightly alter the previously discussed algorithm so that the spacing between generated points is used as the stopping criterion instead of the cardinality. In this case, the DM would specify a distance δ such that she would like the distance between two nondominated points to be δ or less. The modified algorithm, BCSA_δ, is identical to BCSA_N except for the changes shown in Figure 4.3. Note that to determine if the stopping criterion has been met, we look only at the spacing between the first two points in the list L. This is acceptable because we know that we have at least approximately equidistant spacing throughout the representation. Whether we have exactly equidistant spacing depends on the chosen norm and on the cone C which we discuss next.

When the DM's preferences are modeled by an obtuse cone C and the l_1-norm is

```
algorithm BCSA_N
obtain x^1 and x^2 from DM or using Initialization;
input: N, x^1, x^2
begin
    set L = {x^1, x^2};
    while length of L < N do
    begin
        set L' = L;
        for i = 1 to the length of L
            solve BCS(L(i), L(i+1), A) to obtain x*;
            insert x* into L' between L(i) and L(i+1);
            i++;
        end for;
        set L = L';
    end while;
end;
output: L
```

Figure 4.1: Pseudocode for BCSA_N

```
procedure Initialization
input: BOP
begin
    find x^1 ∈ arg lex min{[g_1(x), g_2(x)] : x ∈ X};
    find x^2 ∈ arg lex min{[g_2(x), g_1(x)] : x ∈ X};
end;
output: x^1, x^2
```

Figure 4.2: Pseudocode for Initialization

algorithm BCSA_δ

obtain x^1 and x^2 from DM or using *Initialization*;

input: δ, x^1, x^2

begin

set $L = \{x^1, x^2\}$;
while $||f(L(1)) - f(L(2))|| > \delta$ **do**
begin
set $L' = L$;
for $i = 1$ to the length of L
solve BCS($L(i)$, $L(i+1)$, A) to obtain x^*;
insert x^* into L' between $L(i)$ and $L(i+1)$;
$i++$;
end for;
set $L = L'$;
end while;

end;

output: L

Figure 4.3: Pseudocode for BCSA_δ

selected, we have exactly equidistant spacing because as discussed in Section 3.1, in this case, the triangle inequality becomes an equality. If $p > 1$ or an acute cone is used, we can no longer use the strengthened version of the triangle inequality, so we cannot guarantee that we have exactly equidistant spacing throughout the representation. However, each generated nondominated point is placed so that it is equidistant from the two input points by which it was produced. In other words, if the solution of BCS(x^{1*}, x^{2*}, A) is $\hat{x}$, then $||f(x_a^*) - f(\hat{x})||_p = ||f(x_b^*) - f(\hat{x})||_p$ (note that this is our objective function from (4.6)).

Controlled-Tradeoff We now propose a method for controlling the tradeoffs of the generated solution points. Since the variable u in (4.5) represents the tradeoff between the objective functions, u can be used to generate a Pareto point with a specified tradeoff simply by changing the objective function as shown in (4.7) where t is the desired tradeoff.

$$\begin{aligned} \text{minimize} \quad & ||u - t|| \\ \text{subject to} \quad & \nabla_x f_1(x) + u \nabla_x (f_2(x) - \epsilon) = 0 \\ & f_2(x) \leq \epsilon \\ & x \in X \\ & u(f_2(x) - \epsilon) = 0 \\ & u \geq 0 \end{aligned} \tag{4.7}$$

This problem has several possible applications. It can be used to determine the input points for the BCS method. For instance, if the DM knows that she is not willing to give up more than five units of f_1 to obtain one additional unit of f_2, we could generate the solution point having a tradeoff of one-fifth. This point could then be used as one of the initial input points so that the equidistant representation is generated only over the portion of the Pareto set having a tradeoff of greater than one-fifth. Furthermore, instead of generating equidistant solution points, we can solve formulation (4.7) multiple times to generate a discrete representation with tradeoffs chosen by the DM. Both of these applications using controlled-tradeoff would allow the DM to focus on areas of the nondominated set in which she may be more interested.

4.2 Multiobjective Approach

Extending the Bilevel Controlled Spacing approach to multiobjective problems with more than two criteria is relatively straightforward. Through numerical investigations, we found that two versions of the approach naturally present themselves. The first, which we call the center method, generalizes problem (4.5) exactly as one would expect. That is, just as we input two reference points and find the point equidistant between them in two dimensions, in p dimensions, we input p reference points and find the center point among them. We discuss the details of this method first below. In the second method, the slicing method, instead of working in the dimension p of the problem, we fix the value of a chosen criterion and work instead in $p-1$ dimensions. In fact, we can continue fixing criterion values until we are again in two dimensions at which point we apply equation (4.5) just as before. This method is discussed in more detail following our discussion of the center method below. Finally, it is important again to note that both of these methods are intended for use on convex MOPs because we utilize the KKT optimality conditions which are necessary and sufficient only for convex problems.

4.2.1 Center Method

Pareto Cone The Bilevel Controlled Spacing formulation utilizing the center method (BCSC(x^{1*}, x^{2*}, ..., x^{p*})) is given below where $y^{i*} = f(x^{i*}),\ i = 1, \ldots, p$, are previously produced Pareto points (typically, the individual objective minima in the first iteration).

$$\begin{array}{ll} \text{minimize} & \max\{||y^{1*} - f(x)||, ||y^{2*} - f(x)||, \ldots, ||y^{p*} - f(x)||\} \\ \text{subject to} & x \in X_{WE} \end{array} \tag{4.8}$$

As discussed in the biobjective case in Section 4.1, we may rewrite the condition $x \in X_{WE}$ in equation (4.8) in terms of the ϵ-constraint method:

$$\begin{aligned}
\text{minimize} \quad & \max\{||y^{1*} - f(x)||, ||y^{2*} - f(x)||, \ldots, ||y^{p*} - f(x)||\} \\
\text{subject to} \quad & x = \arg\min \quad f_1(\tilde{x}) \\
& \qquad\qquad \text{subject to} \;\; f_i(\tilde{x}) \leq \epsilon_i, \;\text{ for } i = 2, \ldots, p \\
& \qquad\qquad\qquad\qquad\quad \tilde{x} \in X
\end{aligned} \tag{4.9}$$

Next, we reformulate (4.9) as a single-level optimization problem by replacing the lower-level problem with its KKT conditions:

$$\begin{aligned}
\text{minimize} \quad & \max\{||y^{1*} - f(x)||, ||y^{2*} - f(x)||, \ldots, ||y^{p*} - f(x)||\} \\
\text{subject to} \quad & \nabla_x f_1(x) + u_2 \nabla_x (f_2(x) - \epsilon_2) + \ldots + u_p \nabla_x (f_p(x) - \epsilon_p) = 0 \\
& f_i(x) \leq \epsilon_i, \;\text{ for } i = 2, \ldots, p \\
& u_i(f_i(x) - \epsilon_i) = 0, \;\text{ for } i = 2, \ldots, p \\
& u_i \geq 0, \;\text{ for } i = 2, \ldots, p \\
& x \in X
\end{aligned} \tag{4.10}$$

where $u_i \in \mathbb{R}$ are dual multipliers.

In three dimensions, we can picture the center method as finding the center Pareto point of the curved triangle formed by the three input points. For each Pareto point generated, three new triangles are formed and continuing in this way, we investigate all regions of the Pareto set. As discussed in Section 4.1, we are only guaranteed that the newly produced point is as close to equidistant as possible between the generating input points. Because of this, the areas of the triangles may differ, so certain areas of the Pareto set will need more refinement than others. Thus, when implementing the center method, we investigate a triangle only if its area (which we estimate as the area of the planar triangle formed by the input points) is greater than a prespecified level. This prevents redundancies in the resulting representation.

General Cones Just as in the biobjective approach, we can generalize the BCSC method to notions of optimality defined by arbitrary convex, polyhedral preference cones using the results given in Proposition 2.0.8. The single-level optimization problem given a general cone C defined by the matrix A, BCSC(x^{1*}, x^{2*}, ..., x^{p*}, A), is

$$\begin{aligned} \text{minimize} \quad & \max\{||y^{1*} - f(x)||, ||y^{2*} - f(x)||, \ldots, ||y^{p*} - f(x)||\} \\ \text{subject to} \quad & \nabla_x g_1(x) + v_2 \nabla_x (g_2(x) - \epsilon_2) + \ldots + v_p \nabla_x (g_p(x) - \epsilon_p) = 0 \\ & g_i(x) \leq \epsilon_i, \text{ for } i = 2, \ldots, p \\ & v_i(g_i(x) - \epsilon_i) = 0, \text{ for } i = 2, \ldots, p \\ & v_i \geq 0, \text{ for } i = 2, \ldots, p \\ & x \in X \end{aligned} \tag{4.11}$$

where $v_i \in \mathbb{R}$ are dual multipliers associated with the transformed functions.

Proposition 4.2.1. *Let C be a convex cone defined by $C = \{y \in \mathbb{R}^p : Ay \geqq 0\}$. If $\hat{x}$ is an optimal solution to BCSC(x^{1*}, x^{2*}, ..., x^{p*}, A), then $f(\hat{x}) \in N_W(Y, C)$.*

Proof. Let $\hat{x}$ be an optimal solution to BCSC(x^{1*}, x^{2*}, ..., x^{p*}, A). Since the KKT optimality conditions are both necessary and sufficient for optimality and the ϵ-constraint method guarantees at least a weakly efficient point, $\hat{x} \in E_W(X, g, \mathbb{R}^p_{\geqq}) = E_W(X, Af, \mathbb{R}^p_{\geqq})$. By Proposition 2.0.8, $\hat{x} \in E_W(X, f, C)$ which implies $f(\hat{x}) \in N_W(Y, C)$. This completes the proof. □

The advantages of this method are that it can be implemented in exactly the same manner as the BCSA for biobjective problems given in Section 5.1, although the bookkeeping for the algorithm becomes more complex. Additionally, a check for the area of each search region (in three dimensions, a triangle) needs to be added. The major drawback of this method is that it is not conducive to visualization for more than three objectives. This disadvantage led us to the idea for the slicing method which we discuss next.

4.2.2 Slicing Method

Pareto Cone Recall that one of the goals of our work is to aid in the visualization of the nondominated set. In this respect, however, the center method left us unsatisfied. Because of this, we suggest an alternate method for applying the Bilevel Controlled Spacing method to multiobjective problems, called the slicing method. The main concept behind the slicing method is that although we may not be able to see the Pareto set as a whole in higher dimensions, we can visualize its two- or three-dimensional cross-sections.

Let $j, k \in \{1, 2, \ldots, p\}$, $j < k$, and

$$\tilde{X} = \{x \in \mathbb{R}^n : f_i(x) = f_i^* \text{ for } i = 1, \ldots, p, i \neq j, i \neq k\} \tag{4.12}$$

where the f_i^* values are fixed scalars. Given this, the Bilevel Controlled Spacing Formulation using the slicing method (BCSS(x^{j*}, x^{k*})) is given below:

$$\begin{array}{ll} \text{minimize} & \max\{||y^{j*} - f(x)||, ||y^{k*} - f(x)||\} \\ \text{subject to} & x \in \tilde{X} \\ & x \in X_{WE} \end{array} \tag{4.13}$$

where

$$\begin{array}{rll} x^{j*} = & \arg\operatorname{lex}\min & [f_j(x), f_k(x)] \\ & \text{subject to} & x \in \tilde{X} \\ & & x \in X_{WE} \end{array} \tag{4.14}$$

and

$$\begin{array}{rll} x^{k*} = & \arg\operatorname{lex}\min & [f_k(x), f_j(x)] \\ & \text{subject to} & x \in \tilde{X} \\ & & x \in X_{WE} \end{array} \tag{4.15}$$

Note that in (4.13), $y^{j*} = f(x^{j*})$ and $y^{k*} = f(x^{k*})$. As in the center method, we rewrite

$x \in X_{WE}$ using the ϵ-constraint method.

$$\begin{aligned}
\text{minimize} \quad & \max\{||y^{j*} - f(x)||, ||y^{k*} - f(x)||\} \\
\text{subject to} \quad & x \in \tilde{X} \\
& x = \arg\min \quad f_1(\tilde{x}) \\
& \qquad \text{subject to } \; f_i(\tilde{x}) \leq \epsilon_i, \text{ for } i = 2, \ldots, p \\
& \qquad\qquad\qquad \tilde{x} \in X
\end{aligned} \tag{4.16}$$

Finally, we arrive at our single-level optimization problem by rewriting the lower-level ϵ-constraint problem in terms of its KKT conditions.

$$\begin{aligned}
\text{minimize} \quad & \max\{||y^{j*} - f(x)||, ||y^{k*} - f(x)||\} \\
\text{subject to} \quad & x \in \tilde{X} \\
& \nabla_x f_1(x) + u_2 \nabla_x (f_2(x) - \epsilon_2) + \ldots + u_p \nabla_x (f_p(x) - \epsilon_p) = 0 \\
& f_i(x) \leq \epsilon_i, \text{ for } i = 2, \ldots, p \\
& u_i(f_i(x) - \epsilon_i) = 0, \text{ for } i = 2, \ldots, p \\
& u_i \geq 0, \text{ for } i = 2, \ldots, p \\
& x \in X
\end{aligned} \tag{4.17}$$

In problems (4.13) through (4.17), we reduce the $p-$dimensional Pareto set to a two-dimensional cross-section. To reduce the Pareto set to a three-dimensional cross-section, we would instead fix all but three of the objective function values. Also note that the same process as described above would be use to solve problems (4.14) and (4.15).

General Cones Notice that the slicing method essentially reduces the initial optimization problem to two or three dimensions at which point the center method is applied. Because of this, the derivation of the BCSS method for general cones is the same as discussed in the previous section. Proposition 4.2.1 also applies here.

Implementation To produce a discrete representation of the entire nondominated set, we need to generate a collection of cross-sections. That is, we must systematically vary the values of f_i^* in the set $\tilde{X}$ (4.12). The Bilevel Controlled Spacing Slicing Algorithm (BCSSA) for general cones is shown in Figure 4.4. Note that as mentioned, the biobjective bilevel algorithm is nested inside the multiobjective algorithm. BCSSA takes as inputs the indices (j,k) of the objective functions that are not fixed, the scalars n_i which are the number of slices desired with respect to each fixed objective function, and the spacing value δ which is used as the stopping criterion. Next, for each fixed objective function, we determine the minimum and maximum values, f_i^l and f_i^h, over the nondominated set. We calculate the distance between these extreme values and divide by $n_i - 1$ to find the spacing between each slice. A series of nested for-loops is used to investigate all possible cross-sections. Note that these loops are only over the fixed objective functions. For each possible combination of the fixed values, we specify the set $\tilde{X}$ and determine the minimum values of $f_j(x)$ and $f_k(x)$ over this cross-section of the nondominated set. These points are used as the reference points in the biobjective BCSA$_\delta$. We recommend the use of BCSA$_\delta$ over BCSA$_N$ because the length of each cross-section will (most likely) be different. The former refines as much or as little as necessary to meet the spacing criterion, while the latter does not adapt to the problem and generates N nondominated points regardless of the length of the cross-section resulting in highly variable spacing among the cross-sections. Each time we run BCSA$_\delta$, we obtain a list L of efficient points which we add to the list of lists, $\mathcal{L}$. BCSSA returns $\mathcal{L}$ at the conclusion of the algorithm.

Algorithmically, when applying the slicing method to an MOP, it does not matter which objectives we choose to fix. However, the resulting representations can be quite different in terms of decision-making because different sets of nondominated points (i.e., different alternatives) are presented to the DM. Additionally, due to the increased complexity of higher dimensions, slicing with respect to different objective functions can lead to varying coverage in certain areas of the nondominated set: in particular, along the boundary of the

algorithm BCSSA

input: $j, k \in \{1, \ldots, p\}$, n_i for $i = 1, \ldots, p, i \neq j, i \neq k$, δ

begin

find $f_i^l = \min\{f_i(x) : x \in E_W(X, f, C)\}$ for $i = 1, \ldots, p, i \neq j, i \neq k$;
find $f_i^h = \max\{f_i(x) : x \in E_W(X, f, C)\}$ for $i = 1, \ldots, p, i \neq j, i \neq k$;
compute $d_i = (f_i^h - f_i^l)/(n_i - 1)$ for $i = 1, \ldots, p, i \neq j, i \neq k$;
set $\mathcal{L} = \{\}$;
for $count_1 = 1, \ldots, n_1$
$f_1^* = f_1^l + (count_1 - 1)d_1$;
$\vdots$
for $count_p = 1, \ldots, n_p$
$f_p^* = f_p^l + (count_p - 1)d_p$;
set $\tilde{X} = \{x \in \mathbb{R}^n : f_i(x) = f_i^*$ for $i = 1, \ldots, p, i \neq j, i \neq k\}$;
find $x^1 \in \arg\operatorname{lex}\min\{[f_j(x), f_k(x)] : x \in \tilde{X}, x \in E_W(X, f, C)\}$;
find $x^2 \in \arg\operatorname{lex}\min\{[f_k(x), f_j(x)] : x \in \tilde{X}, x \in E_W(X, f, C)\}$;
run algorithm BCSA$_\delta$ with inputs δ, x^1, x^2 to obtain L;
append L to $\mathcal{L}$;
end for;
end for;

end;

output: $\mathcal{L}$

Figure 4.4: Pseudocode for BCSSA

nondominated set. Both of these issues can be seen in Figures 4.5 and 4.6 which illustrate the same nondominated set sliced with respect to two different objective functions. Note that in Figure 4.5 we lack coverage around the nondominated point farthest to the right of the figure, while in Figure 4.6 we lack coverage around the nondominated point at the top of the figure. Given this, if the DM has no preferences about which objective to fix, it would be beneficial to generate several different representations sliced with respect to different objective functions. On the other hand, if the DM is more informed about one of the objective functions than the others, then we recommend slicing with respect to this objective function.

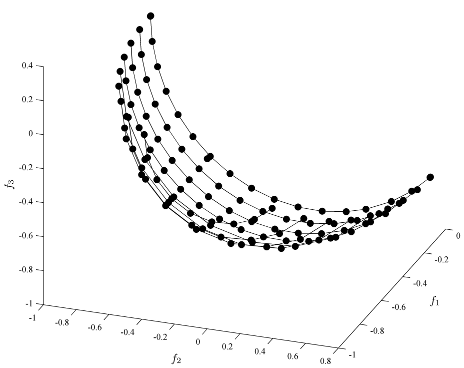

Figure 4.5: Example sliced wrt f_1

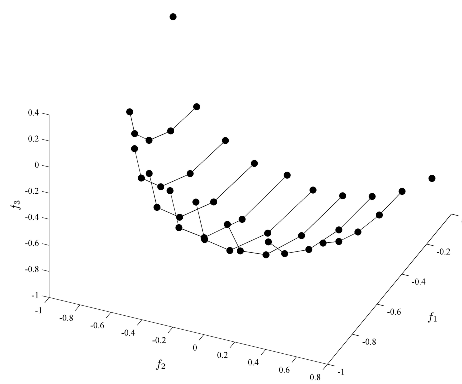

Figure 4.6: Example sliced wrt f_2

The advantages of the slicing method are that it aids in the visualization of the nondominated set even for problems with more than three dimensions. Additionally, the slicing method can easily integrate DM preferences. For example, the DM can choose to investigate only certain cross-sections quite easily. Along the same lines, it is also possible to integrate the controlled-tradeoff method presented in (4.7) at the two-dimensional level so that the DM can consider specific tradeoffs that may be of interest to her. The main disadvantage of the slicing method is that, as with the other bilevel techniques, it is only applicable to convex MOPs.

Chapter 5

Constraint Controlled Spacing

In this chapter, we present the Constraint Controlled Spacing (CCS) approach for generating discrete representations of the nondominated set. Instead of utilizing a bilevel structure as in the Bilevel Controlled Spacing approach, in this method we add a constraint to the epsilon-constraint problem to control the spacing of newly generated points. The chapter is divided into three sections. In Section 5.1, we discuss the CCS approach for biobjective problems; we begin assuming the Pareto notion of optimality (Section 5.1.1) and then move to notions of optimality defined by general cones (Section 5.1.2). In Section 5.1.3, we discuss the implementation of the biobjective approach, and in Section 5.1.4, we consider a family of elliptic norms. In Section 5.2, we give some linear algebra results related to simplicial cones. These results are used in Section 5.3 where we present the CCS approach for multiobjective problems, again beginning with the Pareto cone (Section 5.3.1) and then moving to general cones (Section 5.3.2). Section 5.3.3 concludes the chapter with a discussion of the implementation of the multiobjective approach.

5.1 Biobjective Approach

5.1.1 Pareto Cone

The epsilon constraint method [17] is a well-known scalarization in the field of multiple-objective programming. The formulation ($\mathrm{P}_1(\epsilon)$) for the biobjective case is given in (5.1):

$$\begin{aligned} \text{minimize} \quad & f_1(x) \\ \text{subject to} \quad & f_2(x) \leq \epsilon \\ & x \in X \end{aligned} \tag{5.1}$$

where $\epsilon \in \mathbb{R}$ is a parameter. In [17], it is proved that if $\hat{x}$ is an optimal solution of $\mathrm{P}_1(\epsilon)$ for some ϵ, then $f(\hat{x}) \in Y_{WN}$. Alternately, it is shown that if $f(\hat{x}) \in Y_{WN}$, then there exists an ϵ such that $\hat{x}$ is an optimal solution of $\mathrm{P}_1(\epsilon)$. Typically, ϵ is varied in a uniform manner between two fixed reference points to obtain a discrete representation of the Pareto set (see for example, [40]). However, in almost every case, uniformly varying ϵ in the epsilon constraint method does not lead to uniformly spaced Pareto points in the representation as shown in Figure 5.1. To remedy this, we add a spacing constraint to the epsilon constraint formulation to ensure equidistant spacing of the generated Pareto points between the reference points which, without loss of generality, are chosen to be y^1 (4.2) and y^2 (4.3). Alternatively, the reference points may be chosen by the DM. Notice that if we produce equidistant Pareto points between our two "end" points, then we have attained complete coverage of the Pareto set as well.

The Constraint Controlled Spacing formulation ($\mathrm{CCS}_1(\delta, x^*)$) for a biobjective program, assuming Pareto preferences, is as follows. Let $x^* \in X_{WE}$ such that $||f(x^*) - y^2|| \geq \delta$ where δ is a fixed scalar representing the desired spacing between points in the Pareto set.

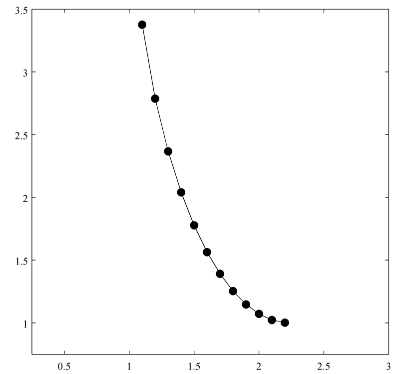

Figure 5.1: A Pareto set generated using uniformly spaced epsilons

Further, let $||\cdot||$ be an l_p-norm. Then $\text{CCS}_1(\delta,\, x^*)$ is defined as:

$$\begin{aligned} \text{minimize} \quad & f_1(x) \\ \text{subject to} \quad & f_1(x) \le f_1(x^2) \\ & f_2(x) \le f_2(x^*) \\ & ||f(x) - f(x^*)|| \ge \delta \\ & x \in X \end{aligned} \tag{5.2}$$

The requirement that $||f(x^*) - y^2|| \ge \delta$ is needed to ensure that (5.2) has a feasible solution. The first constraint in (5.2) is needed only from a theoretical point of view to ensure that all feasible solutions to (5.2) are "above" the weak Pareto set. This is used in the proof of Proposition 5.1.3 when we assume that we can reduce the f_2-value of any feasible solution by a certain amount and obtain another feasible point. This constraint can be omitted in practice since we are minimizing $f_1(x)$. The second constraint is similar to the epsilon constraint in (5.1). Here, however, a previously produced point (in the first iteration, y^1) plays the role of epsilon and is used as a reference for placing the new point. The third constraint controls the spacing by forcing the new point to be at least δ distance away from the input point. This constraint takes the form of the δ-level curve for the appropriate l_p-norm, and, thus, is nonlinear. Notice that the epsilon constraint eliminates the majority of (or, in the cases of the l_1- and l_∞-norms, eliminates entirely) the non-convexity in the feasible region created by the spacing constraint (see Figure 5.2). After the first iteration, the newly generated point becomes the input point, and in this way, we "walk down" the Pareto set until we reach a fixed stopping point (in our work, y^2). Figure 5.2(b) shows a typical iteration with the diamond-shaped constraint representing the δ-level curve for the l_1-norm. The parameter δ is selected based on the cardinality preferences of the DM and is discussed at the end of this section.

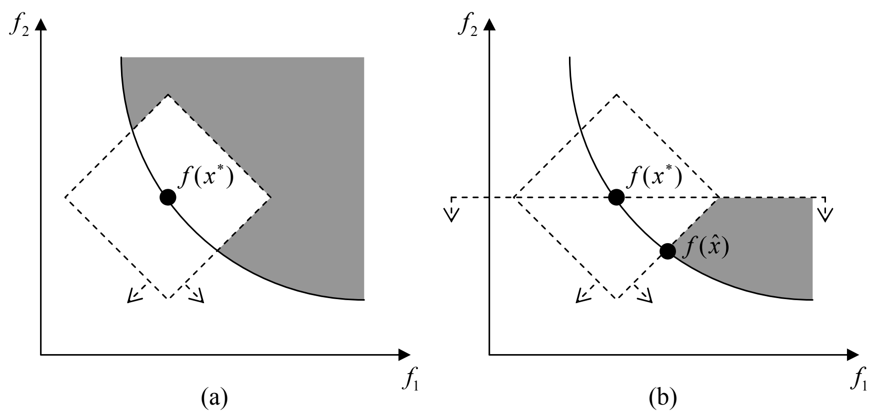

Figure 5.2: Example feasible regions for the l_1-norm (a) without and (b) with the epsilon constraint

5.1.2 General Cones

Using the results given in Proposition 2.0.8, the CCS method can be generalized to produce equidistantly-spaced nondominated points where optimality is defined by an arbitrary obtuse preference cone. Let $C \supseteq \mathbb{R}^2_{\geqq}$ be a cone defined by a matrix A and let $z = g(x) = Af(x)$ as in the transformed MOP (2.2). We obtain the reference ("end") points, $z^1 = g(x^1)$ and $z^2 = g(x^2)$, of the Pareto set in a similar manner as before (see (4.2) and (4.3)):

$$z^1 := \operatorname{lex}\min\{[g_1(x), g_2(x)] : x \in X\}, \tag{5.3}$$

and

$$z^2 := \operatorname{lex}\min\{[g_2(x), g_1(x)] : x \in X\}. \tag{5.4}$$

As in formulation (5.2), we ensure that the problem has a feasible solution by requiring that $||f(x^*) - f(x^2)|| \geq \delta$ where x^2 is now defined as the preimage of z^2, δ is the desired spacing, $x^* \in E_W(X, f, C)$, and $|| \cdot ||$ is an l_p-norm. Notice that this requirement is in the space of the original problem, not the transformed problem. The Constraint Controlled Spacing Problem for an obtuse preference cone defined by the matrix A, $\mathrm{CCS}_1(\delta,\, x^*,\, A)$, is formulated as follows where A_i is the i^{th} row of the matrix A:

$$\begin{array}{ll} \text{minimize} & A_1 f_1(x) \\ \text{subject to} & f_1(x) \leq f_1(x^2) \\ & A_2 f_2(x) \leq A_2 f_2(x^*) \\ & ||f(x) - f(x^*)|| \geq \delta \\ & x \in X \end{array} \tag{5.5}$$

which we rewrite with simplified notation as

$$\begin{array}{ll} \text{minimize} & g_1(x) \\ \text{subject to} & f_1(x) \leq f_1(x^2) \\ & g_2(x) \leq g_2(x^*) \\ & ||f(x) - f(x^*)|| \geq \delta \\ & x \in X \end{array} \tag{5.6}$$

The constraints serve the same purposes as discussed at the beginning of the section: the first is purely theoretical as was discussed in the Pareto cone section and is needed here for the same reason, the second is the ϵ-constraint, and the third is the spacing constraint. However, it is clear from (5.6) that the ϵ-constraint and the spacing constraint are now operating in different spaces. That is, the ϵ-constraint is applied to the transformed functions while the spacing constraint is applied to the original functions. This is done because we seek equidistant nondominated points of the original problem. If we had, instead, used the spacing constraint $||g(x) - g(x^*)|| \geq \delta$, we would obtain equidistant Pareto points of the transformed problem. Unfortunately, though, when we recover the nondominated points of the original problem using Proposition 2.0.8, there is no guarantee that the spacing remains equidistant.

We now prove some propositions which lead to our main result.

Proposition 5.1.1. *Let C be a polyhedral cone defined by $C = \{y \in \mathbb{R}^p : Ay \geqq 0\}$. The matrix A has all nonnegative entries if and only if $\mathbb{R}^p_{\geqq} \subseteq C$.*

Proof. ($\Rightarrow$) Suppose the matrix A has all nonnegative entries. That is, suppose $a_{ij} \geq 0$ for all i, j. Let $y \in \mathbb{R}^p_{\geqq}$ which implies $y \geqq 0$ by definition. Then, $Ay \geqq 0$ since $a_{ij} \geq 0$ for all i, j. Thus, $y \in C$.
($\Leftarrow$) Suppose that $\mathbb{R}^p_{\geqq} \subseteq C$, and assume that there exists an i' and a j' such that $a_{i'j'} < 0$. Consider the unit vector $e^{j'} \in \mathbb{R}^p_{\geqq}$ which has a 1 in the j'^{th} position and 0s everywhere else. Multiplying the i'^{th} row of A by $e^{j'}$ gives $A_{i'}e^{j'} = a_{i'j'} < 0$. Thus, $Ae^{j'} \ngeqq 0$ which implies that $e^{j'} \notin C$ and $\mathbb{R}^p_{\geqq} \nsubseteq C$, a contradiction. This completes the proof. $\square$

Proposition 5.1.2. *Any pointed, polyhedral cone $C \subset \mathbb{R}^2$ can be represented by a 2×2 matrix A having positive determinant.*

Proof. Let $C \subset \mathbb{R}^2$ be a pointed cone defined by $C = \{y \in \mathbb{R}^2 : Ay \geqq 0\}$. Since C is pointed, the determinant of A is non-zero. Suppose that the determinant of A is negative. Then one may exchange the rows of A without changing the cone. Exchanging two rows in a matrix changes the sign of the determinant of the matrix, so the determinant of A is now positive. This completes the proof. $\square$

We next prove our main theorem for this section.

Theorem 5.1.3. *Let $C \supseteq \mathbb{R}^2_{\geqq}$ be a pointed cone defined by $C = \{y \in \mathbb{R}^2 : Ay \geqq 0\}$. If $N_W(Y, C)$ is connected and $\hat{x}$ is the unique optimal solution of $CCS_1(\delta,\ x^*,\ A)$, then $f(\hat{x}) \in N_W(Y, C)$.*

Proof. Assume $N_W(Y, C)$ is connected and let $\hat{x}$ be the unique optimal solution of $\text{CCS}_1(\delta,$ x^*, $A)$. Let $\hat{y} = f(\hat{x})$ and $\hat{z} = Af(\hat{x})$. Suppose that $\hat{y} \notin N_W(Y, C)$. By definition, $\hat{y} \notin N_W(Y, C)$ implies $(\hat{y} - \text{int } C) \cap Y \neq \emptyset$. This further implies that $(\hat{y} - C) \cap Y \neq \emptyset$. In particular, since $y^* = f(x^*)$ and $y^2 = f(x^2)$ (where x^2 is defined as the preimage of z^2 from (5.4)) are both in $N_W(Y, C)$, $N_W(Y, C)$ is connected, and $f_1(\hat{x}) \leq f_1(x^2)$, there must exist a $\lambda > 0$ and $\tilde{y} = f(\tilde{x}) \in Y$ such that $\tilde{y} = \hat{y} + \lambda(0, -1)^T$. Let A have the form

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

such that the determinant of A is positive (see Proposition 5.1.2). Then, the resulting components of $\tilde{z}$ are as follows:

$$\tilde{z} = A\tilde{y} = A(\hat{y} + \lambda(0, -1)^T) = \hat{z} + \lambda A(0, -1)^T = \begin{bmatrix} \hat{z}_1 - \lambda a_{12} \\ \hat{z}_2 - \lambda a_{22} \end{bmatrix}$$

Note that since C is obtuse, all components of the matrix A are nonnegative by Proposition 5.1.1. Combining this fact with $\lambda > 0$ gives that

$$\tilde{z}_1 \leq \hat{z}_1, \tag{5.7}$$

and

$$\tilde{z}_2 \leq \hat{z}_2. \tag{5.8}$$

To establish the feasibility of $\tilde{x}$, recall that since $\hat{x}$ is optimal for $\text{CCS}_1(\delta, x^*, A)$, we have

$$\hat{y}_1 \leq y_1^2, \tag{5.9}$$

$$\hat{z}_2 \leq z_2^*, \tag{5.10}$$

and

$$||\hat{y} - y^*|| \geq \delta. \tag{5.11}$$

Combining (5.8) and (5.10) gives $\tilde{z}_2 \leq z_2^*$, so $\tilde{x}$ is feasible to the ϵ-constraint in (5.6). Further, since $\tilde{y} = \hat{y} + \lambda(0, -1)^T$, we know that

$$\tilde{y}_1 = \hat{y}_1, \tag{5.12}$$

and

$$\tilde{y}_2 < \hat{y}_2. \tag{5.13}$$

Combining (5.9) and (5.12) gives $\tilde{y}_1 \leq y_1^2$, so $\tilde{x}$ is feasible to the first constraint in (5.6). To complete the feasibility argument, we determine the relationship between $\hat{y}_2$ and y_2^*. From (5.10) we can derive the following where A_i denotes the i^{th} row of A:

$$\begin{array}{rcl} A_2\hat{y} & \leq & A_2 y^* \\ a_{21}\hat{y}_1 + a_{22}\hat{y}_2 & \leq & a_{21}y_1^* + a_{22}y_2^* \\ \hat{y}_1 & \leq & y_1^* + \frac{a_{22}}{a_{21}}(y_2^* - \hat{y}_2). \end{array} \tag{5.14}$$

Note that since $y^* \in N_W(Y, C)$, then $Ay^* \in AN_W(Y, C)$ which we may rewrite as $z^* \in AN_W(Y, C)$. By Proposition 2.0.8, $AN_W(Y, C) = N_W(AY, \mathbb{R}^p_>)$. Thus, $z^* \in N_W(AY, \mathbb{R}^p_>)$ which by definition means that $(z^* - \mathbb{R}^2_>) \cap AY = \emptyset$. In particular, given (5.10), this implies that we must have $\hat{z}_1 \geq z_1^*$ which gives

$$\begin{array}{rcl} A_1\hat{y} & \geq & A_1 y^* \\ a_{11}\hat{y}_1 + a_{12}\hat{y}_2 & \geq & a_{11}y_1^* + a_{12}y_2^* \\ \hat{y}_1 & \geq & y_1^* + \frac{a_{12}}{a_{11}}(y_2^* - \hat{y}_2). \end{array} \tag{5.15}$$

Combining the last inequalities of (5.14) and (5.15), we have

$$\begin{aligned} y_1^* + \tfrac{a_{12}}{a_{11}}(y_2^* - \hat{y}_2) &\leq y_1^* + \tfrac{a_{22}}{a_{21}}(y_2^* - \hat{y}_2) \\ a_{12}a_{21}(y_2^* - \hat{y}_2) &\leq a_{11}a_{22}(y_2^* - \hat{y}_2) \\ 0 &\leq (a_{11}a_{22} - a_{12}a_{21})(y_2^* - \hat{y}_2) \end{aligned} \tag{5.16}$$

Note that $(a_{11}a_{22} - a_{12}a_{21})$ is the determinant of A which by assumption is positive. Hence, we have that $\hat{y}_2 \leq y_2^*$. This fact along with (5.11), (5.12), and (5.13) implies that $||f(\tilde{x}) - f(x^*)|| > \delta$. Thus, $\tilde{x}$ is feasible to $\text{CCS}_1(\delta,\ x^*,\ A)$ with $\tilde{z}_1 \leq \hat{z}_1$. If $\tilde{z}_1 < \hat{z}_1$, then $\hat{x}$ is not an optimal solution of $\text{CCS}_1(\delta,\ x^*,\ A)$. If $\tilde{z}_1 = \hat{z}_1$, then $\hat{x}$ is not a unique optimal solution of $\text{CCS}_1(\delta,\ x^*,\ A)$. Hence, both cases lead to contradictions which implies that $f(\hat{x}) \in N_W(Y, C)$. This completes the proof. □

Corollary 5.1.4. *Let $C \supseteq \mathbb{R}^2_{\geqq}$ be a pointed cone defined by $C = \{y \in \mathbb{R}^2 : Ay \geqq 0\}$. If $N(Y, C)$ is connected and $\hat{x}$ is the unique optimal solution of $CCS_1(\delta,\ x^*,\ A)$, then $f(\hat{x}) \in N(Y, C)$.*

Proof. Follows directly from the proof of Theorem 5.1.3. □

The following proposition shows that the spacing between nondominated points actually equals δ. That is, $\text{CCS}_1(\delta,\ x^*,\ A)$ produces equidistant nondominated points when $N_W(Y, C)$ is connected.

Proposition 5.1.5. *Let $C \supseteq \mathbb{R}^2_{\geqq}$ be a pointed cone defined by $C = \{y \in \mathbb{R}^2 : Ay \geqq 0\}$ and assume $N_W(Y, C)$ is connected. If $x^* \in E_W(X, f, C)$ and $\hat{x}$ is the unique optimal solution of $CCS_1(\delta,\ x^*,\ A)$, then $||f(\hat{x}) - f(x^*)|| = \delta$.*

Proof. Assume $N_W(Y, C)$ is connected. Let $x^* \in E_W(X, f, C)$ and $\hat{x}$ be the unique optimal solution of $\text{CCS}_1(\delta,\ x^*,\ A)$. Then by Theorem 5.1.3, $f(\hat{x}) \in N_W(Y, C)$. Suppose that $||f(\hat{x}) - f(x^*)|| > \delta$. Note that since $x^* \in E_W(X, f, C)$ and $N_W(Y, C)$ is connected, we must have that $\{x \in \mathbb{R}^2 : ||f(x) - f(x^*)|| = \delta\} \cap N_W(Y, C) \neq \emptyset$. Further, since we require $||f(x^*) - f(x^2)|| \geq \delta$, we must have that $\{x \in \mathbb{R}^2 : ||f(x) - f(x^*)|| = \delta, f_1(x) \leq$

$f_1(x^2)$, and $g_2(x) \leq g_2(x^*)\} \cap N_W(Y, C) \neq \emptyset$. Thus, let $\tilde{x} \in E_W(X, f, C)$ such that $g_2(\tilde{x}) \leq g_2(x^*)$ and $||f(\tilde{x}) - f(x^*)|| = \delta$. Then we must have the following ordering with respect to g_1: $g_1(x^*) \leq g_1(\tilde{x}) \leq g_1(\hat{x})$ with all points distinct. If $g_1(\tilde{x}) < g_1(\hat{x})$, then $\hat{x}$ is not an optimal solution of $\mathrm{CCS}_1(\delta,\, x^*,\, A)$. If $g_1(\tilde{x}) = g_1(\hat{x})$, then $\hat{x}$ is not a unique optimal solution $\mathrm{CCS}_1(\delta,\, x^*,\, A)$. Hence, both cases lead to contradictions. This completes the proof. □

If the nondominated set is not connected, we are still guaranteed to have equidistant points in the regions of $N_W(Y, C)$ which are locally connected.

If $\hat{x}$ is not the unique solution of $\mathrm{CCS}_1(\delta,\, x^*,\, A)$, one cannot guarantee that $f(\hat{x})$ is nondominated (see Figure 5.3). Fortunately, this is only an issue with the l_∞-norm in the family of l_p-norms. Thus, if we use any other l_p-norm, the uniqueness restriction may be relaxed. Further, note that while Theorem 5.1.3 requires that the solution $\hat{x}$ be unique, we actually only need $f(\hat{x})$ to be unique.

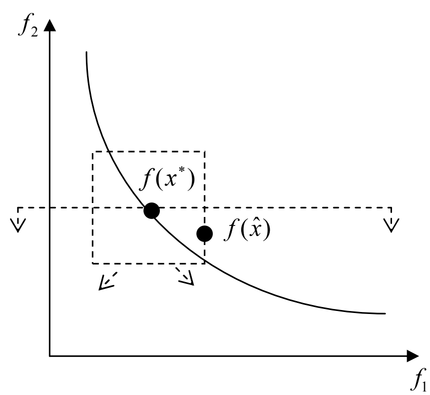

Figure 5.3: A dominated solution may be produced when using the l_∞-norm.

If $N_W(Y, C)$ is not connected, $\mathrm{CCS}_1(\delta,\, x^*,\, A)$ may not produce a nondominated point as shown in Figure 5.4. However, one may still apply the method to a problem with a disconnected nondominated set by checking the nondominance of the solution point $\hat{x}$. This check is performed by solving an instance of the epsilon constraint problem, denoted $\mathrm{P}_2(\epsilon)$, in the space of the transformed problem. We let epsilon be the appropriate criterion value of the current candidate solution $\hat{x}$ (as shown in (5.17)).

$$\begin{aligned} \text{minimize} \quad & g_2(x) \\ \text{subject to} \quad & g_1(x) \leq g_1(\hat{x}) \\ & x \in X \end{aligned} \tag{5.17}$$

As mentioned previously, a solution to the epsilon constraint problem is at least weakly nondominated, so if the solution to the check problem, x', is equal in the objective space to the candidate solution (i.e., $g_2(x') = g_2(\hat{x})$), then the candidate solution is weakly non-dominated. Otherwise, we discard the candidate solution and use the solution of the check problem as our new input point.

Additionally, notice that if a larger δ is chosen, the disconnected portion of the nondominated set may become a non-issue. That is, depending on the magnitude of δ, the size of the gap in the nondominated set, and the placement of $f(x^*)$, it is possible that the algorithm will not detect the gap in the nondominated set and will run as if the nondominated set were connected.

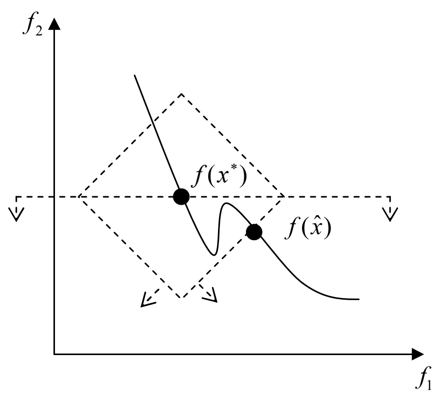

Figure 5.4: A dominated point may be produced when Y_{WN} is not connected.

The selection of the parameter δ should be based on the chosen l_p-norm and the DM's cardinality preference, N. Note that N should be the largest number of solution points that the DM can simultaneously compare. We first consider the case where the (weakly) nondominated set is known to be connected. If the l_1-norm is selected, given the input points (i.e., z^1 and z^2) and N (which includes the input points), δ should be chosen as

$$\delta = \frac{||z^1 - z^2||_1}{N - 1}. \tag{5.18}$$

The formula in (5.18) is valid because the triangle inequality becomes an equality for the l_1-norm when the (weakly) nondominated set is monotonic (which is the case when C is obtuse). When $p > 1$, one can no longer use the strengthened version of the triangle inequality regardless of the curvature of the (weakly) nondominated set, so δ should be chosen experimentally as:

$$\frac{||z^1 - z^2||_1}{N - 1} \leq \delta \leq \frac{||z^1 - z^2||_\infty}{N - 1}. \tag{5.19}$$

Notice that if a δ is chosen that does not "fit" the specific problem being solved, the spacing between the final generated nondominated point and the second input point will be smaller than δ. If no information is known about the connectivity of the (weakly) nondominated set *a priori*, the guidelines above should still be followed for the selection of δ. However, in this case, the number of solution points generated is less than or equal to N.

In the case of an acute cone, $\text{CCS}_1(\delta,\ x^*,\ A)$ may not produce a nondominated point. Figure 5.5 (a) shows the cone C and its associated negative cone $-C$. Figure 5.5 (b) shows how this cone would be applied to a generic problem. The oblique ray of $-C$ is used as the constraint while the other represents level curves of the objective function. In this example, we simply minimize f_1. Several level-curves of f_1 are shown along with the level curve corresponding to the optimal solution $\hat{x}$. However, notice that $f(\hat{x})$ is clearly not a nondominated point. This occurs because of the extreme non-convexity of the feasible region when an acute cone is used. On the other hand, when an obtuse cone is used, the ϵ-constraint eliminates the majority of (or, in the cases of the l_1- and l_∞-norms, eliminates entirely) the non-convexity in the feasible region created by the spacing constraint (see Figure 5.6).

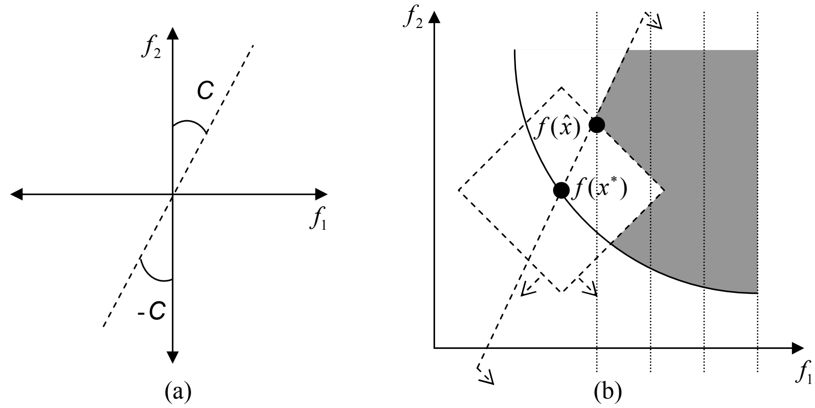

Figure 5.5: Example of $\mathrm{CCS}_1(\delta,\, x^*,\, A)$ when C is acute

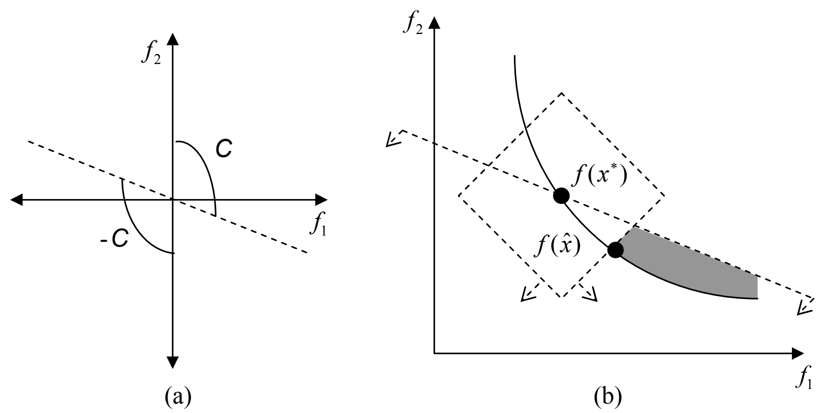

Figure 5.6: Example of $\mathrm{CCS}_1(\delta,\, x^*,\, A)$ when C is obtuse

5.1.3 Implementation

The Constraint Controlled Spacing Algorithm (CCSA) is shown in Figure 5.7. The algorithm begins by determining the individual objective minima (see Figure 4.2) which are used as the initial reference points. Other points chosen by the DM may be used instead. The while-loop depends on the chosen cardinality, N, and on the requirement that the problem has a feasible solution. Recall that computationally, the constraint $f_1(x) \leq f_1(x^2)$ is unnecessary, so this constraint my be eliminated when solving $\text{CCS}_1(\delta, x^*, A)$. The check for nondominance is integrated into the algorithm after the initial "solve" step. If, however, the problem is known to have a connected nondominated set, this check may be eliminated. The algorithm outputs L, the list of efficient solutions.

5.1.4 Elliptic Norms

In the context of the Pareto cone, we also consider elliptic norms which are a class of norms not usually applied to multiple-objective programming. Rustem [64] introduces the following elliptic norm which is defined by a symmetric, positive definite, $p \times p$ matrix Q:

$$||y||_Q^2 = y^T Q y. \tag{5.20}$$

In terms of decision-making, the matrix Q allows DMs to quantify their preferences. The diagonal elements represent the importance of each individual criterion, while the off-diagonal elements represent the DMs tradeoff preferences.

Unfortunately, replacing the norm in problem (5.2) with the elliptic norm in (5.20) is not as straightforward as it may appear. Choosing the matrix Q is the first difficulty. It is unclear how to determine the magnitudes of the entries and through numerical experiments, we have found that the magnitudes of the elements of Q do not necessarily coincide with the magnitudes of the objective functions. Applying the norm in (5.20) to our problem and

algorithm CCSA

obtain x^1 and x^2 from DM or using *Initialization*

input: δ, N

begin

let $x^* = x^1$, let $L = \{x^1\}$, and set $i = 1$;
while $i \leq N - 2$ and $||f(x^*) - f(x^2)|| \geq \delta$ **do**
begin
solve $\text{CCS}_1(\delta, x^*, A)$ to obtain $x^{(i)}$;
solve $\text{P}_2(g_1(x^{(i)}))$ to obtain x';
if $g_2(x^{(i)}) = g_2(x')$ **then** set $x^* = x^{(i)}$;
else set $x^* = x'$;
end if;
set $L(i + 1) = x^*$;
$i + +$;
end while;
set $L(N) = x^2$;

end;

output: L

Figure 5.7: Pseudocode for CCSA

doing a bit of algebra we have:

$$\begin{aligned} ||f(x) - f(x^*)||_Q^2 &= (f(x) - f(x^*))^T Q(f(x) - f(x^*)) \qquad (5.21) \\ &= q_{11}(f_1(x) - f_1(x^*))^2 + 2q_{12}(f_1(x) - f_1(x^*))(f_2(x) - f_2(x^*)) \\ &\quad + q_{22}(f_2(x) - f_2(x^*))^2 \end{aligned}$$

where

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix}.$$

Note that since Q is positive definite, we know that $q_{11}q_{22} - q_{12}^2 > 0$ which implies that $q_{12}^2 - q_{11}q_{22} < 0$, and additionally that $q_{12}^2 - 4q_{11}q_{22} < 0$. Hence (see for example [1]), equation (5.21) is the equation of a rotated ellipse with center $(f_1(x^*), f_2(x^*))$. On the other hand, we can use the standard rotation matrix to rotate each of the ordered pairs α degrees from the horizontal, as shown in (5.22).

$$[f_1(x) - f_1(x^*),\ f_2(x) - f_2(x^*)] \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \qquad (5.22)$$

$$= \begin{bmatrix} \cos(\alpha)(f_1(x) - f_1(x^*)) - \sin(\alpha)(f_2(x) - f_2(x^*)) \\ \sin(\alpha)(f_1(x) - f_1(x^*)) + \cos(\alpha)(f_2(x) - f_2(x^*)) \end{bmatrix}^T. \qquad (5.23)$$

Substituting the results from (5.23) into the usual equation for an ellipse and simplifying gives

$$\begin{aligned} a^2b^2 &= (b^2\cos^2(\alpha) + a^2\sin^2(\alpha))(f_1(x) - f_1(x^*))^2 \qquad (5.24) \\ &\quad - 2\cos(\alpha)\sin(\alpha)(b^2 - a^2)(f_1(x) - f_1(x^*))(f_2(x) - f_2(x^*)) \\ &\quad + (b^2\sin^2(\alpha) + a^2\cos^2(\alpha))(f_2(x) - f_2(x^*))^2 \end{aligned}$$

where a is the radius of the ellipse in the f_2 direction, b is the radius of the ellipse in the

f_1 direction, and α is the degree of rotation of the ellipse from horizontal (see Figure 5.8). Note that when applying equation (5.24) to problem (5.2), we have a less than or equal to instead of an equality with a^2b^2 serving in the place of δ. The equation in (5.24) is much easier to apply than (5.21) because a, b, and α are all quantities which are easily interpreted in the context of the problem. However, the meaning of a, b, and α in terms of decision making is unclear.

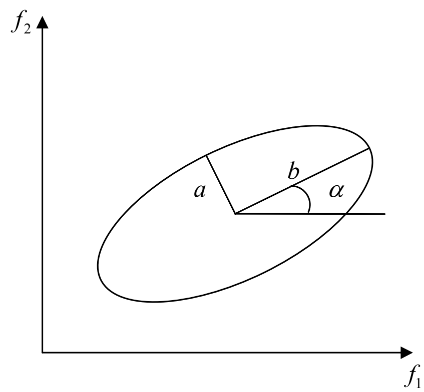

Figure 5.8: A rotated ellipse

Comparing the equations in (5.21) and (5.24), we see that

$$
\begin{aligned}
q_{11} &= b^2\cos^2(\alpha) + a^2\sin^2(\alpha), \\
q_{12} &= -\cos(\alpha)\sin(\alpha)(b^2 - a^2), \text{ and} \\
q_{22} &= b^2\sin^2(\alpha) + a^2\cos^2(\alpha).
\end{aligned}
$$

Thus, it seems as though we should be able to resolve the issue of the matrix Q being difficult to specify but having a clear application to decision making on the one hand, and the quantities a, b, and α being easy to select but not having a straightforward interpretation in terms of decision making, on the other. However, approaching the problem from either direction leads the DM to difficulties. The DM understands the meaning of the matrix Q but cannot quite determine how to choose the entries, and she can easily choose the quantities a, b, and α, but is not quite sure how these quantities reflect her preferences.

5.2 Linear Algebra for Simplicial Cones

In this section, we prove several propositions about obtuse, simplicial cones $C \supseteq \mathbb{R}^p_{\geqq}$ defined by $C = \{y \in \mathbb{R}^p : Ay \geqq 0\}$. Note that by Proposition 5.1.1, the matrix A has all nonnegative entries. Moreover, since C is simplicial, A is a $p \times p$ matrix.

Given a matrix A, a minor of A is a matrix obtained by deleting certain rows and columns of A. We denote a minor of A by $M_{(i_1,j_1)}(A)$ where the subscript specifies which rows and columns to delete: in this case, row i_1 and column j_1. If more than one row and one column are deleted, more ordered pairs are appended to the subscript. For instance, $M_{(i_1,j_1),(i_2,j_2)}(A)$ is the minor of A with rows i_1 and i_2 and columns j_1 and j_2 deleted. If we let

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},$$

then

$$M_{(1,1)}(A) = \begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix},$$

and

$$M_{(1,1)(2,2)}(A) = [9] \, .$$

Notice that the minor notation is not unique. That is, $M_{(1,1),(2,2)}(A)$ and $M_{(1,2),(2,1)}(A)$ represent the same minor ([9]) because the same rows and columns are deleted. Lastly, we denote the determinant operator by $\det[\cdot]$.

In both Proposition 5.2.1 and Proposition 5.2.2, for ease of notation and for the sake of brevity, we prove specific instances of general results. Namely, in Proposition 5.2.1, the specific ordered pairs $(i_1, 1)$ and $(i_2, 3)$ were chosen without loss of generality based on the needs of Proposition 5.2.2. Likewise, the specific ordered pairs $(1, 1)$ and $(1, 2)$ were chosen based on Proposition 5.2.3 as will be seen.

Proposition 5.2.1. *Let A be a $p \times p$ matrix with nonnegative entries. Then, the following holds:*

$$\begin{aligned} & det\big[M_{(1,2)(i_1,1)(i_2,j_1)}(A)\big]\, det\big[M_{(1,1)(i_2,3)(i_1,j_2)}(A)\big] && (5.25)\\ & \quad - det\big[M_{(1,2)(i_1,1)(i_2,j_2)}(A)\big]\, det\big[M_{(1,1)(i_2,3)(i_1,j_1)}(A)\big] \\ = \;& det\big[M_{(1,2)(i_1,1)(i_2,3)}(A)\big]\, det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)}(A)\big] && (5.26) \end{aligned}$$

Proof. We proceed by induction. For the base case, we consider a 5×5 matrix (because this is the smallest matrix for which the minors in (5.25) exist) of the form

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}.$$

Note that in (5.25) i_1, $i_2 \neq 1$ and j_1, $j_2 \neq 1, 2, 3$, so we choose (without loss of generality) $i_1 = 2$, $i_2 = 3$, $j_1 = 4$, and $j_2 = 5$. Then,

$$\begin{aligned} & \det\big[M_{(1,2)(2,1)(3,4)}(A)\big]\det\big[M_{(1,1)(3,3)(2,5)}(A)\big] \\ & \quad - \det\big[M_{(1,2)(2,1)(3,5)}(A)\big]\det\big[M_{(1,1)(3,3)(2,4)}(A)\big] \\ = \;& (a_{43}a_{55} - a_{45}a_{53})(a_{42}a_{54} - a_{44}a_{52}) - (a_{43}a_{54} - a_{44}a_{53})(a_{42}a_{55} - a_{45}a_{52}) \\ = \;& a_{42}a_{43}a_{54}a_{55} - a_{43}a_{44}a_{52}a_{55} - a_{42}a_{45}a_{53}a_{54} + a_{44}a_{45}a_{52}a_{53} \\ & - (a_{42}a_{43}a_{54}a_{55} - a_{43}a_{45}a_{52}a_{54} - a_{42}a_{44}a_{53}a_{55} + a_{44}a_{45}a_{52}a_{53}) \\ = \;& a_{43}a_{45}a_{52}a_{54} - a_{43}a_{44}a_{52}a_{55} - a_{42}a_{45}a_{53}a_{54} + a_{42}a_{44}a_{53}a_{55} \\ = \;& (a_{44}a_{55} - a_{45}a_{54})(a_{42}a_{53} - a_{43}a_{52}) \\ = \;& \det\big[M_{(1,2)(2,1)(3,3)}(A)\big]\det\big[M_{(1,1)(2,4)(3,5)}(A)\big] \end{aligned}$$

Now, assume that the proposition holds for an arbitrary $k-1 \times k-1$ matrix with nonnegative entries, and consider a $k \times k$ matrix. We begin by writing each of the determinants in (5.25) in terms of its Laplace expansion along the l^{th} row where $l \neq 1$, $l \neq i_1$, and $l \neq i_2$. We also assume, without loss of generality, that k and j_1 are even, and j_2 is odd (this is to determine the signs in the definition of the determinant), $1 < i_1 < i_2 < k$, and $1 < j_1 < j_2 < k$.

$$\det\big[M_{(1,2)(i_1,1)(i_2,j_1)}(A)\big]\det\big[M_{(1,1)(i_2,3)(i_1,j_2)}(A)\big] \tag{5.27}$$
$$- \det\big[M_{(1,2)(i_1,1)(i_2,j_2)}(A)\big]\det\big[M_{(1,1)(i_2,3)(i_1,j_1)}(A)\big]$$
$$= \Big[\Big(a_{l,3}\det\big[M_{(1,2)(i_1,1)(i_2,j_1)(l,3)}(A)\big] - \ldots + a_{l,j_1-1}\det\big[M_{(1,2)(i_1,1)(i_2,j_1)(l,j_1-1)}(A)\big] \tag{5.28}$$
$$- a_{l,j_1+1}\det\big[M_{(1,2)(i_1,1)(i_2,j_1)(l,j_1+1)}(A)\big] + \ldots + a_{l,k}\det\big[M_{(1,2)(i_1,1)(i_2,j_1)(l,k)}(A)\big]\Big)$$
$$* \Big(a_{l,2}\det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,2)}(A)\big] - a_{l,4}\det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,4)}(A)\big]$$
$$+ \ldots - a_{l,j_2-1}\det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,j_2-1)}(A)\big] + a_{l,j_2+1}\det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,j_2+1)}(A)\big]$$
$$- \ldots + a_{l,k}\det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,k)}(A)\big]\Big)\Big]$$
$$- \Big[\Big(a_{l,3}\det\big[M_{(1,2)(i_1,1)(i_2,j_2)(l,3)}(A)\big] - \ldots - a_{l,j_2-1}\det\big[M_{(1,2)(i_1,1)(i_2,j_2)(l,j_2-1)}(A)\big]$$
$$+ a_{l,j_2+1}\det\big[M_{(1,2)(i_1,1)(i_2,j_2)(l,j_2+1)}(A)\big] - \ldots + a_{l,k}\det\big[M_{(1,2)(i_1,1)(i_2,j_2)(l,k)}(A)\big]\Big)$$
$$* \Big(a_{l,2}\det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,2)}(A)\big] - a_{l,4}\det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,4)}(A)\big]$$
$$+ \ldots + a_{l,j_1-1}\det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,j_1-1)}(A)\big] - a_{l,j_1+1}\det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,j_1+1)}(A)\big]$$
$$+ \ldots + a_{l,k}\det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,k)}(A)\big]\Big)\Big]$$

Notice that, once multiplied out, the number of terms in (5.28) is $2(k-3)^2$. Now, to begin the simplification of (5.28), note that there are four terms that cancel:

$$a_{l,3}a_{l,2}\det\big[M_{(1,2)(i_1,1)(i_2,j_1)(l,3)}(A)\big]\det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,2)}(A)\big]$$
$$- a_{l,3}a_{l,2}\det\big[M_{(1,2)(i_1,1)(i_2,j_2)(l,3)}(A)\big]\det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,2)}(A)\big]$$

and

$$
\begin{aligned}
&- a_{l,j_2} a_{l,j_1} \det\big[M_{(1,2)(i_1,1)(i_2,j_1)(l,j_2)}(A)\big] \det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,j_1)}(A)\big] \\
&\quad - \Big(- a_{l,j_1} a_{l,j_2} \det\big[M_{(1,2)(i_1,1)(i_2,j_2)(l,j_1)}(A)\big] \det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,j_2)}(A)\big]\Big).
\end{aligned}
$$

Next, we have four terms that we can rewrite into more convenient forms:

$$
\begin{aligned}
&-\Big(- a_{l,j_1} a_{l,2} \det\big[M_{(1,2)(i_1,1)(i_2,j_2)(l,j_1)}(A)\big] \det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,2)}(A)\big]\Big) \\
&\quad = \quad a_{l,j_1} a_{l,2} \det\big[M_{(1,2)(i_1,1)(i_2,3)(l,j_1)}(A)\big] \det\big[M_{(1,1)(i_2,j_2)(i_1,j_1)(l,2)}(A)\big],
\end{aligned}
$$

$$
\begin{aligned}
&- a_{l,j_2} a_{l,2} \det\big[M_{(1,2)(i_1,1)(i_2,j_1)(l,j_2)}(A)\big] \det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,2)}(A)\big] \\
&\quad = \quad - a_{l,j_2} a_{l,2} \det\big[M_{(1,2)(i_1,1)(i_2,3)(l,j_2)}(A)\big] \det\big[M_{(1,1)(i_2,j_1)(i_1,j_2)(l,2)}(A)\big],
\end{aligned}
$$

$$
\begin{aligned}
&a_{l,3} a_{l,j_2} \det\big[M_{(1,2)(i_1,1)(i_2,j_2)(l,3)}(A)\big] \det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,j_2)}(A)\big] \\
&\quad = \quad a_{l,3} a_{l,j_2} \det\big[M_{(1,2)(i_1,1)(i_2,3)(l,j_2)}(A)\big] \det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)(l,3)}(A)\big],
\end{aligned}
$$

and

$$
\begin{aligned}
&- a_{l,3} a_{l,j_1} \det\big[M_{(1,2)(i_1,1)(i_2,j_1)(l,3)}(A)\big] \det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,j_1)}(A)\big] \\
&\quad = \quad - a_{l,3} a_{l,j_1} \det\big[M_{(1,2)(i_1,1)(i_2,3)(l,j_1)}(A)\big] \det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)(l,3)}(A)\big].
\end{aligned}
$$

Lastly, each of the remaining terms can be placed into pairs and reduced using induction. We show one case and all the others follow similarly:

$$
\begin{aligned}
&a_{l,k} a_{l,k} \Big(\det\big[M_{(1,2)(i_1,1)(i_2,j_1)(l,k)}(A)\big] \det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,k)}(A)\big] \\
&\quad - \det\big[M_{(1,2)(i_1,1)(i_2,j_2)(l,k)}(A)\big] \det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,k)}(A)\big]\Big)
\end{aligned} \tag{5.29}
$$

Notice that the statement inside the large parentheses in (5.29) is precisely the statement in (5.27) applied to a $k-1 \times k-1$ matrix. Hence, by the inductive hypothesis, we have

$$\begin{aligned} & a_{l,k}a_{l,k}\Big(\det\big[M_{(1,2)(i_1,1)(i_2,j_1)(l,k)}(A)\big]\det\big[M_{(1,1)(i_2,3)(i_1,j_2)(l,k)}(A)\big] \\ & \qquad - \det\big[M_{(1,2)(i_1,1)(i_2,j_2)(l,k)}(A)\big]\det\big[M_{(1,1)(i_2,3)(i_1,j_1)(l,k)}(A)\big]\Big) \\ = \;& a_{l,k}a_{l,k}\det\big[M_{(1,2)(i_1,1)(i_2,3)(l,k)}(A)\big]\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)(l,k)}(A)\big]. \end{aligned}$$

Considering the cancelled terms, the rewritten terms, and the paired terms, we have reduced the number of terms from $2(k-3)^2$ to $(k-3)^2$. It is not difficult to see that the remaining terms can be factored as

$$\begin{aligned} & \Big(a_{l,4}\det\big[M_{(1,2)(i_1,1)(i_2,3)(l,4)}(A)\big] - \ldots + a_{l,k}\det\big[M_{(1,2)(i_1,1)(i_2,3)(l,k)}(A)\big]\Big) \\ * \;& \Big(a_{l,2}\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)(l,2)}(A)\big] - \ldots - a_{l,j_1-1}\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)(l,j_1-1)}(A)\big] \\ & + a_{l,j_1+1}\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)(l,j_1+1)}(A)\big] - \ldots + a_{l,j_2-1}\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)(l,j_2-1)}(A)\big] \\ & - a_{l,j_2+1}\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)(l,j_2+1)}(A)\big] + \ldots + a_{l,k}\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)(l,k)}(A)\big]\Big) \\ = \;& \det\big[M_{(1,2)(i_1,1)(i_2,3)}(A)\big]\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)}(A)\big]. \end{aligned} \tag{5.30}$$

Notice that the factorization in (5.30) indeed contains $(k-3)^2$ terms. This completes the proof. □

Proposition 5.2.2. *Let A be a $p \times p$ matrix with nonnegative entries. Then, the following* *holds:*

$$\begin{aligned}
& det\big[M_{(1,1)(i_1,j_1)}(A)\big]\, det\big[M_{(1,2)(i_2,j_2)}(A)\big] \\
& \quad + det\big[M_{(1,2)(i_1,j_1)}(A)\big]\, det\big[M_{(1,1)(i_2,j_2)}(A)\big] \\
& \quad - det\big[M_{(1,1)(i_1,j_2)}(A)\big]\, det\big[M_{(1,2)(i_2,j_1)}(A)\big] \\
& \quad - det\big[M_{(1,2)(i_1,j_2)}(A)\big]\, det\big[M_{(1,1)(i_2,j_1)}(A)\big] \\
= \quad & det\big[M_{(1,1)}(A)\big]\, det\big[M_{(1,2)(i_1,j_1)(i_2,j_2)}(A)\big] \\
& \quad + det\big[M_{(1,2)}(A)\big]\, det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)}(A)\big]
\end{aligned} \tag{5.31}$$

Proof. As in the previous proof, we assume, without loss of generality, that p and j_1 are even, and j_2 is odd (this is to determine the signs in the definition of the determinant), $1 < i_1 < i_2 < p$, and $1 < j_1 < j_2 < p$. We begin by writing each of the determinants in (5.31) according to its Laplace expansion along the ${i_1}^{th}$ or ${i_2}^{th}$ row, whichever is appropriate.

$$\begin{aligned}
& \det\big[M_{(1,1)(i_1,j_1)}(A)\big] \\
= \quad & a_{i_2,2}\det\big[M_{(1,1)(i_2,2)(i_1,j_1)}(A)\big] - \ldots - a_{i_2,j_1-1}\det\big[M_{(1,1)(i_2,j_1-1)(i_1,j_1)}(A)\big] \\
& + a_{i_2,j_1+1}\det\big[M_{(1,1)(i_2,j_1+1)(i_1,j_1)}(A)\big] - \ldots - a_{i_2,k}\det\big[M_{(1,1)(i_2,k)(i_1,j_1)}(A)\big]
\end{aligned} \tag{5.32}$$

$$\begin{aligned}
& \det\big[M_{(1,2)(i_2,j_2)}(A)\big] \\
= \quad & a_{i_1,1}\det\big[M_{(1,2)(i_1,1)(i_2,j_2)}(A)\big] - a_{i_1,3}\det\big[M_{(1,2)(i_1,3)(i_2,j_2)}(A)\big] \\
& + \ldots + a_{i_1,j_2-1}\det\big[M_{(1,2)(i_1,j_2-1)(i_2,j_2)}(A)\big] - a_{i_1,j_2+1}\det\big[M_{(1,2)(i_1,j_2+1)(i_2,j_2)}(A)\big] \\
& + \ldots - a_{i_1,k}\det\big[M_{(1,2)(i_1,k)(i_2,j_2)}(A)\big]
\end{aligned} \tag{5.33}$$

$$
\begin{aligned}
&\det\bigl[M_{(1,2)(i_1,j_1)}(A)\bigr] \\
&= a_{i_2,1}\det\bigl[M_{(1,2)(i_2,1)(i_1,j_1)}(A)\bigr] - a_{i_2,3}\det\bigl[M_{(1,2)(i_2,3)(i_1,j_1)}(A)\bigr] \\
&\quad + \ldots - a_{i_2,j_1-1}\det\bigl[M_{(1,2)(i_2,j_1-1)(i_1,j_1)}(A)\bigr] + a_{i_2,j_1+1}\det\bigl[M_{(1,2)(i_2,j_1+1)(i_1,j_1)}(A)\bigr] \\
&\quad - \ldots - a_{i_2,k}\det\bigl[M_{(1,2)(i_2,k)(i_1,j_1)}(A)\bigr]
\end{aligned} \tag{5.34}
$$

$$
\begin{aligned}
&\det\bigl[M_{(1,1)(i_2,j_2)}(A)\bigr] \\
&= a_{i_1,2}\det\bigl[M_{(1,1)(i_1,2)(i_2,j_2)}(A)\bigr] - \ldots + a_{i_1,j_2-1}\det\bigl[M_{(1,1)(i_1,j_2-1)(i_2,j_2)}(A)\bigr] \\
&\quad - a_{i_1,j_1+1}\det\bigl[M_{(1,1)(i_1,j_2+1)(i_2,j_2)}(A)\bigr] + \ldots - a_{i_1,k}\det\bigl[M_{(1,1)(i_1,k)(i_2,j_2)}(A)\bigr]
\end{aligned} \tag{5.35}
$$

$$
\begin{aligned}
&\det\bigl[M_{(1,1)(i_1,j_2)}(A)\bigr] \\
&= a_{i_2,2}\det\bigl[M_{(1,1)(i_2,2)(i_1,j_2)}(A)\bigr] - \ldots + a_{i_2,j_2-1}\det\bigl[M_{(1,1)(i_2,j_2-1)(i_1,j_2)}(A)\bigr] \\
&\quad - a_{i_2,j_2+1}\det\bigl[M_{(1,1)(i_2,j_2+1)(i_1,j_2)}(A)\bigr] + \ldots - a_{i_2,k}\det\bigl[M_{(1,1)(i_2,k)(i_1,j_2)}(A)\bigr]
\end{aligned} \tag{5.36}
$$

$$
\begin{aligned}
&\det\bigl[M_{(1,2)(i_2,j_1)}(A)\bigr] \\
&= a_{i_1,1}\det\bigl[M_{(1,2)(i_1,1)(i_2,j_1)}(A)\bigr] - a_{i_1,3}\det\bigl[M_{(1,2)(i_1,3)(i_2,j_1)}(A)\bigr] \\
&\quad + \ldots - a_{i_1,j_1-1}\det\bigl[M_{(1,2)(i_1,j_1-1)(i_2,j_1)}(A)\bigr] + a_{i_1,j_1+1}\det\bigl[M_{(1,2)(i_1,j_1+1)(i_2,j_1)}(A)\bigr] \\
&\quad - \ldots - a_{i_1,k}\det\bigl[M_{(1,2)(i_1,k)(i_2,j_1)}(A)\bigr]
\end{aligned} \tag{5.37}
$$

$$
\begin{aligned}
&\det\bigl[M_{(1,2)(i_1,j_2)}(A)\bigr] \\
&= a_{i_2,1}\det\bigl[M_{(1,2)(i_2,1)(i_1,j_2)}(A)\bigr] - a_{i_2,3}\det\bigl[M_{(1,2)(i_2,3)(i_1,j_2)}(A)\bigr] \\
&\quad + \ldots + a_{i_2,j_2-1}\det\bigl[M_{(1,2)(i_2,j_2-1)(i_1,j_2)}(A)\bigr] - a_{i_2,j_2+1}\det\bigl[M_{(1,2)(i_2,j_2+1)(i_1,j_2)}(A)\bigr] \\
&\quad + \ldots - a_{i_2,k}\det\bigl[M_{(1,2)(i_2,k)(i_1,j_2)}(A)\bigr]
\end{aligned} \tag{5.38}
$$

$$
\begin{aligned}
&\det\big[M_{(1,1)(i_2,j_1)}(A)\big] \\
&= \; a_{i_1,2}\det\big[M_{(1,1)(i_1,2)(i_2,j_1)}(A)\big] - \ldots - a_{i_1,j_1-1}\det\big[M_{(1,1)(i_1,j_1-1)(i_2,j_1)}(A)\big] \\
&\quad + a_{i_1,j_1+1}\det\big[M_{(1,1)(i_1,j_1+1)(i_2,j_1)}(A)\big] - \ldots - a_{i_1,k}\det\big[M_{(1,1)(i_1,k)(i_2,j_1)}(A)\big]
\end{aligned} \tag{5.39}
$$

Note equations (5.32), (5.35), (5.36), and (5.39) contain the term $\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)}(A)\big]$, while equations (5.33), (5.34), (5.37), and (5.38) contain the term $\det\big[M_{(1,2)(i_1,j_1)(i_2,j_2)}(A)\big]$. These terms are important because they need to be factored in order to obtain the result in (5.31). Now, notice that substituting the determinant equations in (5.32) - (5.39) into equation (5.31) yields $4(k-2)^2$ terms. First, we consider $4(k-4)$ terms which cancel. For $j \neq 1, 2, j_1, j_2$, we have the following terms:

$$a_{i_1,j}a_{i_2,j}\det\big[M_{(1,1)(i_2,j)(i_1,j_1)}(A)\big]\det\big[M_{(1,2)(i_1,j)(i_2,j_2)}(A)\big] \tag{5.40}$$

$$a_{i_1,j}a_{i_2,j}\det\big[M_{(1,1)(i_1,j)(i_2,j_2)}(A)\big]\det\big[M_{(1,2)(i_2,j)(i_1,j_1)}(A)\big] \tag{5.41}$$

$$-\, a_{i_1,j}a_{i_2,j}\det\big[M_{(1,1)(i_2,j)(i_1,j_2)}(A)\big]\det\big[M_{(1,2)(i_1,j)(i_2,j_1)}(A)\big] \tag{5.42}$$

$$-\, a_{i_1,j}a_{i_2,j}\det\big[M_{(1,1)(i_1,j)(i_2,j_1)}(A)\big]\det\big[M_{(1,2)(i_2,j)(i_1,j_2)}(A)\big] \tag{5.43}$$

Note that the terms given in (5.40) and (5.43) cancel. Likewise, (5.41) and (5.42) cancel. This leaves $4(k^2 - 5k + 8)$ terms. Without loss of generality, we focus only on factoring the term $\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)}(A)\big]$ and showing that the leftover terms equal $\det\big[M_{(1,2)}(A)\big]$. Factoring $\det\big[M_{(1,2)(i_1,j_1)(i_2,j_2)}(A)\big]$ follows similarly. Because of this, we only need to consider half of the remaining terms: $2(k^2-5k+8)$. Now, we consider all the terms from which $\det\big[M_{(1,2)(i_1,j_1)(i_2,j_2)}(A)\big]$ may be factored directly. There are a total of

$2(k-3)+2(k-2)=4k-10$ of these terms, which are shown below:

$$
\begin{aligned}
&+a_{i_1,1} && \Big(-a_{i_2,j_1}\det\big[M_{(1,2)(i_1,1)(i_2,j_1)}(A)\big]+a_{i_2,j_2}\det\big[M_{(1,2)(i_1,1)(i_2,j_1)}(A)\big]\Big)\\
&-a_{i_1,3} && \Big(a_{i_2,j_1}\det\big[M_{(1,2)(i_1,3)(i_2,j_1)}(A)\big]-a_{i_2,j_2}\det\big[M_{(1,2)(i_1,3)(i_2,j_1)}(A)\big]\Big)\\
&\vdots && \\
&+a_{1_1,j_2} && \Big(a_{i_2,1}\det\big[M_{(1,2)(i_1,j_1)(i_2,1)}(A)\big]-a_{i_2,3}\det\big[M_{(1,2)(i_1,j_1)(i_2,3)}(A)\big]\\
& && +\ldots-a_{i_2,j_1-1}\det\big[M_{(1,2)(i_1,j_1)(i_2,j_1-1)}(A)\big]+a_{i_2,j_1+1}\det\big[M_{(1,2)(i_1,j_1)(i_2,j_1+1)}(A)\big]\\
& && -\ldots-a_{i_2,k}\det\big[M_{(1,2)(i_1,j_1)(i_2,k)}(A)\big]\Big)\\
&\vdots && \\
&-a_{i_1,j_2} && \Big(a_{i_2,1}\det\big[M_{(1,2)(i_1,j_2)(i_2,1)}(A)\big]-a_{i_2,3}\det\big[M_{(1,2)(i_1,j_2)(i_2,3)}(A)\big]\\
& && +\ldots+a_{i_2,j_2-1}\det\big[M_{(1,2)(i_1,j_2)(i_2,j_2-1)}(A)\big]-a_{i_2,j_2+1}\det\big[M_{(1,2)(i_1,j_2)(i_2,j_2+1)}(A)\big]\\
& && +\ldots-a_{i_2,k}\det\big[M_{(1,2)(i_1,j_2)(i_2,k)}(A)\big]\Big)\\
&\vdots && \\
&+a_{i_1,k} && \Big(a_{i_2,j_1}\det\big[M_{(1,2)(i_1,k)(i_2,j_1)}(A)\big]-a_{i_2,j_2}\det\big[M_{(1,2)(i_1,k)(i_2,j_2)}(A)\big]\Big)
\end{aligned}
\tag{5.44}
$$

In (5.44), we have grouped the factored terms so that one can see the construction of $\det\big[M_{(1,2)}(A)\big]$. To fill in the remaining terms in (5.44), we use Proposition 5.2.1. We show one case and the rest follow similarly. Notice that in (5.44) the first missing term is

$$a_{i_1,1}a_{i_2,3}\det\big[M_{(1,2)(i_1,1)(i_2,3)}(A)\big]\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)}(A)\big]. \tag{5.45}$$

From (5.32) - (5.39), we see that we have the following:

$$
\begin{aligned}
&a_{i_1,1}a_{i_2,3}\det\big[M_{(1,2)(i_1,1)(i_2,j_1)}(A)\big]\det\big[M_{(1,1)(i_2,3)(i_1,j_2)}(A)\big]\\
&\quad-a_{i_1,1}a_{i_2,3}\det\big[M_{(1,2)(i_1,1)(i_2,j_2)}(A)\big]\det\big[M_{(1,1)(i_2,3)(i_1,j_1)}(A)\big].
\end{aligned}
\tag{5.46}
$$

By Proposition 5.2.1, (5.46) is equal to (5.45). Thus, we are able to fill in the missing terms of (5.44), so that (5.44) equals $\det\big[M_{(1,2)}(A)\big]$ which is what we wanted to show. Lastly, we count the number of terms to which we apply Proposition 5.2.1. To do this, consider (5.33) multiplied by (5.32). The first term in (5.33), $a_{i_1,1}\det\big[M_{(1,2)(i_1,1)(i_2,j_2)}(A)\big]$ is multiplied by $k-3$ terms in (5.32) (all the terms except $a_{i_2,j_2}\det\big[M_{(1,1)(i_2,j_2)(i_1,j_1)}(A)\big]$ because we previously accounted for all terms that could be factored directly). The remaining $k-4$ terms in (5.33) (we leave out the term $a_{i_1,j_1}\det\big[M_{(1,2)(i_1,j_1)(i_2,j_2)}(A)\big]$ because those terms are used in the other half of the proof which we inferred) are multiplied by $k-4$ terms in (5.32), the same ones as the first term except now also omitting the terms we already counted in the cancellation argument. This gives a total of $(k-3)+(k-4)^2 = k^2-7k+13$ terms. Of course, each of these terms has a "matching" term from the product of (5.36) and (5.37) which results in Proposition 5.2.1 being applied to $2k^2-14k+26$ terms. Notice that the number of terms factored directly, $4k-10$, plus $2k^2-14k+26$ gives a total of $2k^2-10k+16 = 2(k^2-5k+8)$ terms accounted for which is the number we desired. This completes the proof. $\square$

Proposition 5.2.3. *Let A be a $p \times p$ matrix with nonnegative entries. Then, the following holds:*

$$\begin{aligned} & det\big[M_{(i_1,j_1)}(A)\big]\,det\big[M_{(i_2,j_2)}(A)\big] \\ & \qquad - det\big[M_{(i_1,j_2)}(A)\big]\,det\big[M_{(i_2,j_1)}(A)\big] \\ = & \quad det\big[M_{(i_1,j_1)(i_2,j_2)}(A)\big]\,det(A) \end{aligned} \tag{5.47}$$

Proof. We proceed by induction. For the base case, we consider a 3×3 matrix of the form

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

Without loss of generality, we choose $i_1 = j_1 = 1$ and $i_2 = j_2 = 2$. Then,

$$
\begin{aligned}
&\det\bigl[M_{(1,1)}(A)\bigr]\det\bigl[M_{(2,2)}(A)\bigr] - \det\bigl[M_{(1,2)}(A)\bigr]\det\bigl[M_{(2,1)}(A)\bigr] \\
&\quad = \quad (a_{22}a_{33} - a_{23}a_{32})(a_{11}a_{33} - a_{13}a_{31}) - (a_{21}a_{33}a_{23}a_{31})(a_{12}a_{33} - a_{13}a_{32}) \\
&\quad = \quad a_{11}a_{22}a_{33}^2 - a_{13}a_{22}a_{31}a_{33} - a_{11}a_{23}a_{32}a_{33} + a_{13}a_{23}a_{31}a_{32} \\
&\qquad\quad -(a_{12}a_{21}a_{33}^2 - a_{13}a_{21}a_{32}a_{33} - a_{12}a_{23}a_{31}a_{33} + a_{13}a_{23}a_{31}a_{32}) \\
&\quad = \quad a_{11}a_{22}a_{33}^2 - a_{13}a_{22}a_{31}a_{33} - a_{11}a_{23}a_{32}a_{33} \\
&\qquad\quad - a_{12}a_{21}a_{33}^2 + a_{13}a_{21}a_{32}a_{33} + a_{12}a_{23}a_{31}a_{33} \\
&\quad = \quad a_{33}\bigl[a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})\bigr] \\
&\quad = \quad \det\bigl[M_{(1,1)(2,2)}(A)\bigr]\det(A)
\end{aligned}
$$

Now, assume that the proposition holds for an arbitrary $k-1 \times k-1$ matrix with nonnegative entries, and consider a $k \times k$ matrix. We begin by writing each of the determinants in (5.47) in terms of its Laplace expansion along the first row, assuming, without loss of generality, that k and j_1 are even, and j_1 is odd (this is to determine the signs in the definition of the determinant), $1 < i_1 < i_2 < k$, and $1 < j_1 < j_2 < k$.

$$
\begin{aligned}
&\det\bigl[M_{(i_1,j_1)}(A)\bigr] \\
&\quad = \quad a_{11}\det\bigl[M_{(1,1)(i_1,j_1)}(A)\bigr] - \ldots + a_{1,j_1-1}\det\bigl[M_{(1,j_1-1)(i_1,j_1)}(A)\bigr] \\
&\qquad\quad - a_{1,j_1+1}\det\bigl[M_{(1,j_1+1)(i_1,j_1)}(A)\bigr] + \ldots + a_{1,k}\det\bigl[M_{(1,k)(i_1,j_1)}(A)\bigr]
\end{aligned}
\tag{5.48}
$$

$$
\begin{aligned}
&\det\bigl[M_{(i_2,j_2)}(A)\bigr] \\
&\quad = \quad a_{11}\det\bigl[M_{(1,1)(i_2,j_2)}(A)\bigr] - \ldots - a_{1,j_2-1}\det\bigl[M_{(1,j_2-1)(i_2,j_2)}(A)\bigr] \\
&\qquad\quad + a_{1,j_2+1}\det\bigl[M_{(1,j_2+1)(i_2,j_2)}(A)\bigr] - \ldots + a_{1,k}\det\bigl[M_{(1,k)(i_2,j_2)}(A)\bigr]
\end{aligned}
\tag{5.49}
$$

$$\begin{aligned}
&\det\bigl[M_{(i_1,j_2)}(A)\bigr] \qquad (5.50)\\
&= a_{11}\det\bigl[M_{(1,1)(i_1,j_2)}(A)\bigr] - \ldots - a_{1,j_2-1}\det\bigl[M_{(1,j_2-1)(i_1,j_2)}(A)\bigr]\\
&\quad + a_{1,j_2+1}\det\bigl[M_{(1,j_2+1)(i_1,j_2)}(A)\bigr] - \ldots + a_{1,k}\det\bigl[M_{(1,k)(i_1,j_2)}(A)\bigr]
\end{aligned}$$

$$\begin{aligned}
&\det\bigl[M_{(i_2,j_1)}(A)\bigr] \qquad (5.51)\\
&= a_{11}\det\bigl[M_{(1,1)(i_2,j_1)}(A)\bigr] - \ldots + a_{1,j_1-1}\det\bigl[M_{(1,j_1-1)(i_2,j_1)}(A)\bigr]\\
&\quad - a_{1,j_1+1}\det\bigl[M_{(1,j_1+1)(i_2,j_1)}(A)\bigr] + \ldots + a_{1,k}\det\bigl[M_{(1,k)(i_2,j_1)}(A)\bigr]
\end{aligned}$$

Notice that substituting equations (5.48) - (5.51) into (5.47) yields a total of $2(k-1)^2$ terms. Two of these terms cancel:

$$\begin{aligned}
&a_{1,j_2}a_{1,j_1}\det\bigl[M_{(1,j_2)(i_1,j_1)}(A)\bigr]\det\bigl[M_{(1,j_1)(i_2,j_2)}(A)\bigr]\\
&\quad - a_{1,j_1}a_{1,j_2}\det\bigl[M_{(1,j_1)(i_1,j_2)}(A)\bigr]\det\bigl[M_{(1,j_2)(i_2,j_1)}(A)\bigr].
\end{aligned}$$

Next, we apply the inductive hypothesis to several groupings of terms. First, all the terms with squared coefficients may be reduced using induction. For instance,

$$\begin{aligned}
&a_{1,1}^2\Bigl(\det\bigl[M_{(1,1)(i_1,j_1)}(A)\bigr]\det\bigl[M_{(1,1)(i_2,j_2)}(A)\bigr]\\
&\qquad - \det\bigl[M_{(1,1)(i_1,j_2)}(A)\bigr]\det\bigl[M_{(1,1)(i_2,j_1)}(A)\bigr]\Bigr)\\
&= a_{1,1}^2\Bigl(\det\bigl[M_{(1,1)(i_1,j_1)(i_2,j_2)}(A)\bigr]\det\bigl[M_{(1,1)}(A)\bigr]\Bigr).
\end{aligned}$$

There are a total of $2(k-2)$ of these terms which are reduced to $k-2$ terms after induction. Two other groupings to which we can apply induction are terms having the coefficients

$a_{1,j_1}a_{1,j}$ or $a_{1,j}a_{1,j_2}$ where $j \neq j_1, j_2$. For instance,

$$\begin{aligned}
&-a_{1,1}a_{1,j_1}\Big(\det\big[M_{(1,1)(i_1,j_1)}(A)\big]\det\big[M_{(1,j_1)(i_2,j_2)}(A)\big] \\
&\qquad - \det\big[M_{(1,j_1)(i_1,j_2)}(A)\big]\det\big[M_{(1,1)(i_2,j_1)}(A)\big]\Big) \\
&= \quad -a_{1,1}a_{1,j_1}\Big(\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)}(A)\big]\det\big[M_{(1,j_1)}(A)\big]\Big).
\end{aligned}$$

There are a total of $4(k-2)$ terms of this types which are reduced to $2(k-2)$ terms through induction. Finally, we consider the remaining mixed terms. All of these terms can be put into groupings of four terms each and reduced using Proposition 5.2.2. For example, consider the terms with the coefficient $a_{11}a_{12}$:

$$\begin{aligned}
&-a_{11}a_{12}\Big(\det\big[M_{(1,1)(i_1,j_1)}(A)\big]\det\big[M_{(1,2)(i_2,j_2)}(A)\big] + \det\big[M_{(1,2)(i_1,j_1)}(A)\big]\det\big[M_{(1,1)(i_2,j_2)}(A)\big] \\
&\qquad - \det\big[M_{(1,1)(i_1,j_2)}(A)\big]\det\big[M_{(1,2)(i_2,j_1)}(A)\big] - \det\big[M_{(1,2)(i_1,j_2)}(A)\big]\det\big[M_{(1,1)(i_2,j_1)}(A)\big]\Big) \\
&= \quad -a_{11}a_{12}\Big(\det\big[M_{(1,1)}(A)\big]\det\big[M_{(1,2)(i_1,j_1)(i_2,j_2)}(A)\big] \\
&\qquad + \det\big[M_{(1,2)}(A)\big]\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)}(A)\big]\Big).
\end{aligned}$$

To determine the number of terms of this type, consider the first term in equation (5.48), $a_{11}\det\big[M_{(1,1)(i_1,j_1)}(A)\big]$. This term is multiplied by all the terms in (5.49) except for the terms $a_{11}\det\big[M_{(1,1)(i_2,j_2)}(A)\big]$ and $a_{1,j_1}\det\big[M_{(1,j_1)(i_2,j_2)}(A)\big]$ because we have already accounted for these terms. Similarly, every term in (5.48) is multiplied by $k-3$ terms in (5.49), except the term $a_{1,j_2}\det\big[M_{(1,j_2)(i_1,j_1)}(A)\big]$. So taking into account all four equations (5.48) - (5.51), there are a total of $2(k-2)(k-3)$ mixed terms which are reduced by half to $(k-2)(k-3)$ terms using Proposition 5.2.2. Now, notice that we have accounted for all the terms we began with because $2+2(k-2)+4(k-2)+2(k-2)(k-3) = 2k^2-4k+2 = 2(k-1)^2$.

It is not difficult to see that the remaining rewritten terms can be factored as follows:

$$\begin{aligned}
&\Big(a_{11}\det\big[M_{(1,1)}(A)\big] - a_{12}\det\big[M_{(1,2)}(A)\big] + \ldots - a_{l,k}\det\big[M_{(1,k)}(A)\big]\Big) \\
&\quad * \quad \Big(a_{11}\det\big[M_{(1,1)(i_1,j_1)(i_2,j_2)}(A)\big] - a_{12}\det\big[M_{(1,2)(i_1,j_1)(i_2,j_2)}(A)\big] \\
&\qquad + \ldots + a_{1,j_1-1}\det\big[M_{(1,j_1-1)(i_1,j_1)(i_2,j_2)}(A)\big] - a_{1,j_1+1}\det\big[M_{(1,j_1+1)(i_1,j_1)(i_2,j_2)}(A)\big] \\
&\qquad + \ldots + a_{1,j_2-1}\det\big[M_{(1,j_2-1)(i_1,j_1)(i_2,j_2)}(A)\big] - a_{1,j_2+1}\det\big[M_{(1,j_2+1)(i_1,j_1)(i_2,j_2)}(A)\big] \\
&\qquad + \ldots - a_{1,k}\det\big[M_{(1,k)(i_1,j_1)(i_2,j_2)}(A)\big]\Big) \\
&\quad = \quad \det(A)\,\det\big[M_{(i_1,j_1)(i_2,j_2)}(A)\big].
\end{aligned} \tag{5.52}$$

Finally, notice that the number of terms which remain is $(k-2)+2(k-2)+(k-2)(k-3) = k(k-2)$ which is precisely the number of terms in equation (5.52). This completes the proof. □

5.3 Multiobjective Approach

5.3.1 Pareto Cone

The formulation of the Constraint Controlled Spacing problem for biobjective problems (5.2) generalizes quite easily to multiobjective problems. The implementation, however, is not as straightforward and is discussed later. As before, let $x^* \in X_{WE}$, δ be the desired spacing between Pareto points, and $||\cdot||$ be an l_p-norm. Then, the Constraint

Controlled Spacing problem for an MOP ($\text{CCS}_1^p(\delta, x^*)$) is:

$$
\begin{aligned}
\text{minimize} \quad & f_1(x) \\
\text{subject to} \quad & f_1(x) \leq f_1(x^p) \\
& \vdots \\
& f_{p-1}(x) \leq f_{p-1}(x^p) \\
& f_2(x) \leq f_2(x^*) \\
& \vdots \\
& f_p(x) \leq f_p(x^*) \\
& ||f(x) - f(x^*)|| \geq \delta \\
& x \in X
\end{aligned}
\tag{5.53}
$$

where

$$x^p \in \arg\min\{f_p(x) : x \in X\}. \tag{5.54}$$

Notice that the problem feasibility check $||f(x^*) - y^2|| \geq \delta$ in the biobjective case does not extend well to higher dimensions because we now have a Pareto surface instead of a two-dimensional curve. Thus, checking the initial feasibility of the problem becomes much more complex. Additionally, we would like to stress again that the first $p-1$ constraints in (5.53) are needed only theoretically and may be omitted in practice.

We now prove that problem (5.53) generates a weak Pareto point.

Theorem 5.3.1. *If Y_{WN} is connected and $\hat{x}$ is the unique optimal solution of $CCS_1^p(\delta, x^*)$, then $f(\hat{x}) \in Y_{WN}$.*

Proof. Assume Y_{WN} is connected and let $\hat{x}$ be the unique optimal solution of $\text{CCS}_1^p(\delta, x^*)$. Let $\hat{y} = f(\hat{x})$. Suppose that $\hat{y} \notin Y_{WN}$. By definition, $\hat{y} \notin Y_{WN}$ implies $(\hat{y} - \text{int}\ \mathbb{R}^p_{\geqq}) \cap Y \neq \emptyset$. This further implies that $(\hat{y} - \mathbb{R}^p_{\geqq}) \cap Y \neq \emptyset$. In particular, since $y^* = f(x^*)$ and $y^p = f(x^p)$ are both in Y_{WN}, Y_{WN} is connected, and $f_i(\hat{x}) \leq f_i(x^p)$ for $i = 1, \ldots, p-1$, there must exist a $\lambda > 0$ and $\tilde{y} = f(\tilde{x}) \in Y$ such that $\tilde{y} = \hat{y} + \lambda(0, \ldots, 0, -1)^T$. Then, we have that

$$\tilde{y}_i = \hat{y}_i, \text{for } i = 1, \ldots p-1, \tag{5.55}$$

and

$$\tilde{y}_p < \hat{y}_p. \tag{5.56}$$

To establish the feasibility of $\tilde{x}$, recall that since $\hat{x}$ is optimal for $\text{CCS}_1^p(\delta,\, x^*)$, we have

$$\hat{y}_i \leq y_i^p, \text{for } i = 1, \ldots, p-1, \tag{5.57}$$

$$\hat{y}_i \leq y_i^*, \text{for } i = 2, \ldots, p, \tag{5.58}$$

and

$$||\hat{y} - y^*|| \geq \delta. \tag{5.59}$$

Combining (5.55) - (5.58) gives

$$\tilde{y}_i \leq y_i^p, \text{for } i = 1, \ldots, p-1,$$

and

$$\tilde{y}_i \leq y_i^*, \text{for } i = 2, \ldots, p,$$

so $\tilde{x}$ is feasible to the first $2(p-1)$ constraints in (5.53). Further, combining (5.55), (5.56), and (5.59) with $\hat{y}_p \leq y_p^*$ from (5.58) gives that $||\tilde{y} - y^*|| > \delta$. Thus, $\tilde{x}$ is feasible to $\text{CCS}_1^p(\delta,\, x^*)$ with $\tilde{y}_1 = \hat{y}_1$ which implies that $\hat{x}$ is not the unique optimal solution to $\text{CCS}_1^p(\delta,\, x^*)$. This is a contradiction, so $f(\hat{x}) \in Y_{WN}$. This completes the proof. $\square$

5.3.2 General Cones

As in the biobjective case, the CCS problem (5.53) can be generalized to problems where optimality is defined by an arbitrary obtuse, simplicial preference cone, C, by using the results of Proposition 2.0.8. Let $C \supseteq \mathbb{R}^p_{\geqq}$ be defined by the matrix A and let $g(x) = Af(x)$. Given that $x^* \in E_W(X, f, C)$, then $\text{CCS}_1^p(\delta,\, x^*,\, A)$ is defined as:

$$\begin{array}{ll} \text{minimize} & g_1(x) \\ \text{subject to} & f_1(x) \leq f_1(x^p) \\ & \vdots \\ & f_{p-1}(x) \leq f_{p-1}(x^p) \\ & g_2(x) \leq g_2(x^*) \\ & \vdots \\ & g_p(x) \leq g_p(x^*) \\ & ||f(x) - f(x^*)|| \geq \delta \\ & x \in X \end{array} \tag{5.60}$$

where

$$x^p \in \arg\min\{g_p(x) : x \in X\}.$$

Before proving that problem (5.60) is valid under certain assumptions about the matrix A, we present a result which generalizes Proposition 5.1.2 to more than two dimensions.

Proposition 5.3.2. *Let $C \supseteq \mathbb{R}^p_{\geqq}$ be a simplicial cone defined by $C = \{d \in \mathbb{R}^p : Ad \geqq 0\}$. Then, A can be written as a positive definite matrix using only row exchanges.*

Proof. We proceed by induction. For the base case, we consider a 3×3 matrix A of the form

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

and show that A can be written as a positive definite matrix. Without loss of generality, we assume that the determinant of A is positive. Otherwise, we can exchange two of the rows to make the determinant positive. Now, notice that we may perform an even number of row exchanges without affecting the determinant of A. For the 3×3 case, this leads to two additional candidate matrices which represent the same cone, C, as the original matrix A:

$$A' = \begin{bmatrix} a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{11} & a_{12} & a_{13} \end{bmatrix} \text{ and } A'' = \begin{bmatrix} a_{31} & a_{32} & a_{33} \\ a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}.$$

We must show that at least one of A, A', and A'' is positive definite. To do this, we need only check the signs of the determinants of the second order leading principal minors. Suppose that the determinants of the second order leading principal minors of A, A', and A'' are all nonpositive. That is, suppose that

$$\begin{aligned} &a_{11}a_{22} - a_{12}a_{21} \leq 0, \\ &a_{21}a_{32} - a_{22}a_{31} \leq 0, \text{ and} \\ &a_{12}a_{31} - a_{11}a_{32} \leq 0. \end{aligned} \tag{5.61}$$

However, consider the determinant of A:

$$\det(A) = a_{13}(a_{21}a_{32} - a_{22}a_{31}) - a_{23}(a_{11}a_{32} - a_{12}a_{31}) + a_{33}(a_{11}a_{22} - a_{12}a_{21}).$$

If the inequalities in (5.61) hold, then $\det(A) \leq 0$ which is a contradiction. Thus, at least one of A, A', and A'' is positive definite. Notice that we do not need to consider the case when the determinant of the first order leading principal minor is 0 because this would cause the determinant of the second order leading principal minor to be less than or equal to 0. Now, we assume that the proposition holds for $k-1 \times k-1$ matrices and consider the case of a $k \times k$ matrix A of the form:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix}.$$

Again, without loss of generality, we assume that the determinant of A is positive. In this case, we must only show that we can obtain a matrix A' from A with an even number of row exchanges in which the determinant of the $(k-1)^{st}$ leading principal minor is positive. Then, by induction, we know that we are able to reorder the first $k-1$ rows in an even number of exchanges to obtain a positive definite $(k-1)^{st}$ leading principal minor. First, consider the determinant of A. There are two cases: k is odd (5.62) or k is even (5.63).

$$\det(A) = \tag{5.62}$$
$$a_{1k} \begin{vmatrix} a_{21} & \cdots & a_{2,k-1} \\ a_{31} & \cdots & a_{3,k-1} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{k,k-1} \end{vmatrix} - a_{2k} \begin{vmatrix} a_{11} & \cdots & a_{1,k-1} \\ a_{31} & \cdots & a_{3,k-1} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{k,k-1} \end{vmatrix} + \cdots + a_{kk} \begin{vmatrix} a_{11} & \cdots & a_{1,k-1} \\ a_{21} & \cdots & a_{2,k-1} \\ \vdots & \ddots & \vdots \\ a_{k-1,1} & \cdots & a_{k-1,k-1} \end{vmatrix}$$

$$\det(A) = \tag{5.63}$$
$$-a_{1k} \begin{vmatrix} a_{21} & \cdots & a_{2,k-1} \\ a_{31} & \cdots & a_{3,k-1} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{k,k-1} \end{vmatrix} + a_{2k} \begin{vmatrix} a_{11} & \cdots & a_{1,k-1} \\ a_{31} & \cdots & a_{3,k-1} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{k,k-1} \end{vmatrix} - \cdots + a_{kk} \begin{vmatrix} a_{11} & \cdots & a_{1,k-1} \\ a_{21} & \cdots & a_{2,k-1} \\ \vdots & \ddots & \vdots \\ a_{k-1,1} & \cdots & a_{k-1,k-1} \end{vmatrix}$$

Suppose that the $(k-1)^{st}$ leading principal minor of A has a nonpositive determinant. That is, suppose

$$\begin{vmatrix} a_{11} & \cdots & a_{1,k-1} \\ a_{21} & \cdots & a_{2,k-1} \\ \vdots & \ddots & \vdots \\ a_{k-1,1} & \cdots & a_{k-1,k-1} \end{vmatrix} \leq 0. \tag{5.64}$$

From this, we can determine the signs of the determinants in (5.62) and (5.63) by calculating the number of row exchanges necessary to obtain a specific matrix from the matrix in (5.64). In the odd case, consider the first matrix in equation (5.62). To obtain this matrix from the one in (5.64), we must perform $k-1$ row exchanges. Since k is odd, $k-1$ is even, so the first determinant in (5.62) is nonpositive. Notice, now, that to obtain each consecutive matrix from the one before it in (5.62) requires a single row exchange so that the signs of the determinants alternate between nonpositive and nonnegative. So from equation (5.62) we have

$$a_{1k}\underbrace{\begin{vmatrix} a_{21} & \cdots & a_{2,k-1} \\ a_{31} & \cdots & a_{3,k-1} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{k,k-1} \end{vmatrix}}_{\leq 0} - a_{2k}\underbrace{\begin{vmatrix} a_{11} & \cdots & a_{1,k-1} \\ a_{31} & \cdots & a_{3,k-1} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{k,k-1} \end{vmatrix}}_{\geq 0} + \cdots + a_{kk}\underbrace{\begin{vmatrix} a_{11} & \cdots & a_{1,k-1} \\ a_{21} & \cdots & a_{2,k-1} \\ \vdots & \ddots & \vdots \\ a_{k-1,1} & \cdots & a_{k-1,k-1} \end{vmatrix}}_{\leq 0} \leq 0$$

which is a contradiction. The case when k is even is similar except that the first determinant in (5.63) is nonnegative because $k-1$ is odd. This completes the proof. $\square$

Following, we use the results from Section 5.2 and some additional assumptions about the cone C to prove that problem (5.60) generates a weakly nondominated point for a certain class of simplicial cones which is the main result for this section.

Theorem 5.3.3. *Let $C \supseteq \mathbb{R}^p_{\geqq}$ be a simplicial cone defined by $C = \{y \in \mathbb{R}^p : Ay \geqq 0\}$ where A is a positive definite matrix. Additionally, we assume that the following minors of A have* *nonpositive determinants: all minors of the form*

$$\begin{bmatrix} a_{11} & \cdots & a_{1,j} \\ \vdots & \ddots & \vdots \\ a_{j-1,1} & \cdots & a_{j-1,j} \\ a_{i,1} & \cdots & a_{i,j} \end{bmatrix} \tag{5.65}$$

where $j = 2, \ldots, p-1$ and $i = j+1, \ldots, p$. If $N_W(Y, C)$ is connected and $\hat{x}$ is the unique optimal solution of $CCS_1^p(\delta,\ x^*,\ A)$, then $f(\hat{x}) \in N_W(Y, C)$.

Proof. Assume $N_W(Y, C)$ is connected and let $\hat{x}$ be the unique optimal solution of $\mathrm{CCS}_1^p(\delta,\ x^*,\ A)$. Let $\hat{y} = f(\hat{x})$ and $\hat{z} = Af(\hat{x})$. Suppose that $\hat{y} \notin N_W(Y, C)$. By definition, $\hat{y} \notin N_W(Y, C)$ implies $(\hat{y} - \text{int } C) \cap Y \neq \emptyset$. This further implies that $(\hat{y} - C) \cap Y \neq \emptyset$. In particular, since $y^* = f(x^*)$ and $y^p = f(x^p)$ are both in $N_W(Y, C)$, $N_W(Y, C)$ is connected, and $f_i(\hat{x}) \leq f_i(x^p)$ for $i = 1, \ldots, p-1$, there must exist a $\lambda > 0$ and $\tilde{y} = f(\tilde{x}) \in Y$ such that $\tilde{y} = \hat{y} + \lambda(0, \ldots, 0, -1)^T$. Let A be a positive definite matrix of the form

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{bmatrix}.$$

Then, the resulting components of $\tilde{z}$ are as follows:

$$\tilde{z} = A\tilde{y} = A(\hat{y} + \lambda(0, \ldots, 0, -1)^T) = \hat{z} + \lambda A(0, \ldots, 0, -1)^T = \begin{bmatrix} \hat{z}_1 - \lambda a_{1p} \\ \hat{z}_2 - \lambda a_{2p} \\ \vdots \\ \hat{z}_p - \lambda a_{pp} \end{bmatrix}.$$

Note that since C is obtuse, by Proposition 5.1.1, all components of the matrix A are non-negative. Combining this fact with $\lambda > 0$ gives that

$$\tilde{z}_1 \leq \hat{z}_1, \tag{5.66}$$

and

$$\tilde{z}_i \leq \hat{z}_i, \text{for } i = 2, \ldots, p. \tag{5.67}$$

To establish the feasibility of $\tilde{x}$, recall that since $\hat{x}$ is optimal for $\text{CCS}_1^p(\delta,\, x^*,\, A)$, we have

$$\hat{y}_i \leq y_i^p, \text{for } i = 1, \ldots, p-1, \tag{5.68}$$

$$\hat{z}_i \leq z_i^*, \text{for } i = 2, \ldots, p, \tag{5.69}$$

and

$$||\hat{y} - y^*|| \geq \delta. \tag{5.70}$$

Combining (5.67) and (5.69) gives $\tilde{z}_i \leq z_i^*, \text{for } i = 2, \ldots, p$, so $\tilde{x}$ is feasible to the second set of $p-1$ constraints in (5.60). Further, since $\tilde{y} = \hat{y} + \lambda(0, \ldots, 0, -1)^T$, we know that

$$\tilde{y}_i = \hat{y}_i, \text{for } i = 1, \ldots, p-1, \tag{5.71}$$

and

$$\tilde{y}_p < \hat{y}_p. \tag{5.72}$$

Combining (5.68) and (5.71) gives that $\tilde{y}_i \leq y_i^p$ for $i = 1, \ldots, p-1$, so $\tilde{x}$ is feasible to the first set of $p-1$ constraints in (5.60). Now, to complete the feasibility argument, we determine the relationship between $\hat{y}_p$ and y_p^*. From (5.69) we can derive the following where A_i denotes the i^{th} row of A:

$$A_i\hat{y} \quad \leq \quad A_i y^*, \text{for } i = 2, \ldots, p. \tag{5.73}$$

Note that since $y^* \in N_W(Y,C)$, then $Ay^* \in AN_W(Y,C)$ which we may rewrite as $z^* \in AN_W(Y,C)$. By Proposition 2.0.8, $AN_W(Y,C) = N_W(AY, \mathbb{R}^p_>)$. Thus, $z^* \in N_W(AY, \mathbb{R}^p_>)$ which by definition means that $(z^* - \mathbb{R}^p_>) \cap AY = \emptyset$. In particular, given (5.69), this implies that we must have $\hat{z}_1 \geq z_1^*$ which gives

$$A_1\hat{y} \quad \geq \quad A_1 y^*. \tag{5.74}$$

Combining (5.73) and (5.74), we have:

$$\begin{aligned} A_1(\hat{y} - y^*) \quad &\geq \quad 0 \\ A_i(\hat{y} - y^*) \quad &\leq \quad 0, \text{for } i = 2, \ldots, p. \end{aligned} \tag{5.75}$$

Next, we rewrite the inequalities in (5.75) as:

$$\begin{aligned} a_{11}(\hat{y}_1 - y_1^*) + a_{12}(\hat{y}_2 - y_2^*) + \cdots + a_{1p}(\hat{y}_p - y_p^*) \quad &\geq \quad 0 \\ a_{21}(\hat{y}_1 - y_1^*) + a_{22}(\hat{y}_2 - y_2^*) + \cdots + a_{2p}(\hat{y}_p - y_p^*) \quad &\leq \quad 0 \\ a_{31}(\hat{y}_1 - y_1^*) + a_{32}(\hat{y}_2 - y_2^*) + \cdots + a_{3p}(\hat{y}_p - y_p^*) \quad &\leq \quad 0 \\ \vdots \qquad\qquad\qquad & \qquad\; \vdots \\ a_{p1}(\hat{y}_1 - y_1^*) + a_{p2}(\hat{y}_2 - y_2^*) + \cdots + a_{pp}(\hat{y}_p - y_p^*) \quad &\leq \quad 0. \end{aligned} \tag{5.76}$$

Solving all of the inequalities in (5.76) for $(\hat{y}_1 - y_1^*)$ gives:

$$\begin{aligned} (\hat{y}_1 - y_1^*) \quad &\geq \quad -\tfrac{a_{12}}{a_{11}}(\hat{y}_2 - y_2^*) - \cdots - \tfrac{a_{1p}}{a_{11}}(\hat{y}_p - y_p^*) \\ (\hat{y}_1 - y_1^*) \quad &\leq \quad -\tfrac{a_{22}}{a_{21}}(\hat{y}_2 - y_2^*) - \cdots - \tfrac{a_{2p}}{a_{21}}(\hat{y}_p - y_p^*) \\ (\hat{y}_1 - y_1^*) \quad &\leq \quad -\tfrac{a_{32}}{a_{31}}(\hat{y}_2 - y_2^*) - \cdots - \tfrac{a_{3p}}{a_{31}}(\hat{y}_p - y_p^*) \\ \vdots \qquad & \qquad\qquad\qquad \vdots \\ (\hat{y}_1 - y_1^*) \quad &\leq \quad -\tfrac{a_{p2}}{a_{p1}}(\hat{y}_2 - y_2^*) - \cdots - \tfrac{a_{pp}}{a_{p1}}(\hat{y}_p - y_p^*). \end{aligned} \tag{5.77}$$

Substituting the first inequality in (5.77) into the other $p-1$ inequalities, we can eliminate

$(\hat{y}_1 - y_1^*)$:

$$\begin{array}{rcl} -\frac{a_{12}}{a_{11}}(\hat{y}_2 - y_2^*) - \cdots - \frac{a_{1p}}{a_{11}}(\hat{y}_p - y_p^*) & \leq & -\frac{a_{22}}{a_{21}}(\hat{y}_2 - y_2^*) - \cdots - \frac{a_{2p}}{a_{21}}(\hat{y}_p - y_p^*) \\ -\frac{a_{12}}{a_{11}}(\hat{y}_2 - y_2^*) - \cdots - \frac{a_{1p}}{a_{11}}(\hat{y}_p - y_p^*) & \leq & -\frac{a_{32}}{a_{31}}(\hat{y}_2 - y_2^*) - \cdots - \frac{a_{3p}}{a_{31}}(\hat{y}_p - y_p^*) \\ \vdots & & \vdots \\ -\frac{a_{12}}{a_{11}}(\hat{y}_2 - y_2^*) - \cdots - \frac{a_{1p}}{a_{11}}(\hat{y}_p - y_p^*)) & \leq & -\frac{a_{p2}}{a_{p1}}(\hat{y}_2 - y_2^*) - \cdots - \frac{a_{pp}}{a_{p1}}(\hat{y}_p - y_p^*). \end{array} \tag{5.78}$$

Rewriting the inequalities in (5.78), we see determinants of minors of the matrix A emerge:

$$\begin{array}{rcl} (a_{11}a_{22} - a_{12}a_{21})(\hat{y}_2 - y_2^*) & \leq & (a_{13}a_{21} - a_{11}a_{23})(\hat{y}_3 - y_3^*) + \cdots + (a_{1p}a_{21} - a_{11}a_{2p})(\hat{y}_p - y_p^*) \\ (a_{11}a_{32} - a_{12}a_{31})(\hat{y}_2 - y_2^*) & \leq & (a_{13}a_{31} - a_{11}a_{33})(\hat{y}_3 - y_3^*) + \cdots + (a_{1p}a_{31} - a_{11}a_{3p})(\hat{y}_p - y_p^*) \\ \vdots & & \vdots \\ (a_{11}a_{p2} - a_{12}a_{p1})(\hat{y}_2 - y_2^*) & \leq & (a_{13}a_{p1} - a_{11}a_{p3})(\hat{y}_3 - y_3^*) + \cdots + (a_{1p}a_{p1} - a_{11}a_{pp})(\hat{y}_p - y_p^*). \end{array} \tag{5.79}$$

Since A is positive definite by Proposition 5.3.2, we know that all of the principal minors of A have positive determinants. Namely, in (5.79),

$$\begin{array}{rcl} (a_{11}a_{22} - a_{12}a_{21}) & > & 0 \\ (a_{11}a_{33} - a_{13}a_{31}) & > & 0 \\ \vdots & & \vdots \\ (a_{11}a_{pp} - a_{1p}a_{p1}) & > & 0. \end{array}$$

So we may rearrange the inequalities in (5.79) as follows:

$$\begin{array}{rcl} (\hat{y}_2 - y_2^*) & \leq & \frac{(a_{13}a_{21} - a_{11}a_{23})}{(a_{11}a_{22} - a_{12}a_{21})}(\hat{y}_3 - y_3^*) + \cdots + \frac{(a_{1p}a_{21} - a_{11}a_{2p})}{(a_{11}a_{22} - a_{12}a_{21})}(\hat{y}_p - y_p^*) \\ (\hat{y}_3 - y_3^*) & \leq & \frac{(a_{12}a_{31} - a_{11}a_{32})}{(a_{11}a_{33} - a_{13}a_{31})}(\hat{y}_2 - y_2^*) + \cdots + \frac{(a_{1p}a_{31} - a_{11}a_{3p})}{(a_{11}a_{33} - a_{13}a_{31})}(\hat{y}_p - y_p^*) \\ \vdots & & \vdots \\ (\hat{y}_p - y_p^*) & \leq & \frac{(a_{12}a_{p1} - a_{11}a_{p2})}{(a_{11}a_{pp} - a_{1p}a_{p1})}(\hat{y}_2 - y_2^*) + \cdots + \frac{(a_{1,p-1}a_{p1} - a_{11}a_{p,p-1})}{(a_{11}a_{pp} - a_{1p}a_{p1})}(\hat{y}_{p-1} - y_{p-1}^*). \end{array} \tag{5.80}$$

Note that in (5.80) all the coefficients of $(\hat{y}_2 - y_2^*)$ are positive: the numerators are the negative determinants of minors having the form of (5.65) and, thus, are positive, and the denominators are principal minors and are positive since A is PD. Next, we substitute the right hand side of the first inequality in (5.80) for $(\hat{y}_2 - y_2^*)$ in all the remaining inequalities and do some rearranging to get:

$$\begin{aligned}
0 \;\leq\; & \big[(a_{12}a_{31} - a_{11}a_{32})(a_{13}a_{21} - a_{11}a_{23}) - (a_{11}a_{22} - a_{12}a_{21})(a_{11}a_{33} - a_{13}a_{31})\big](\hat{y}_3 - y_3^*) \\
& + \cdots + \big[(a_{11}a_{22} - a_{12}a_{21})(a_{1p}a_{31} - a_{11}a_{3p}) + (a_{12}a_{31} - a_{11}a_{32})(a_{1p}a_{21} - a_{11}a_{2p})\big](\hat{y}_p - y_p^*) \\
\vdots \; & \\
0 \;\leq\; & \big[(a_{12}a_{p1} - a_{11}a_{p2})(a_{13}a_{21} - a_{11}a_{23}) + (a_{11}a_{22} - a_{12}a_{21})(a_{13}a_{p1} - a_{11}a_{p3})\big](\hat{y}_3 - y_3^*) \\
& + \cdots + \big[(a_{12}a_{p1} - a_{11}a_{p2})(a_{1p}a_{21} - a_{11}a_{2p}) - (a_{11}a_{22} - a_{12}a_{21})(a_{11}a_{pp} - a_{1p}a_{p1})\big](\hat{y}_p - y_p^*).
\end{aligned}$$

These inequalities can again be rewritten as:

$$\begin{aligned}
0 \;\leq\; & \big[- a_{11}(a_{22}a_{33} - a_{23}a_{32}) + a_{12}(a_{21}a_{33} - a_{23}a_{31}) - a_{13}(a_{21}a_{32} - a_{22}a_{31})\big](\hat{y}_3 - y_3^*) \\
& + \cdots + \big[- a_{11}(a_{22}a_{3p} - a_{2p}a_{32}) + a_{12}(a_{21}a_{3p} - a_{2p}a_{31}) - a_{1p}(a_{21}a_{32} - a_{22}a_{31})\big](\hat{y}_p - y_p^*) \\
\vdots \; & \\
0 \;\leq\; & \big[- a_{11}(a_{22}a_{p3} - a_{23}a_{p2}) + a_{12}(a_{21}a_{p3} - a_{23}a_{p1}) - a_{13}(a_{21}a_{p2} - a_{22}a_{p1})\big](\hat{y}_3 - y_3^*) \\
& + \cdots + \big[- a_{11}(a_{22}a_{pp} - a_{2p}a_{p2}) + a_{12}(a_{21}a_{pp} - a_{2p}a_{p1}) - a_{1p}(a_{21}a_{p2} - a_{22}a_{p1})\big](\hat{y}_p - y_p^*).
\end{aligned} \tag{5.81}$$

Notice that in (5.81) we see the (negative) determinants of third order minors of A emerge and, in particular, that the coefficient of $(\hat{y}_3 - y_3^*)$ in the final inequality of (5.81) is the negative determinant of a minor having the form of (5.65). Continuing in this fashion, eliminating the term $(\hat{y}_i - y_i^*)$ for each $i \neq p$, we arrive at the following inequality in terms of only $(\hat{y}_p - y_p^*)$:

$$\Big(\det\big[M_{(p-1,p-1)}(A)\big]\det\big[M_{(p,p)}(A)\big] - \det\big[M_{(p-1,p)}(A)\big]\det\big[M_{(p,p-1)}(A)\big]\Big)(\hat{y}_p - y_p^*) \leq 0.$$

Then, by Proposition 5.2.3, equation (5.3.2) becomes

$$\det\big[M_{(p-1,p-1)(p,p)}(A)\big]\det(A)(\hat{y}_p - y_p^*) \leq 0. \tag{5.82}$$

Since A is a positive definite by Proposition 5.3.2, we know that $\det\big[M_{(p-1,p-1)(p,p)}(A)\big] > 0$ and $\det(A) > 0$. Hence, we have that $\hat{y}_p \leq y_p^*$. This fact along with (5.70), (5.71), and (5.72) implies that $||f(\tilde{x}) - f(x^*)|| > \delta$. Thus, $\tilde{x}$ is feasible to $\text{CCS}_1^p(\delta,\, x^*,\, A)$ with $\tilde{z}_1 \leq \hat{z}_1$. If $\tilde{z}_1 < \hat{z}_1$, then $\hat{x}$ is not an optimal solution of $\text{CCS}_1^p(\delta,\, x^*,\, A)$. If $\tilde{z}_1 = \hat{z}_1$, then $\hat{x}$ is not a unique optimal solution of $\text{CCS}_1^p(\delta,\, x^*,\, A)$. Hence, both cases lead to contradictions which implies that $f(\hat{x}) \in N_W(Y, C)$. This completes the proof. $\square$

Next, we give some examples to show that there exist positive definite matrices which satisfy the assumptions given in Theorem 5.3.3 while there also exist some that do not. Matrices A and A' shown in (5.83) and matrices A'' and A''' shown in (5.84) are all positive definite.

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 1 & 2 & 3 \\ 3 & 1 & 5 \end{bmatrix},\ A' = \begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 4 \\ 1 & 2 & 3 \end{bmatrix} \tag{5.83}$$

$$A'' = \begin{bmatrix} 5 & 2 & 0 & 1 \\ 2 & 1 & 3 & 4 \\ 4 & 1 & 4 & 2 \\ 5 & 1 & 0 & 4 \end{bmatrix},\ A''' = \begin{bmatrix} 1 & 1 & 3 & 1 \\ 2 & 4 & 1 & 0 \\ 1 & 3 & 5 & 3 \\ 3 & 0 & 4 & 0 \end{bmatrix} \tag{5.84}$$

Note that for 3×3 matrices we only have to examine the sign of the determinant of one minor: namely,

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{bmatrix}. \tag{5.85}$$

For 4×4 matrices, we additionally must check the signs of the determinants of the following minors:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{bmatrix}, \tag{5.86}$$

and

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}. \tag{5.87}$$

For matrix A given in (5.83), the minor having the form given in (5.85) has a negative determinant and thus satisfies the assumption, while for matrix A' in (5.83), the minor of the form given in (5.85) has positive determinant and so does not satisfy the assumption. For matrix A'' in (5.84), we see that the signs of the determinants of the minors in (5.85) - (5.87) are all negative, so A'' satisfies the assumptions of Theorem 5.3.3. On the other hand, the minor of A''' having the form of (5.85) has a positive determinant. Thus, A''' does not satisfy the assumptions of the theorem.

5.3.3 Implementation

The proofs of both Theorem 5.3.1 and Theorem 5.3.3 require the uniqueness of the solution $\hat{x}$. However, because of the increased dimensionality, problems (5.53) and (5.60) rarely have a unique solution. This is seen in the following simple example. Consider the linear MOP:

$$\begin{aligned} \text{minimize} \quad & f(x) = [-x_1, -x_2, -x_3] \\ \text{subject to} \quad & x_1 + x_2 + x_3 \leq 1 \\ & x \in \mathbb{R}^3_{\geqq} \end{aligned} \tag{5.88}$$

Let $\delta = 0.5$ and $x^* = (1, 0, 0) \in X_E$. Then, CCS_1^3(0.5, (1,0,0), $\mathbb{R}^3_{\geqq}$) is (note that we dropped the "theoretical" constraints):

$$
\begin{aligned}
\text{minimize} \quad & -x_1 \\
\text{subject to} \quad & -x_2 \leq 0 \\
& -x_3 \leq 0 \\
& x_1 + x_2 + x_3 \leq 1 \\
& ||[-x_1, -x_2, -x_3] - [-1, 0, 0]|| \geq 0.5 \\
& x \in \mathbb{R}^3_{\geqq}
\end{aligned}
\tag{5.89}
$$

Assume that we are using the l_1-norm. Then the spacing constraint can be rewritten as:

$$
| - x_1 + 1| + | - x_2| + | - x_3| \geq 0.5 \tag{5.90}
$$

In this case, since we know that the Pareto set of problem (5.88) is the portion of the plane $x_1 + x_2 + x_3 = 1$ which lies in the first orthant, we can eliminate the absolute values from (5.90):

$$
x_1 - x_2 - x_3 \leq 0.5 \tag{5.91}
$$

The KKT conditions for problem (5.89), with the spacing constraint as in (5.91), are:

$$\begin{aligned}
&\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} + u_1 \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} + u_2 \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} + u_3 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + u_4 \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\
&-x_2 \leq 0 \\
&-x_3 \leq 0 \\
&x_1 + x_2 + x_3 \leq 1 \\
&x_1 - x_2 - x_3 \leq 0.5 \\
&u_1(-x_2) = 0 \\
&u_2(-x_3) = 0 \\
&u_3(x_1 + x_2 + x_3 - 1) = 0 \\
&u_4(x_1 - x_2 - x_3 - 0.5) = 0 \\
&u \in \mathbb{R}^4_{\geqq} \\
&x \in \mathbb{R}^3_{\geqq}
\end{aligned} \tag{5.92}$$

Solving the gradient conditions in (5.92), we find that $u_1 = u_2$ and $u_3 = 1 - u_4$. Since $x^* = (1, 0, 0)$, we also know that at least one of x_2 and x_3 must increase from 0. Without loss of generality, assume that $x_2 > 0$. Then, $u_1 = 0$, by complementary slackness, which implies $u_2 = 0$. Substituting these values into the gradient conditions, we find that $u_3 = u_4$. This, combined with $u_3 = 1 - u_4$, implies that $u_3 = u_4 = 0.5$. Thus, we must have that $x_1 + x_2 + x_3 = 1$ and $x_1 - x_2 - x_3 = 0.5$. Adding these two equations together gives that $x_1 = 0.75$ and $x_2 + x_3 = 0.25$. Hence, problem (5.89) has alternate optimal solutions: $\hat{x} = (0.75, x_2, 0.25 - x_2)$ where $x_2 \in [0, 0.25]$.

Additionally, again because of the increase in dimension, neither problem (5.53) nor (5.60) suggest an intuitive structure for an algorithm. In more than two dimensions, there is no inherent organizational method for producing a representation, and this combined with the lack of uniqueness discussed previously creates a somewhat chaotic generation of points, as was seen in computational experiments. Moreover, since the feasibility check from the biobjective approach does not generalize to higher dimensions, we do not have an intuitive stopping criterion to integrate into an algorithm which is based on these problems.

Due to these considerations, we recommend using a modified version of problems (5.53) and (5.60) in conjunction with certain aspects of the Bilevel Controlled Spacing (BCS) approach (which is discussed in Chapter 4) to implement the CCS method in more than two dimensions. The new problem, instead of considering the nondominated set as a whole, reduces the nondominated set to its two-dimensional cross-sections (similar to the slicing technique of the BCS approach) where we may then apply the biobjective CCS method. The BCS approach is used only to determine the reference points for input into the algorithm. This implementation injects the structure we need to formulate an algorithm.

Let $C \supseteq \mathbb{R}^p_{\geqq}$ be a simplicial cone defined by $C = \{y \in \mathbb{R}^p : Ay \geqq 0\}$, and let $x^* \in E_W(X, f, C)$. Since we are now focusing on a two-dimensional cross-section of the problem, we may again apply the feasibility check $||\tilde{f}(x^*) - \tilde{f}(x^2)|| \geq \delta$ where $\tilde{f}(x) = [f_1(x), f_2(x)]$ and

$$x^2 \in \arg \operatorname{lex}\min\{[f_2(x), f_1(x)] : x \in E_W(X, f, C), f_3(x) = f_3^*, \ldots, f_p(x) = f_p^*\}. \tag{5.93}$$

Given this, the Constraint Controlled Spacing formulation using the slicing method ($\text{CCSS}_1^p(\delta$, x^*, $A)$) follows:

$$\begin{array}{ll} \text{minimize} & g_1(x) \\ \text{subject to} & f_1(x) \leq f_1(x^2) \\ & g_2(x) \leq g_2(x^*) \\ & f_3(x) = f_3(x^*) \\ & \vdots \\ & f_p(x) = f_p(x^*) \\ & ||\tilde{f}(x) - \tilde{f}(x^*)|| \geq \delta \\ & x \in X \end{array} \tag{5.94}$$

We prove that problem (5.94) produces a weakly efficient point next.

Theorem 5.3.4. *Let $C \supseteq \mathbb{R}^p_{\geqq}$ be a simplicial cone defined by $C = \{y \in \mathbb{R}^p : Ay \geqq 0\}$. If $N_W(Y, C)$ is connected and $\hat{x}$ is the unique optimal solution of CCSS$_1^p$(δ, x^*, A), then $f(\hat{x}) \in N_W(Y, C)$.*

Proof. Assume $N_W(Y, C)$ is connected and let $\hat{x}$ be the unique optimal solution of CCSS$_1^p$(δ, x^*, A). Let $\hat{y} = f(\hat{x})$ and $\hat{z} = Af(\hat{x})$. Suppose that $\hat{y} \notin N_W(Y, C)$. Recall that we require $||\tilde{f}(x^*) - \tilde{f}(x^2)|| \geq \delta$, so x^2 (as defined in (5.93)) is feasible for CCSS$_1^p$(δ, x^*, A) and the problem always has a feasible solution. Now, notice that if we let $\tilde{X} = \{x \in X : f_3(x) = f_3(x^*), \ldots, f_p(x) = f_p(x^*)\}$, then problem (5.94) becomes

$$\begin{aligned} \text{minimize} \quad & g_1(x) \\ \text{subject to} \quad & f_1(x) \leq f_1(x^2) \\ & g_2(x) \leq g_2(x^*) \\ & ||\tilde{f}(x) - \tilde{f}(x^*)|| \geq \delta \\ & x \in \tilde{X} \end{aligned} \tag{5.95}$$

The problem in (5.95) is now precisely the bicriteria CCS problem (5.6), so we may apply the results of Theorem 5.1.3 and conclude that $f(\hat{x}) \in N_W(Y, C)$. This completes the proof. □

The pseudocode for the Constraint Controlled Spacing Slicing Algorithm (CCSSA) is given in Figure 5.9. CCSSA takes as inputs the indices (j, k) of the objective functions that are not fixed, the scalars n_i which are the number of slices desired with respect to each fixed objective function, and the spacing value δ. Next, for each fixed objective function, we determine the minimum and maximum values, f_i^l and f_i^h, over the nondominated set. That is, to solve for f_i^l, for example, we solve the following bilevel problem:

$$\begin{aligned} \text{minimize} \quad & f_i(x) \\ \text{subject to} \quad & x \in E_W(X, f, C) \end{aligned} \tag{5.96}$$

We reformulate the problem in (5.96) as a single-level problem by rewriting the lower-level problem in terms of its KKT conditions, and then solve the resultant problem using the branch-and-bound method discussed in Chapter 4. We calculate the distance between each pair of extreme values and divide by $n_i - 1$ to find the spacing between each slice. A series of nested for-loops is used to investigate all possible cross-sections. Note that these loops are only over the fixed objective functions. For each possible combination of the fixed values, we specify the set $\tilde{X}$ and determine the minimum values of $f_j(x)$ and $f_k(x)$ over this cross-section of the nondominated set, again using the bilevel methodology previously discussed. These points are used as the reference points in the biobjective CCSA. Since the length of each cross-section will (most likely) be different, the number of points desired, N, which is used as an input of CCSA (see Figure 5.7) cannot be specified as an input. Rather, we recommend using a slightly modified version of CCSA, CCSA$_\delta$, in which the main while-loop depends on there being a feasible solution rather than the input N. The pseudocode for CCSA$_\delta$ is shown in Figure 5.10. The same issue as was discussed with respect to CCSA applies here. That is, if δ does not "fit" the current cross section, then the final two points will be less than δ distance apart. Each time we run CCSA$_\delta$, we obtain a list L of efficient points which we add to the list of lists, $\mathcal{L}$. CCSSA returns $\mathcal{L}$ at the conclusion of the algorithm.

The advantages of the Constraint Controlled Spacing Slicing (CCSS) method are that it aids in the visualization of the nondominated set even for problems with more than three dimensions. An advantage this method has over the Bilevel Controlled Spacing Slicing (BCSS) method is that the CCSS method is not limited to convex problems. However, while the BCSS method can accommodate preferences modeled by both acute and obtuse cones, a disadvantage of the CCSS method is that we can only integrate preferences modeled by obtuse cones. Additionally, since we utilize techniques from the BCSS approach to determine bounds on the nondominated set, there is a possibility that the bounds will be incorrect on nonconvex problems. However, this has not been an issue in any of our computational tests thus far.

algorithm CCSSA

input: $j, k \in \{1, \ldots, p\}$, n_i for $i = 1, \ldots, p, i \neq j, i \neq k$, δ

begin

find $f_i^l = \min\{f_i(x) : x \in E_W(X, f, C)\}$ for $i = 1, \ldots, p, i \neq j, i \neq k$;
find $f_i^h = \max\{f_i(x) : x \in E_W(X, f, C)\}$ for $i = 1, \ldots, p, i \neq j, i \neq k$;
compute $d_i = (f_i^h - f_i^l)/(n_i - 1)$ for $i = 1, \ldots, p, i \neq j, i \neq k$;
set $\mathcal{L} = \{\}$;
for $count_1 = 1, \ldots, n_1$
  $f_1^* = f_1^l + (count_1 - 1)d_1$;
  $\vdots$
  for $count_p = 1, \ldots, n_p$
    $f_p^* = f_p^l + (count_p - 1)d_p$;
    set $\tilde{X} = \{x \in \mathbb{R}^n : f_i(x) = f_i^* \text{ for } i = 1, \ldots, p, i \neq j, i \neq k\}$;
    find $x^1 \in \arg \operatorname{lex}\min\{[f_j(x), f_k(x)] : x \in \tilde{X}, x \in E_W(X, f, C)\}$;
    find $x^2 \in \arg \operatorname{lex}\min\{[f_k(x), f_j(x)] : x \in \tilde{X}, x \in E_W(X, f, C)\}$;
    run algorithm CCSA$_\delta$ with inputs δ, x^1, x^2 to obtain L;
    append L to $\mathcal{L}$;
  end for;
end for;

end;

output: $\mathcal{L}$

Figure 5.9: Pseudocode for CCSSA

algorithm CCSA_δ

obtain x^1 and x^2 from DM or using *Initialization*

input: δ

begin

    let $x^* = x^1$, let $L = \{x^1\}$, and set $i = 1$;
    while $||f(x^*) - f(x^2)|| \geq \delta$ **do**
    begin
        solve $\mathrm{CCS}_1(\delta, x^*, A)$ to obtain $x^{(i)}$;
        solve $\mathrm{P}_2(g_1(x^{(i)}))$ to obtain x';
        if $g_2(x^{(i)}) = g_2(x')$ **then** set $x^* = x^{(i)}$;
        else set $x^* = x'$;
        end if;
        set $L(i+1) = x^*$;
        $i++$;
    end while;
    set $L(i) = x^2$;

end;

output: L

Figure 5.10: Pseudocode for CCSA_δ

Chapter 6

Numerical Experiments

In this chapter, we present the results of our two methods, the Bilevel Controlled Spacing (BCS) method and the Constraint Controlled Spacing (CCS) method, when applied to a variety of test problems. All results were obtained using Matlab version 7.7.0. We first give the BCS results, bicriteria problems in Section 6.1.1 and tricriteria problems in Section 6.1.2. Then we move to the results of the CCS method, with bicriteria results given in Section 6.2.1 and tricriteria results given in Section 6.2.2. In all cases, any lines displayed in the figures are simply to aid in visualization and are not part of the representation. Further, all numerical experiments were performed using the l_1-norm unless otherwise noted.

6.1 Bilevel Controlled Spacing

6.1.1 Biobjective Results

The BCS method was applied to three convex biobjective test problems. Example 1 (6.1) is a simple linear MOP. We set $N = 17$ and solved the problem first with respect to the Pareto cone and then with respect to the acute cone defined by the matrix

$$A = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}.$$

The results are shown in Figures 6.1 and 6.2, respectively. In both figures, the displayed line represents the true nondominated set.

$$\begin{aligned}
\text{minimize} \quad & f(x) = [-x_1 - 2x_2,\ -3x_1 - x_2] \\
\text{subject to} \quad & x_1 + x_2 \leq 6 \\
& 2x_1 + x_2 \leq 9 \\
& x_1 \leq 4 \\
& x_2 \leq 5 \\
& x_1,\ x_2 \geq 0 \\
& x \in \mathbb{R}^2
\end{aligned} \tag{6.1}$$

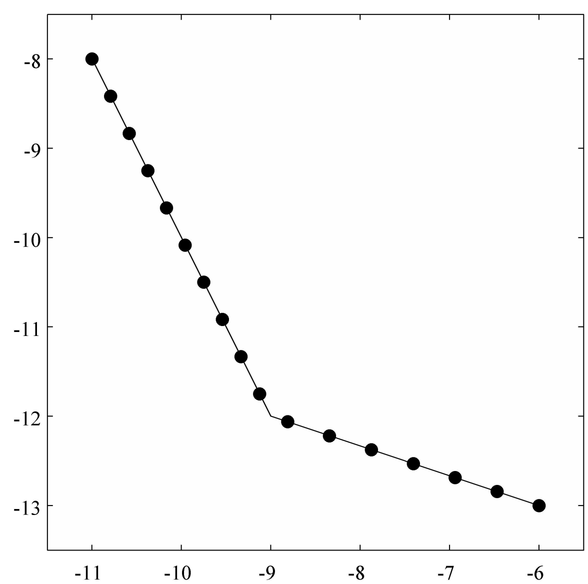

Figure 6.1: Example 1 with the Pareto cone

Example 2 (6.2) is a convex nonlinear MOP from [70]. We first set $N = 17$ and solved the problem with respect to the Pareto cone. The results are shown in Figure 6.3. Then, we set $N = 9$ and solved the problem with respect to the obtuse cone defined by the

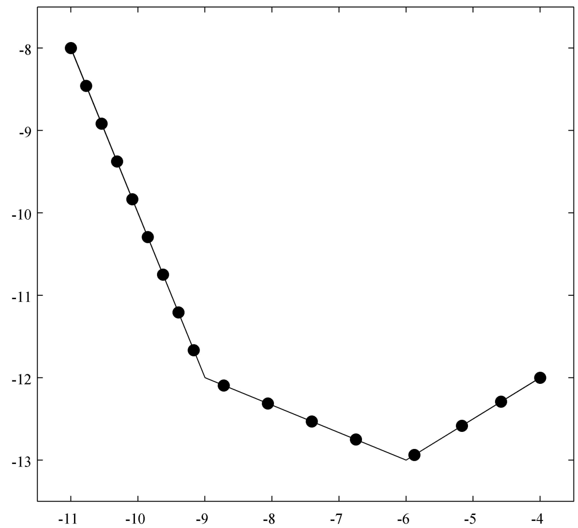

Figure 6.2: Example 1 with an acute cone

matrix

$$A = \begin{bmatrix} 10 & 1 \\ 0 & 1 \end{bmatrix}.$$

The results are shown in Figure 6.4.

$$\begin{aligned} \text{maximize} \quad & f(x) = [x_1 + x_2,\ 10x_1 - x_1^2 + 4x_2 - x_2^2] \\ \text{subject to} \quad & 3x_1 + x_2 \leq 12 \\ & 2x_1 + x_2 \leq 9 \\ & x_1 + 2x_2 \leq 12 \\ & x \in \mathbb{R}^2 \end{aligned} \tag{6.2}$$

Example 3 (6.3) is a simple convex problem from [30] which we use to demonstrate the controlled-tradeoff application of the BCS method. Figure 6.5 shows the generated representation. The solutions points from top to bottom have tradeoffs of $\frac{1}{4}$, $\frac{1}{3}$, $\frac{1}{2}$, 1, 2, 3,

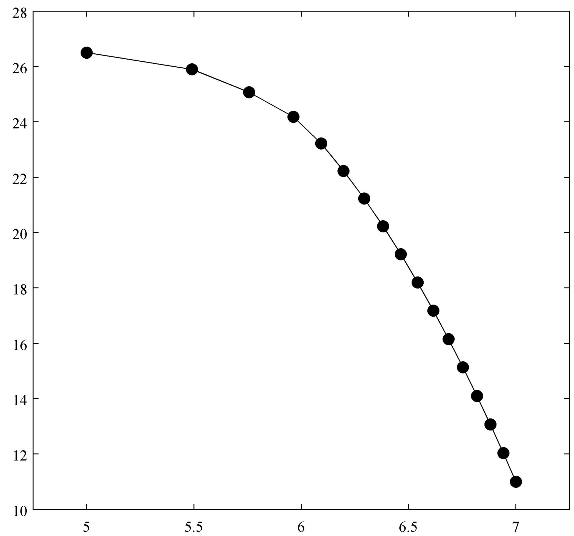

Figure 6.3: Example 2 with the Pareto cone

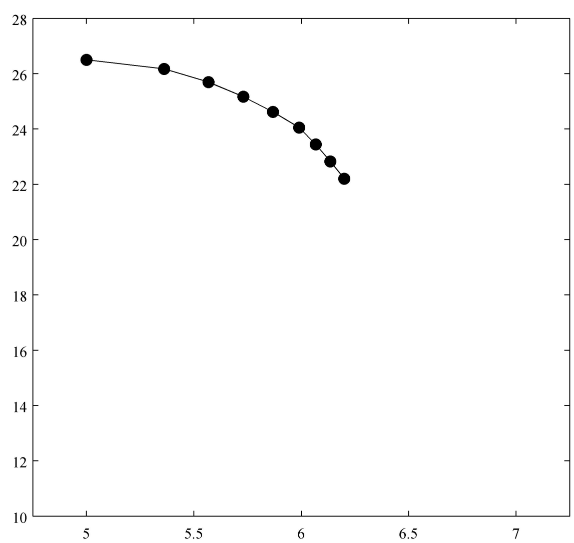

Figure 6.4: Example 2 with an obtuse cone

and 4, respectively. Compare this figure to Figures 6.21 and 6.22.

$$\begin{aligned}
\text{minimize} \quad & f(x) = [\sqrt{1+x_1^2},\ x_1^2 - 4x_1 + x_2 + 5] \\
\text{subject to} \quad & x_1^2 - 4x_1 + x_2 + 5 \leq 3.5 \\
& x_1,\ x_2 \geq 0 \\
& x \in \mathbb{R}^2
\end{aligned} \tag{6.3}$$

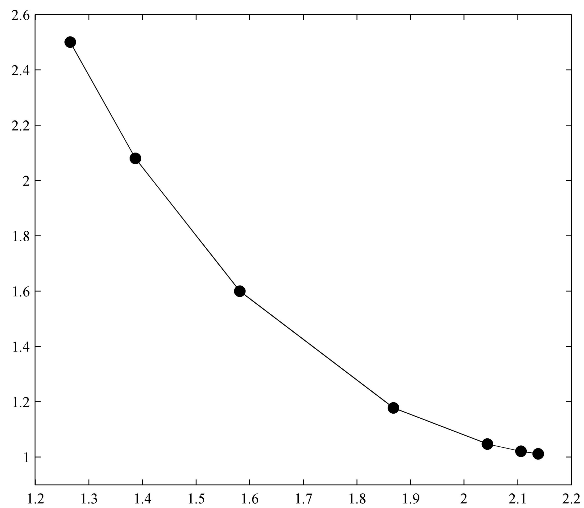

Figure 6.5: Example 3 with controlled-tradeoff

6.1.2 Multiobjective Results

The BCS method was applied to three convex, tricriteria problems. Example 4 (6.4) is a simple linear problem.

$$\begin{aligned}
\text{minimize} \quad & f(x) = [-x_1,\ -x_2,\ -x_3] \\
\text{subject to} \quad & x_1 + x_2 + x_3 \leq 3 \\
& x_1,\ x_2,\ x_3 \geq 0 \\
& x \in \mathbb{R}^3
\end{aligned} \tag{6.4}$$

We first ran the BCS method using the center technique. These results are shown in Figures 6.6 and 6.7. We next applied the BCS method using the slicing technique. These results are shown in Figures 6.8 and 6.9. For this example, the Pareto preference was assumed and the l_2-norm was used.

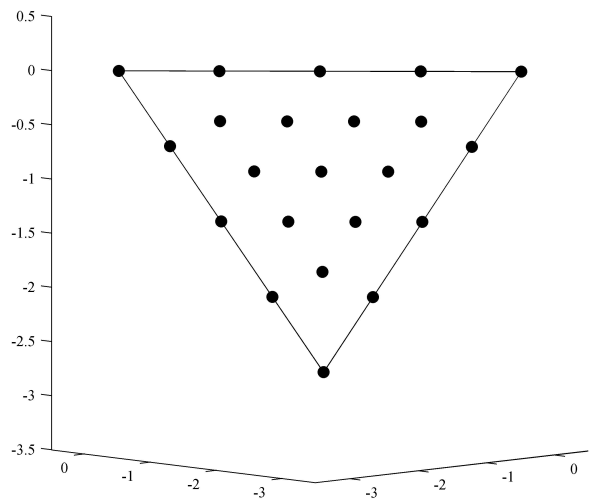

Figure 6.6: Example 4 using the center technique: front view

Example 5 (6.5) is a linear problem with two nondominated faces taken from [43].

$$
\begin{aligned}
\text{minimize} \quad & f(x) = [-4x_1 - x_2 - 2x_3,\ -x_1 - 3x_2 + x_3,\ x_1 - x_2 - 4x_3] \\
\text{subject to} \quad & x_1 + x_2 + x_3 \leq 3 \\
& 2x_1 + 2x_2 + x_3 \leq 4 \\
& x_1 - x_2 \leq 0 \\
& x_1,\ x_2,\ x_3 \geq 0 \\
& x \in \mathbb{R}^3
\end{aligned}
\tag{6.5}
$$

For this example, we applied the slicing method. We chose to fix f_3 and set the number of slices at $n_3 = 10$. The results are shown in Figures 6.10 and 6.11.

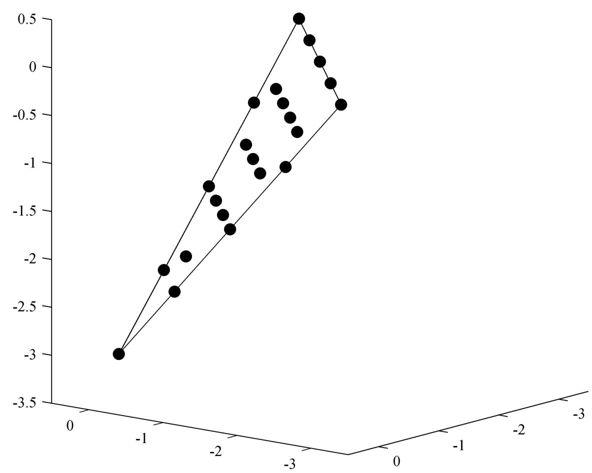

Figure 6.7: Example 4 using the center technique: side view

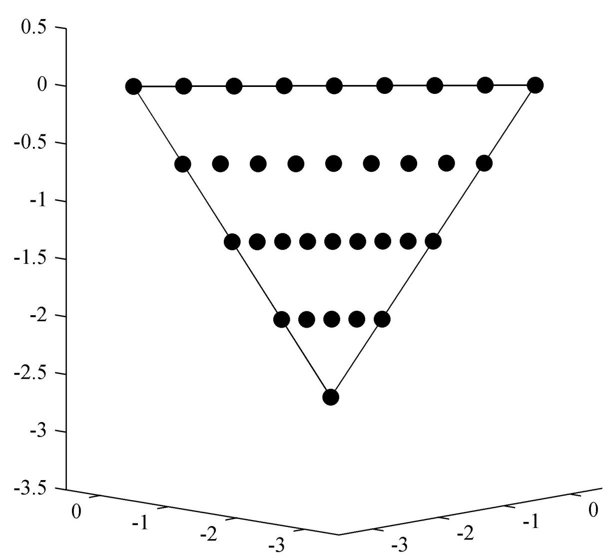

Figure 6.8: Example 4 using the slicing technique: front view

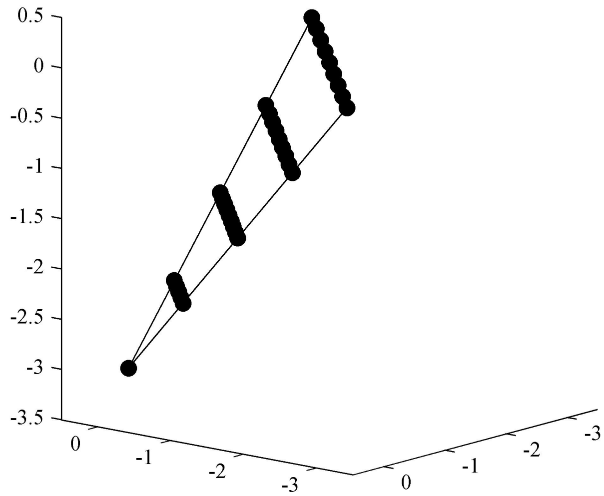

Figure 6.9: Example 4 using the slicing technique: side view

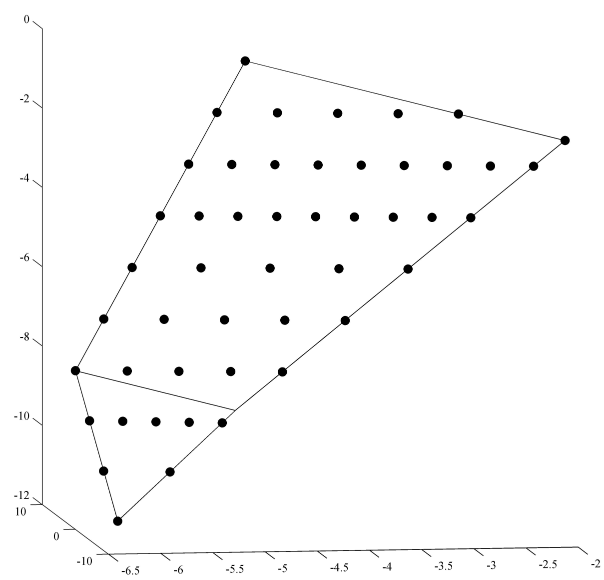

Figure 6.10: Example 5 using the slicing technique: front view

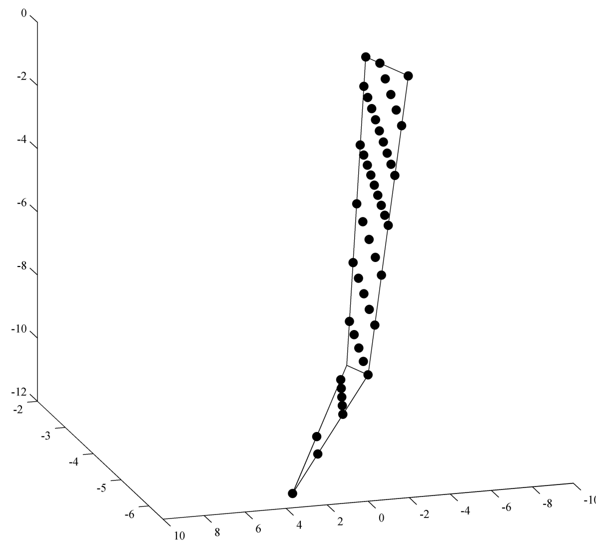

Figure 6.11: Example 5 using the slicing technique: side view

Example 6 (6.6) is a simple convex problem.

$$
\begin{aligned}
\text{minimize} \quad & f(x) = [-x_1,\ -x_2,\ -x_3] \\
\text{subject to} \quad & x_1^2 + x_2^2 + x_3^2 \leq 1 \\
& x_1,\ x_2,\ x_3 \geq 0 \\
& x \in \mathbb{R}^3
\end{aligned}
\tag{6.6}
$$

We ran this example using both the center and the slicing techniques for each of the three types of cones (Pareto, acute, and obtuse). For the center method, we stopped when the areas of all the search regions were less than or equal to 0.06 units. For the slicing method, the number of slices varied depending on the cone, but in all cases, we set $\delta = 0.30$. Additionally, the l_2-norm was used for all tests. The results for the Pareto cone are shown in Figures 6.12, 6.13, and 6.14. Matrices A and A' in (6.7) were used to define the acute and obtuse cones, respectively, that we investigated.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \; A' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{6.7}$$

The results for the acute cone represented by matrix A are shown in Figures 6.15, 6.16, and 6.17. The results for the obtuse cone represented by matrix A' are shown in Figures 6.18, 6.19, and 6.20.

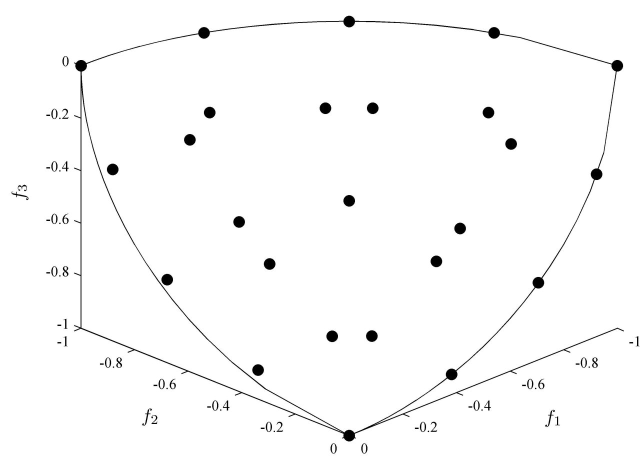

Figure 6.12: Example 6 with the Pareto cone using the center technique

6.2 Constraint Controlled Spacing

6.2.1 Biobjective Results

The CCS method was applied to four biobjective test problems. Example 7 (6.3) is a simple convex problem from [30]. We first set our cardinality preferences to $N = 9$, resulting in $\delta = 0.456$. The results for this run are shown in Figure 6.21. We then let

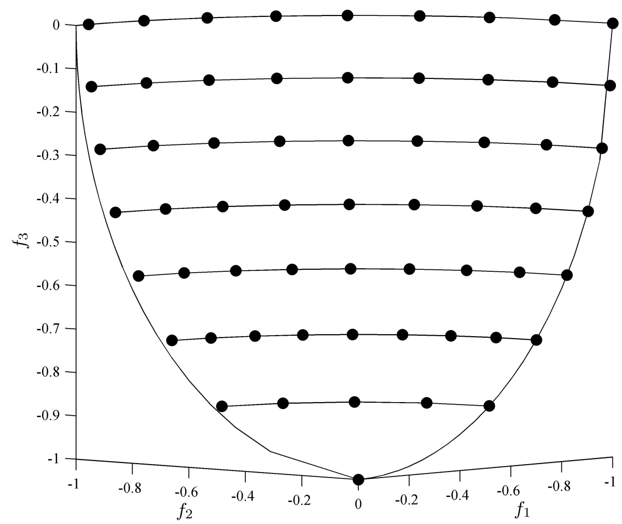

Figure 6.13: Example 6 with the Pareto cone using the slicing technique: front view

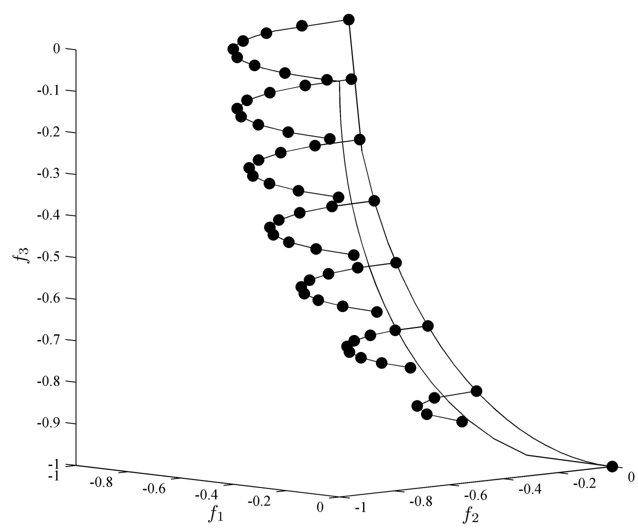

Figure 6.14: Example 6 with the Pareto cone using the slicing technique: side view

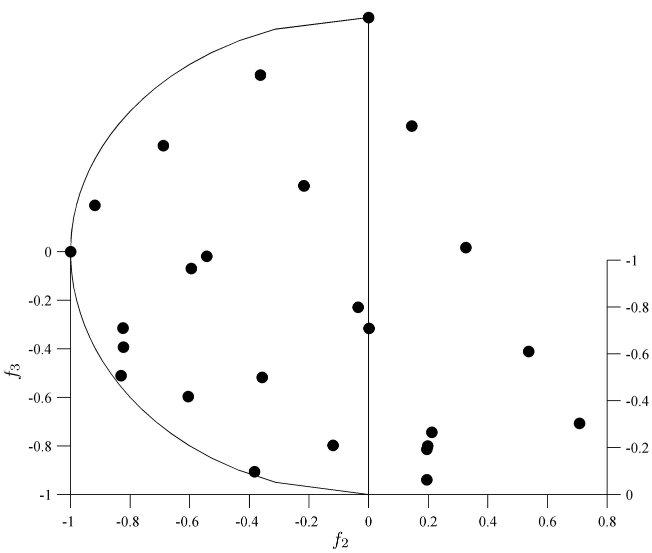

Figure 6.15: Example 6 with an acute cone using the center technique

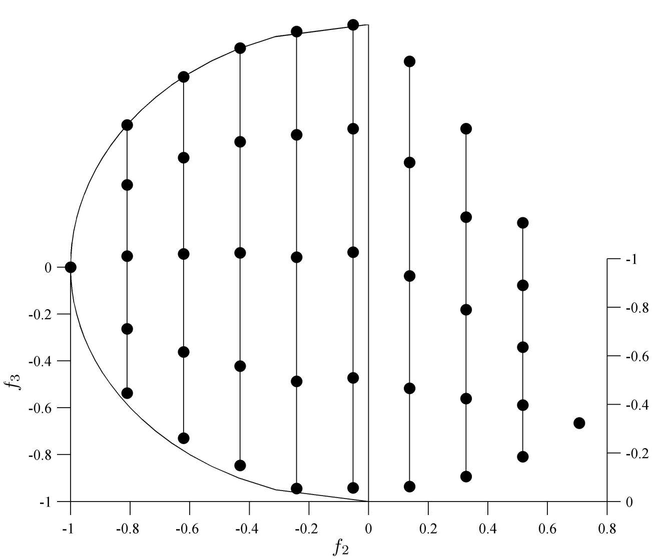

Figure 6.16: Example 6 with an acute cone using the slicing technique: front view

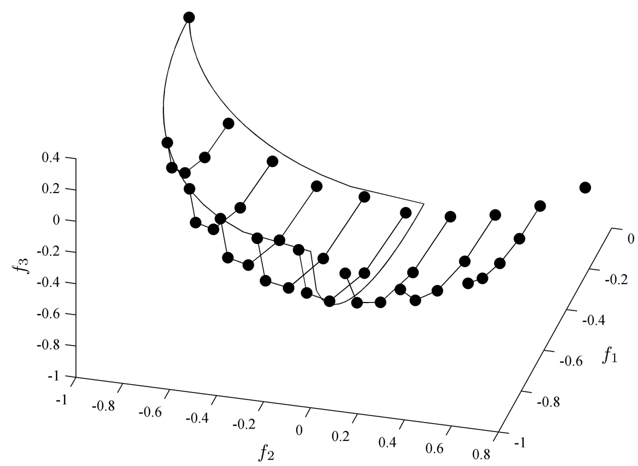

Figure 6.17: Example 6 with an acute cone using the slicing technique: side view

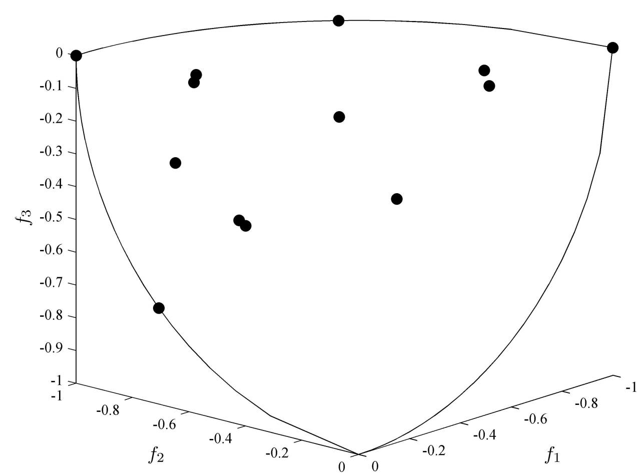

Figure 6.18: Example 6 with an obtuse cone using the center technique

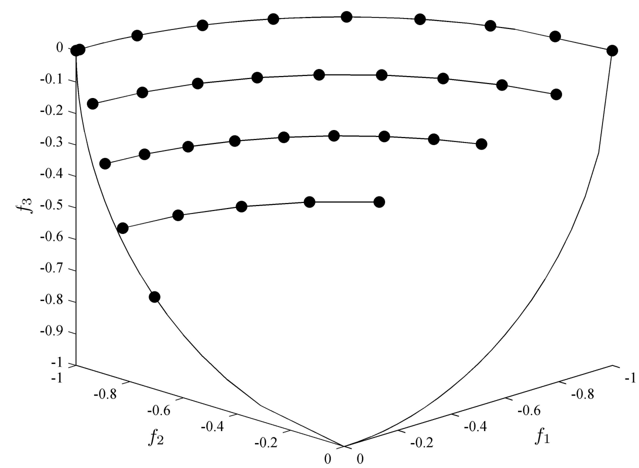

Figure 6.19: Example 6 with an obtuse cone using the slicing technique: front view

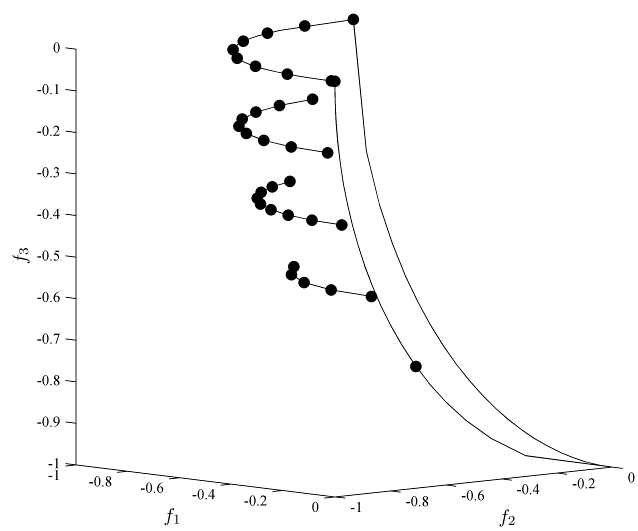

Figure 6.20: Example 6 with an obtuse cone using the slicing technique: side view

$N = 17$ yielding $\delta = 0.228$. These results are shown in Figure 6.22. Both representations were generated with respect to the Pareto cone.

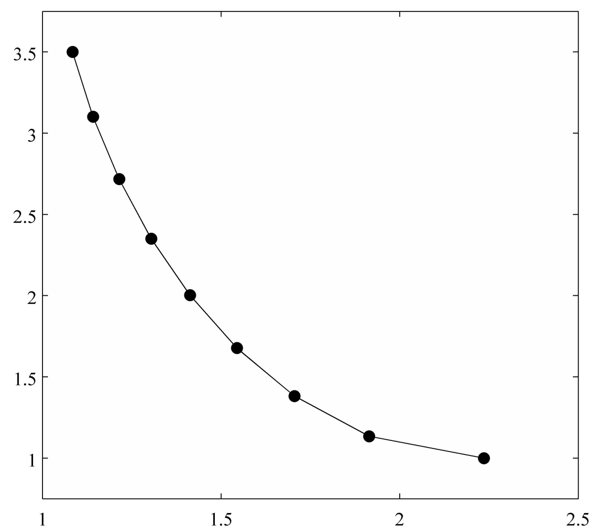

Figure 6.21: Example 7 with the Pareto cone and $N = 9$

Example 8 [70], shown in (6.8), is a nonconvex problem with a disconnected Pareto set and a connected weak Pareto set.

$$
\begin{aligned}
&\text{minimize} && f(x) = [f_1(x),\ f_2(x)] \\
&\text{where} && f_1(x) = 10 + 10 \sum_{j=3}^{4} \sum_{i=1}^{2} (x_i - 2)^j \\
& && f_2(x) = 10 + \sum_{i=1}^{2} (x_i - 3)^2 \\
&\text{subject to} && -x_1 - x_2 + 0.1 \leq 0 \\
& && 0 \leq x_1 \leq 10 \\
& && 0 \leq x_2 \leq 10 \\
& && x \in \mathbb{R}^2
\end{aligned}
\tag{6.8}
$$

We first solved the problem with respect to the Pareto cone with $N = 100$. The results are shown in Figure 6.23. In Figure 6.24, we kept $N = 100$ and solved the problem with

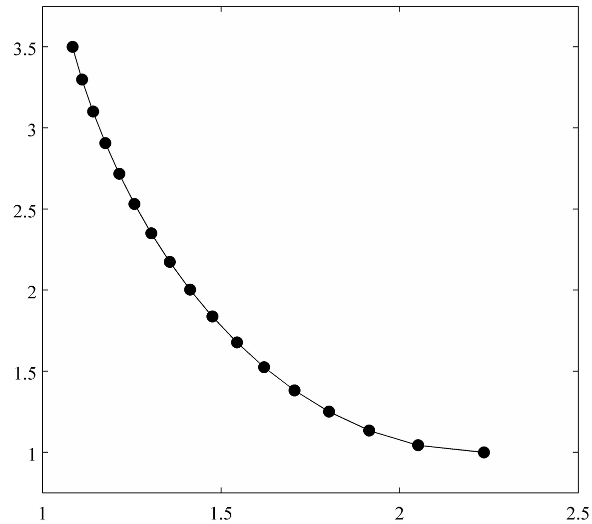

Figure 6.22: Example 7 with the Pareto cone and $N = 17$

respect to the obtuse cone defined by the matrix

$$A = \begin{bmatrix} 3 & 1 \\ 0 & 1 \end{bmatrix}.$$

Note that in both figures we "zoomed in" on the interesting portion of the nondominated set.

Example 9 (6.9), which was originally proposed in [77], is a nonconvex problem with a disconnected weak Pareto set.

$$\begin{aligned} \text{minimize} \quad & f(x) = [x_1,\ x_2] \\ \text{subject to} \quad & x_1^2 + x_2^2 - 1 - 0.1\cos(16\arctan(\tfrac{x_1}{x_2})) \geq 0 \\ & (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5 \\ & x_1,\ x_2 \in [0, \pi] \\ & x \in \mathbb{R}^2 \end{aligned} \tag{6.9}$$

The results when the problem is solved with respect to the Pareto cone and $N = 20$ are

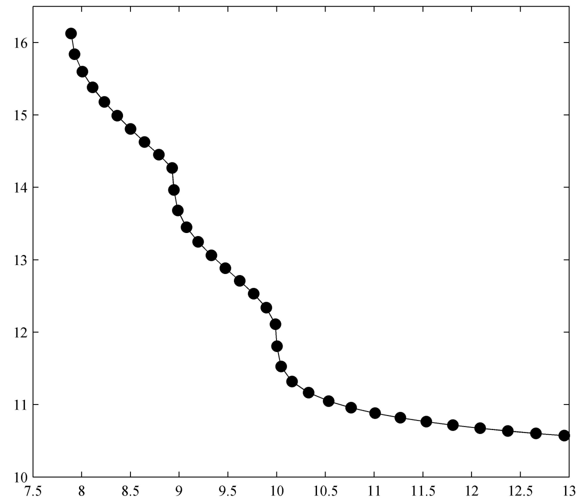

Figure 6.23: Example 8 with the Pareto cone

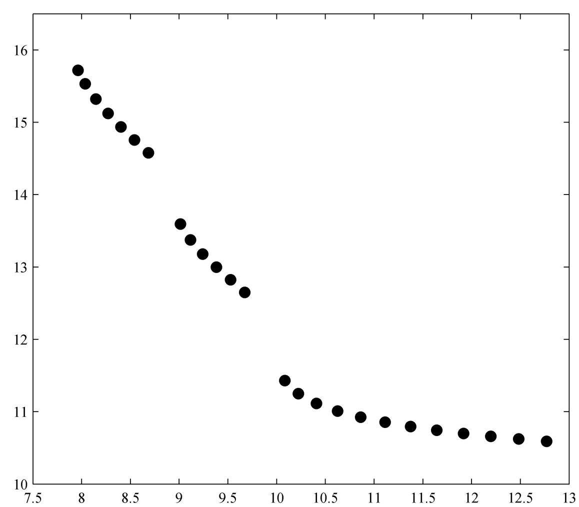

Figure 6.24: Example 8 with an obtuse cone

shown in Figure 6.25. The grey area in the figure denotes Y, the feasible region in the objective space.

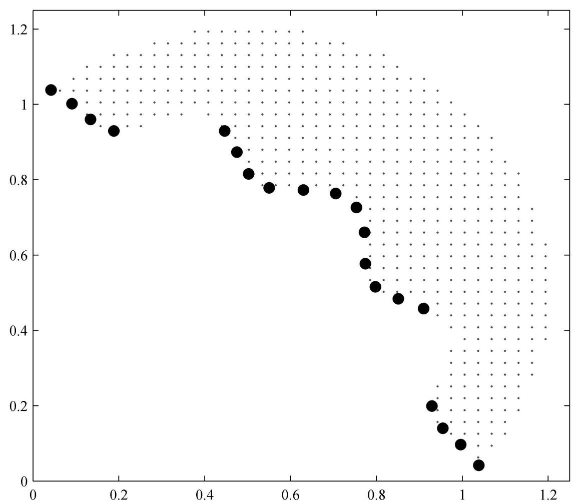

Figure 6.25: Example 9 with the Pareto cone

Example 10 (6.10), a slight modification of a problem presented in [41], is another nonconvex problem with a disconnected weak Pareto set.

$$\begin{aligned} \text{minimize} \quad & f(x) = [-x_1,\ x_1 + x_2^2 - \cos(50x_1)] \\ \text{subject to} \quad & x_1^2 - x_2 \leq 0 \\ & x_1 + 2x_2 - 3 \leq 0 \\ & x_1 \geq 0.5 \\ & x \in \mathbb{R}^2 \end{aligned} \tag{6.10}$$

We let $N = 16$ and solved the problem with respect to the Pareto cone. The results are shown in Figure 6.26. We then let $N = 20$ and solved the problem with respect to the obtuse cone defined by the matrix

$$A = \begin{bmatrix} 31 & 1 \\ 0 & 1 \end{bmatrix}.$$

The line shown in both figures represents the boundary of Y. Notice that in Figure 6.27, the chosen δ does not exactly "fit" the problem so the final two points have a distance of less than δ from each other.

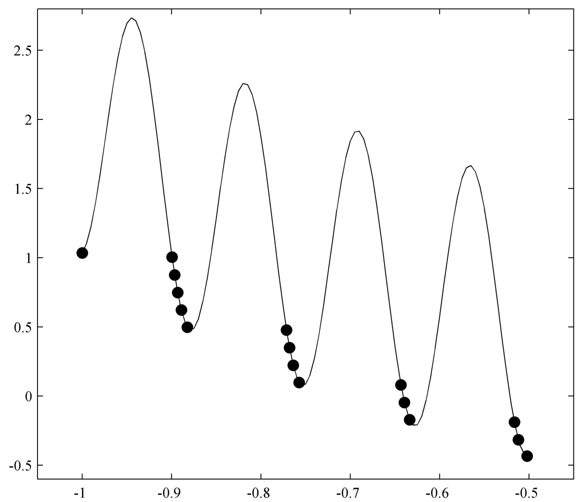

Figure 6.26: Example 10 with the Pareto cone

6.2.2 Multiobjective Results

The CCS method was applied to three tricriteria test problems. For comparison of the BCS method and the CCS method, Example 11 (6.6) is the same as Example 6 in Section 6.1.2. We first ran the problem with respect to the Pareto cone and then with respect to obtuse cone defined by the matrix A' in (6.7). In both cases, we sliced with respect to f_3 and set $\delta = 0.30$. The results for the Pareto case are shown in Figures 6.28 and 6.29, and the results for the obtuse case are shown in Figures 6.30 and 6.31.

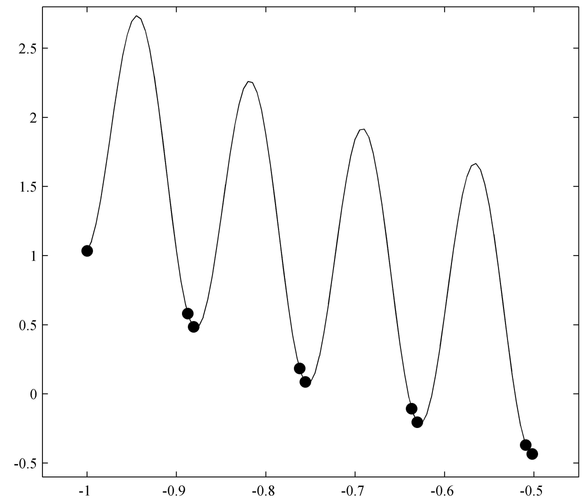

Figure 6.27: Example 10 with an obtuse cone

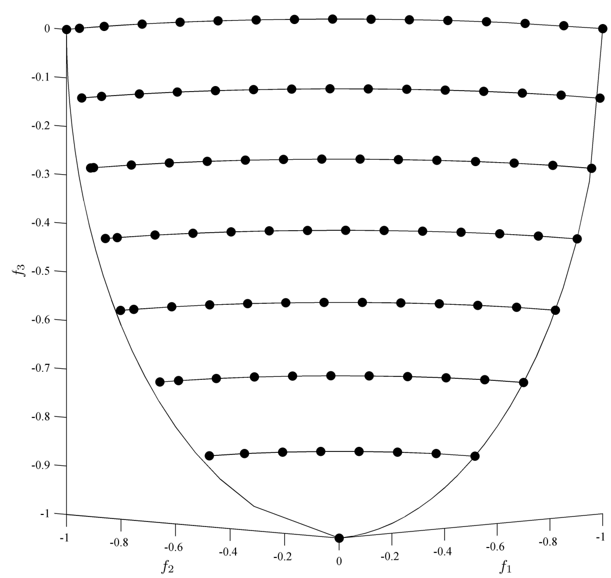

Figure 6.28: Example 11 with the Pareto cone: front view

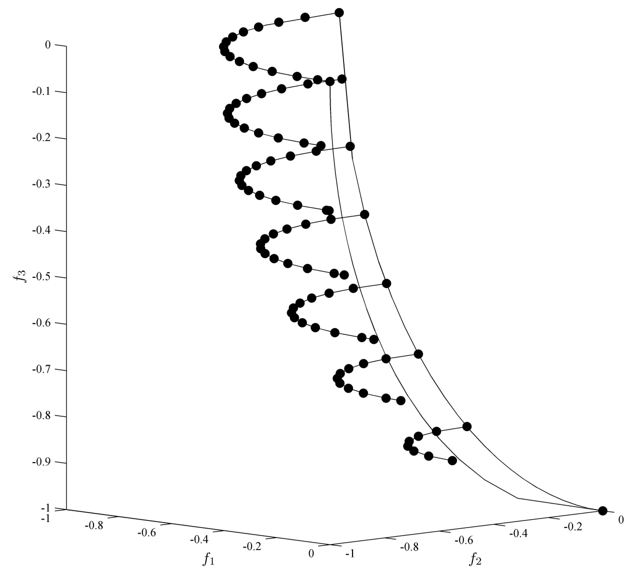

Figure 6.29: Example 11 with the Pareto cone: side view

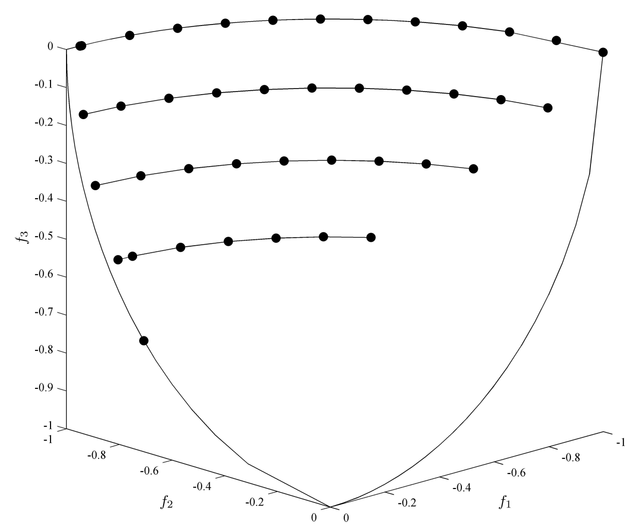

Figure 6.30: Example 11 with an obtuse cone: front view

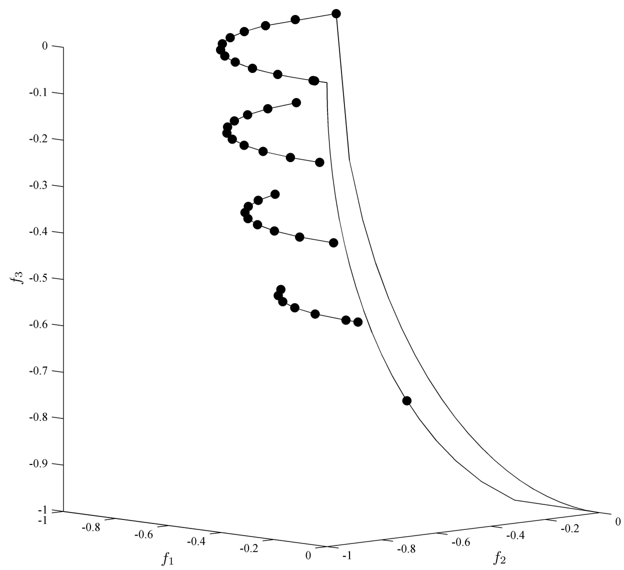

Figure 6.31: Example 11 with an obtuse cone: side view

Example 12 (6.11) is another simple convex problem.

$$
\begin{aligned}
\text{minimize} \quad & f(x) = [x_1,\ x_2,\ x_3] \\
\text{subject to} \quad & 4x_1^2 + x_2^2 - x_3 \leq 10 \\
& x \in \mathbb{R}^3
\end{aligned}
\tag{6.11}
$$

We ran Example 12 with respect to the Pareto cone, slicing with respect to f_3, and setting $\delta = 0.30$. Figures 6.32 and 6.33 show the results.

Example 13 (6.12) is taken from [29] and is a slight modification of a problem found

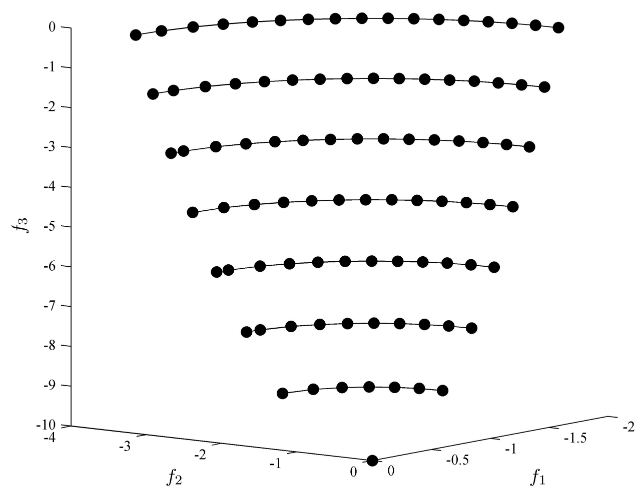

Figure 6.32: Example 12 with the Pareto cone: front view

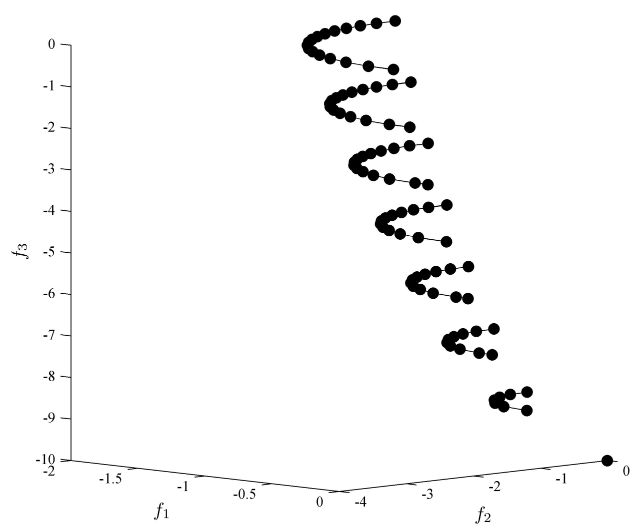

Figure 6.33: Example 12 with the Pareto cone: side view

in [44].

$$\begin{aligned} \text{minimize} \quad & f(x) = [-x_1,\ -x_2,\ -x_3^2] \\ \text{subject to} \quad & -\cos(x_1) - e^{-x_2} + x_3 \leq 0 \\ & 0 \leq x_1 \leq \pi \\ & x_2 \geq 0 \\ & x_3 \geq 1.2 \\ & x \in \mathbb{R}^3 \end{aligned} \tag{6.12}$$

This problem has a nonconvex Pareto set. We ran Example 13 with respect to the Pareto cone and an obtuse cone defined by the matrix

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

In both cases, we sliced with respect to f_2 and set $\delta = 0.20$. We present four views of the nondominated set for both the Pareto cone and the obtuse cone defined by A. For ease of comparison, in Figures 6.34 - 6.41, we show each view first for the Pareto set and then for the nondominated set.

We would like to emphasize that few methods in the literature have been tested on problems in higher than two dimensions. This is because of the fact that higher dimensional MOPs are innately more complex and difficult to solve, regardless of the method or solver used.

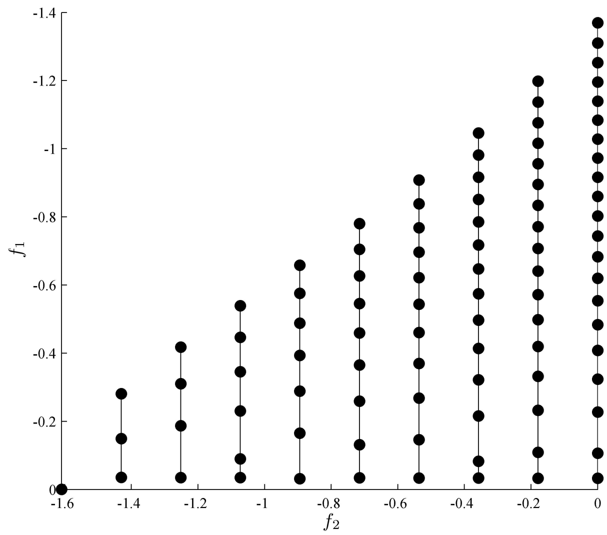

Figure 6.34: Example 13 with the Pareto cone: front view

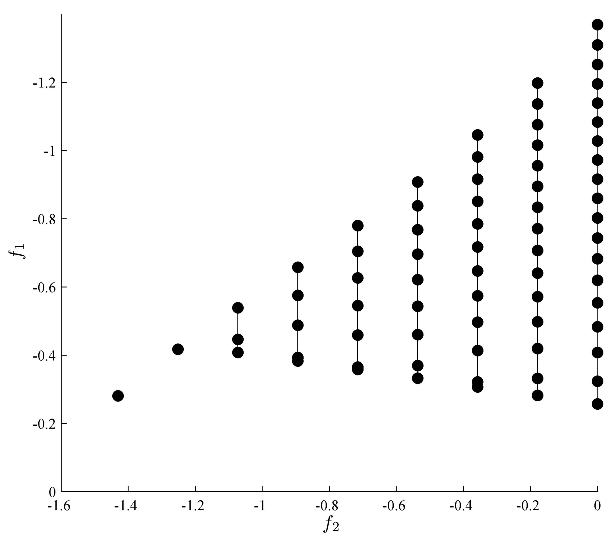

Figure 6.35: Example 13 with an obtuse cone: front view

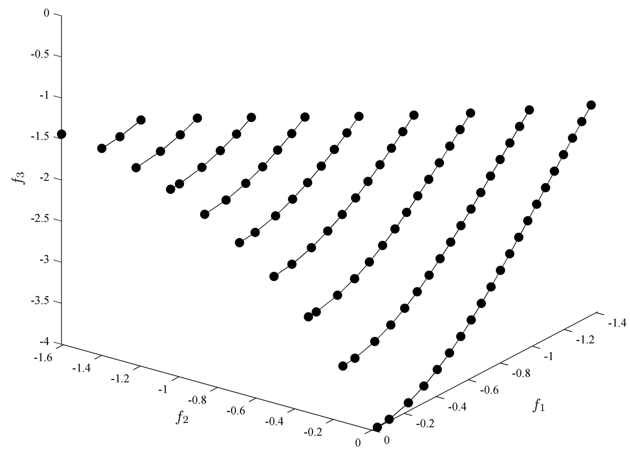

Figure 6.36: Example 13 with the Pareto cone: side view #1

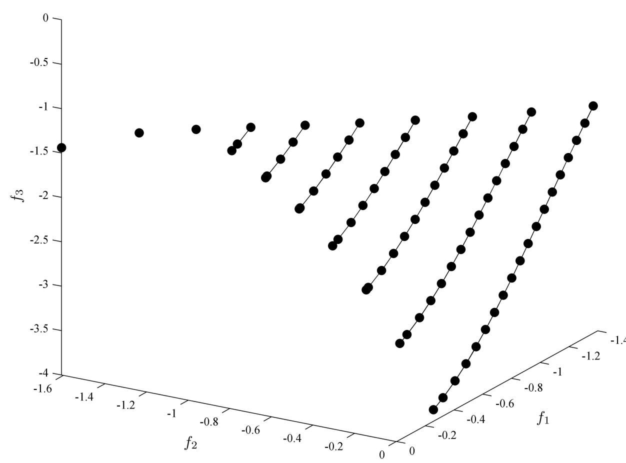

Figure 6.37: Example 13 with an obtuse cone: side view #1

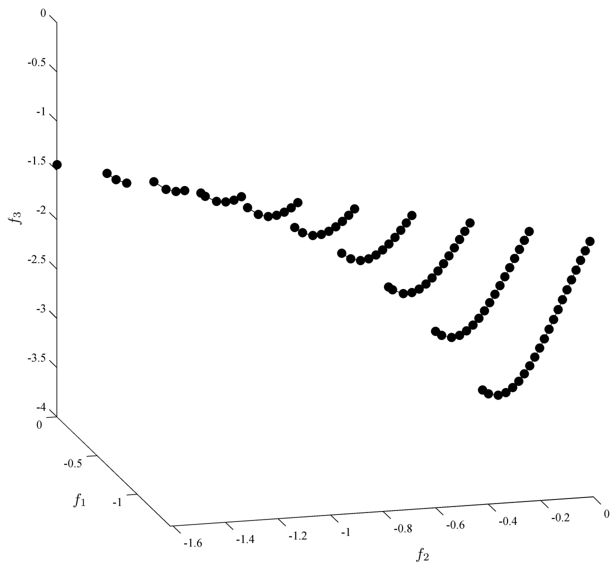

Figure 6.38: Example 13 with the Pareto cone: side view #2

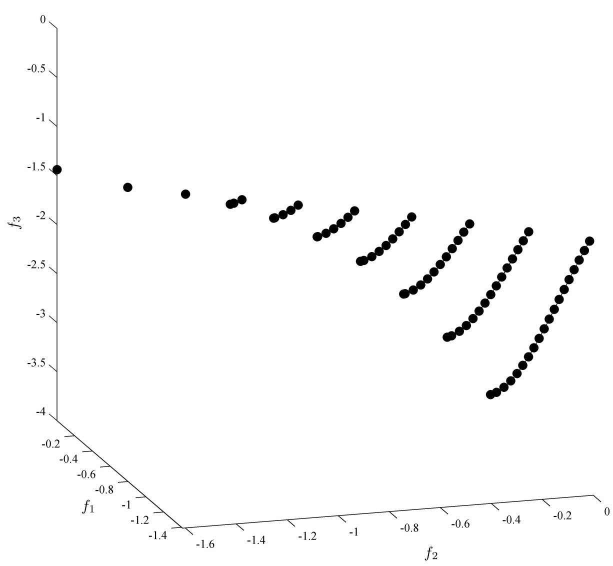

Figure 6.39: Example 13 with an obtuse cone: side view #2

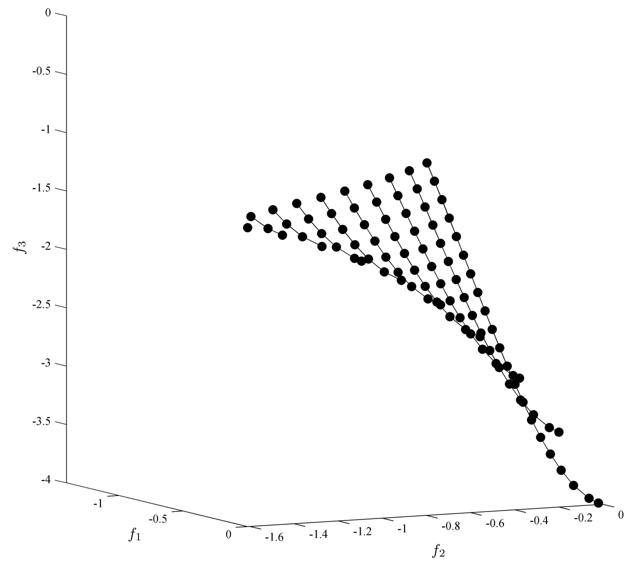

Figure 6.40: Example 13 with the Pareto cone: side view #3

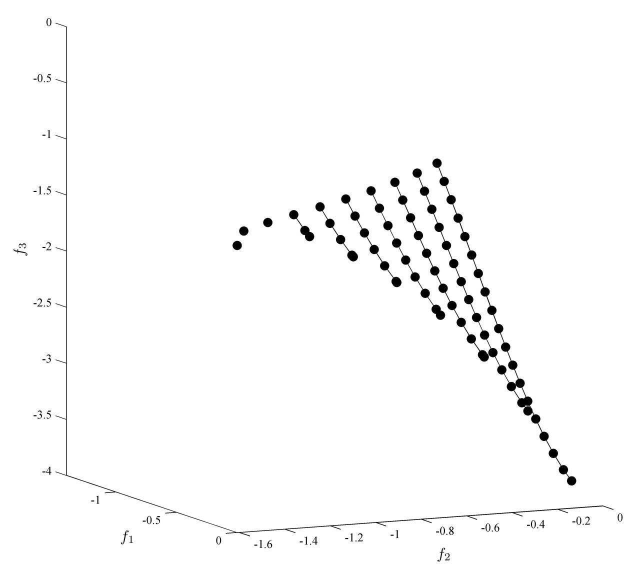

Figure 6.41: Example 13 with an obtuse cone: side view #3

Chapter 7

Conclusion and Further Research

7.1 Conclusion

This dissertation contains several theoretical contributions. In proving the correctness of the Constraint Controlled Spacing (CCS) method, we revealed the different behaviors of acute and obtuse simplicial cones in the general cone extension of the ϵ-constraint scalarization. Moreover, in the multicriteria CCS method, we proved the interesting property that any $p \times p$ matrix with nonnegative entries can be written as a positive definite matrix using only row exchanges. Combined with an earlier proposition which shows that any obtuse, polyhedral cone C, $\mathbb{R}^p_{\geqq} \subseteq C$, can be represented by a matrix having all nonnegative entries, the aforementioned result implies that any obtuse, simplicial cone can be represented by a positive definite matrix.

Methodologically, we presented two algorithms for generating quality discrete representations of the nondominated sets of multiobjective programs: the Bilevel Controlled Spacing (BCS) method and the CCS method. The BCS method uses a bilevel formulation to produce equidistant points: the upper-level problem controls the spacing, while the lower-level problem ensures that the point is nondominated. The CCS method, on the other hand, adds an additional constraint to the ϵ-constraint problem to control the spacing of each newly generated point. The CCS method is applicable to general MOPs while the

BCS method is limited to convex MOPs because of its use of the KKT optimality conditions. On the other hand, the CCS method is limited to notions of optimality defined by obtuse cones, whereas the BCS method can be used with both obtuse and acute cones. The BCS method is easily applied to problems with more than two objective functions and we presented two different approaches for doing so: the center method and the slicing method. On the other hand, although theoretically elegant, the CCS method as initially conceived encounters difficulties in higher dimensions. We presented a slightly altered formulation of the CCS method which integrates the BCS method to overcome these issues. Both methods integrate the cardinality preferences of the DM, although the BCS method is more restrictive. Also, the BCS method can be used to find a solution point with a specified tradeoff or a set of points with tradeoffs chosen by the DM. These extensions of the method promote the involvement of the DM in the optimization process which is important in multi-criterion decision making. Notice that the CCS and BCS methods complement one another in the sense that each is applicable in certain situations while the other is not.

In terms of computational work, both the BCS method and the CCS method show promise on a variety of test problems including linear, convex, nonconvex (CCS only), two-dimensional, and higher-dimensional problems. Previously in the literature, there has been limited testing of methods on higher dimensional problems although some approaches were theoretically applicable to problems with more than two dimensions. We found that the CCS method is much more computationally efficient. The computational issues of the BCS method are due in large part to the fact that the branch-and-bound method explicitly enumerates all possible solutions to the complementarity constraints in the KKT conditions. Thus, as the number of constraints grows, the branch-and-bound method becomes less efficient.

We also presented a comprehensive literature review covering both engineering and mathematical literature. The literature review contains measures proposed for assessing the quality of a discrete representation as well as methods that have been proposed for producing a representation satisfying certain quality criteria. Interestingly though, there is an apparent disconnect between these two areas. We found that the vast majority of quality measures were proposed in the field of engineering. Typically, these measures are used *a posteriori* to evaluate the performances of evolutionary algorithms, but they could just as well be applied to nondominated sets generated by exact algorithms. For the most part, however, these measures are either unknown or unused in the mathematical community. Within the proposed methods, several authors suggest filtering algorithms or other schemes to control the quality of a nondominated set after the outcome points are generated. This type of method seems inefficient because resources are used to generate a selection of nondominated points, some of which are later discarded. Why not incorporate a measure into the algorithm so these points are not produced in the first place? Further, methods which do not incorporate measures at all seem to improve the quality of discrete representations but only in a general sense. Usually, the quality of a resultant nondominated set cannot be explicitly stated or guaranteed. Both of these problems are solved by integrating measures into algorithms *a priori*. Each new point is generated so that when combined with the previously produced points, the updated nondominated set satisfies one or more prespecified quality criteria. Because of this, it is our opinion that the *a priori* class of methods is to be preferred. However, researchers have only recently begun formulating *a priori* methods, so many of these algorithms are limited to special classes of problems or encounter difficulties in higher dimensions.

7.2 Further Research

In both the Constraint Controlled Spacing method and the Bilevel Controlled Spacing method, we used the ϵ-constraint method as a scalarizing function because it is simple to implement and understand. Additionally, it completely characterizes efficiency. That is, any solution to an ϵ-constraint problem is at least weakly efficient and any weakly efficient solution can be found as the solution of some ϵ-constraint problem. However, other scalarizing approaches have this same property (see [28] or [66] for reviews). Thus, it would be interesting to explore whether these other scalarization methods can be integrated into our two approaches instead of the ϵ-constraint scalarization.

In our work with integrating the preferences of the DM through the use of general cones, we require the cones to be simplicial. In more than two dimensions though, there are many non-simplicial cones. For instance, it is not difficult to find a cone in three dimensions that has four (or more) extreme directions. Further study is needed to determine if these non-simplicial cones are compatible with our methods.

Along the same lines, in both methods, we assume that the chosen norm (in the constraint, in the CCS method, and in the objective, in the BCS method) is from the family of l_p-norms. However, there are other classes of norms besides the l_p-family. For the CCS method, at the end of Section 5.1, we consider elliptic norms in two dimensions. Other norms of interest could include block or polyhedral norms. It may also be possible to integrate the more general concept of gauges. A nice review of norms and gauges can be found in Chapter 2 of [70].

Recently, a new measure of the quality of the Pareto set has been proposed called robustness (see for example, [24] or [25]). Robustness measures the sensitivity of Pareto points to small changes in the decision space. Currently, the CCS and BCS methods generate discrete representations with good cardinality, complete coverage, and equidistant spacing. Integrating another measure, such as robustness, would be the next step in their development.

Other areas that could be investigated for both methods include: relaxing the requirement that solutions are Pareto optimal and allowing, instead, ϵ-Pareto solutions (see [47] and [60]); determining the applicability of our methods to integer or mixed-integer MOPs which arise frequently in applications (e.g., recently, [38] and [42]); and applying the developed methods to real-world problems such as an engineering design problem or a portfolio selection problem.

With respect to the BCS approach, the main disadvantage is that it can only be used on convex MOPs because we rewrite the lower-level ϵ-constraint problem in terms of its

KKT conditions to obtain a solvable single-level optimization problem. However, up until this reformulation, the BCS approach ((4.8) or (4.13)) is applicable to general problems. It is only our solution method that restricts its use. Thus, it would be of significant interest to explore other bilevel solution techniques such as the penalty method proposed by Bonnel and Morgan [14].

In terms of the CCS approach, further study is needed to determine the meaning behind the assumptions (see (5.65)) required for the proof of Theorem 5.3.3. Because the specified minors represent lower-dimensional projections of the cone C, our intuition is that these assumptions enforce some sort of necessary relationship between the projections, but this has yet to be verified.

Although there is plenty left to be pondered, researched, and improved in the field of multiple-objective optimization, and indeed, even in this specific area on which we chose to focus this dissertation, one has an overwhelming sense that the importance of this field and of MCDM as a whole has finally been realized and that widespread acceptance and use of MCDM techniques is inevitable, probably sooner rather than later.

Appendices

Appendix A Matlab Code for Example 2

In this appendix, we give the Matlab code for Example 2 (6.2), assuming the Pareto cone, in Chapter 6. The files `example2_f1.m` and `example2_f2.m` contain the objective functions. Note that the objective functions are negative because Example 2 is a maximization problem.

```
example2_f1.m

function z = example2_f1(x,y)

z = -(x+y);
```

```
example2_f2.m

function z = example2_f2(x,y)

z = -(10*x-x^2+4*y-y^2);
```

The file `example2_bilevel.m` implements the pseudocode in Figure 4.1 for Example 2. The first two optimizations (i.e., `fmincon(...)`) are to determine the initial reference points for the Bilevel Controlled Spacing algorithm. This is our implementation of the pseudocode found in Figure 4.2. The objective functions and constraints called by `fmincon`, are `example2_bilevel_subobj.m` and `example2_bilevel_subnlc.m`, respectively. These functions are shown below. In `example2_bilevel_subnlc.m`, `c` stores the inequality constraints while `ceq` stores the equality constraints (in this case, there are none). Following these initial optimizations, we declare any linear inequality ($\texttt{A}x \leq \texttt{b}$) or equality ($\texttt{Aeq}x = \texttt{beq}$) constraints, any lower (`lb`) or upper (`ub`) bounds on the variables, and a starting point (`x0`) for the algorithm. The solution algorithm we used is a branch-and-bound method proposed by Bard (see [3] or [73] for the pseudocode). Our nonlinear implementation (not shown here) is called using `branch_and_bound_nonlinear(...)`. As can be seen, we use a linked-list type structure to store the list of nondominated points throughout the algorithm. After the codes for `example2_bilevel.m`, `example2_bilevel_subobj.m`, and `example2_bilevel_subnlc.m` below, we discuss the objective function and constraints used in the branch-and-bound procedure.

```
example2_bilevel.m

clc;

clear all;


options = optimset('maxfunevals',1000000,'TolX',1e-8,'TolCon',1e-8,...

    'TolFun',1e-8,'MaxIter',100000);


x0 = [0; 0];


j = 1;

v = fmincon('example2_bilevel_subobj',x0,[],[],[],[],[],[],...

    'example2_bilevel_subnlc',options,j);

p0 = [example2_f1(v(1),v(2)); example2_f2(v(1),v(2))];

figure(1);

hold on;

plot(-p0(1),-p0(2),'-or');


j = 2;

v = fmincon('example2_bilevel_subobj',x0,[],[],[],[],[],[],...

    'example2_bilevel_subnlc',options,j);

q0 = [example2_f1(v(1),v(2)); example2_f2(v(1),v(2))];

figure(1);

hold on;
```

```
plot(-q0(1),-q0(2),'-or');

A = [];
b = [];
Aeq = [];
beq = [];
lb = [];
ub = [];
x0 = [0;0;0;0;0;0;0;0];

p = p0;
clear list;
list(1) = libpointer('doublePtr',q0);
k = size(list,2);
N = 17;

while ((k+1) < N)

    for i=1:k

        q = get(list(i),'value');
        [z,zval] = branch_and_bound_nonlinear('example2_bilevel_obj',...
            'example2_bilevel_nlc',A,b,Aeq,beq,lb,ub,x0,p,q);
        x1 = z(2);
        x2 = z(3);
        p1 = [example2_f1(x1,x2); example2_f2(x1,x2)];
        figure(1);
        hold on;
```

```
        plot(-p1(1), -p1(2), '-ok');

        newlist(2*i-1) = libpointer('doublePtr',p1);

        newlist(2*i) = libpointer('doublePtr',q);

        p = q;

    end

    list = newlist;

    clear newlist;

    k = size(list,2);

    p = p0;

end

h(1,:)=p0';

for l=1:size(list,2)

    h(l+1,:)=get(list(l),'value');

end

figure(1);

plot(-h(:,1),-h(:,2),'-ok');
```

```
example2_bilevel_subobj.m

function z = example2_bilevel_subobj(x,j)

x1 = x(1);

x2 = x(2);
```

```
f = [example2_f1(x1,x2); example2_f2(x1,x2)];

if (j == 1)
    z = f(1);
else
    z = f(2);
end
```

```
example2_bilevel_subnlc.m
function [c,ceq] = example2_bilevel_subnlc(x,j)

x1 = x(1);
x2 = x(2);

c = [3*x1+x2-12; 2*x1+x2-9; x1+2*x2-12];

ceq = 0;
```

The functions `example2_bilevel_obj.m` and `example2_bilevel_nlc.m`, shown below, are the objective and constraint functions of the branch-and-bound algorithm, respectively. Recall that to use the branch-and-bound procedure, we first reformulate the lower-level problem using the ϵ-constraint scalarization and then rewrite this new problem according to its KKT conditions (see Chapter 4). This increases the number of optimization variables from two (`x1`, `x2`) to eight (`t`, `x1`, `x2`, `e`, `u1`, `u2`, `u3`, `u4`) as is shown in the beginning of both `example2_bilevel_obj.m` and `example2_bilevel_nlc.m` below. The variables `p` and `q` which are input into both functions are the reference nondominated points between which we would like to find a new nondominated point in each iteration. The objective is to minimize `t` since `t` represents the maximum of the distances from `p` to the new point and from `q` to the new point. The function `example2_bilevel_nlc.m` contains the inequality (`c`) and equality (`ceq`) constraints for the problem. It also contains the complementary slackness conditions (`CS`) which need to be satisfied. The vector `P` is used to store the branching tree that is formed while using the branch-and-bound algorithm, and the for-loop at the end of `example2_bilevel_nlc.m` utilizes the information in `P` to append additional equality constraints to `ceq`.

```
example2_bilevel_obj.m
function z = example2_bilevel_obj(x,P,p,q)

t = x(1);
x1 = x(2);
x2 = x(3);
e = x(4);
u1 = x(5);
u2 = x(6);
u3 = x(7);
u4 = x(8);

z = t;
```

```
example2_bilevel_nlc.m
function [c,ceq_new] = example2_bilevel_nlc(x,P,p,q)

global CS;
```

```
t = x(1);
x1 = x(2);
x2 = x(3);
e = x(4);
u1 = x(5);
u2 = x(6);
u3 = x(7);
u4 = x(8);

f = [example2_f1(x1,x2); example2_f2(x1,x2)];

c = [norm(f-p,1)-t; norm(f-q,1)-t; f(1)-e; 3*x1+x2-12; 2*x1+x2-9;...
    x1+2*x2-12; -u1; -u2; -u3; -u4];

ceq = [-u1-10+2*x1+3*u2+2*u3+u4; -u1-4+2*x2+u2+u3+2*u4];

CS = [u1; u2; u3; u4; f(1)-e; 3*x1+x2-12; 2*x1+x2-9; x1+2*x2-12];

n = size(CS,1);
a = [];

l=1;
for i=1:size(P,2)
    if (P(i)>0)
        a(l,1) = CS(P(i));
        l=l+1;
    else if (P(i)<0)
            a(l,1) = CS(n/2-P(i));
```

```
            l=l+1;
        end
    end
end

ceq_new = vertcat(ceq,a);
```

Appendix B Matlab Code for Example 7

In this appendix, we give the Matlab code for Example 7 (6.3), assuming the Pareto cone, in Chapter 6. The files `example7_f1.m` and `example7_f2.m` contain the objective functions.

```
example7_f1.m
function z = example7_f1(x,y)
z = sqrt(1+x^2);
```

```
example7_f2.m
function z = example7_f2(x,y)
z = x^2-4*x+y+5;
```

The file `example7_constraint.m` implements the pseudocode in Figure 5.7 without the additional check step. We input the reference points `a0` and `b0`, the desired spacing value `d`, the initial nondominated point `a`, and the initial epsilon `e`. Given these inputs, `fmincon` calls the objective function `example7_constraint_obj.m` and the nonlinear constraint function `example7_constraint_nlc`. These functions are shown below. In `example7_constraint_nlc.m`, `c` stores the inequality constraints while `ceq` stores the equality constraints (in this case, there are none). Notice that we have removed the absolute value from the norm constraint in `c` because we are taking advantage of the two-dimensional structure of the problem.

```
example7_constraint.m
clc;
clear all;
```

```
options = optimset('maxfunevals',1000000,'TolX',1e-10,'TolCon',1e-10,...
'TolFun',1e-10,'MaxIter',1000000);

a0 = [1.0842; 3.5000];
b0 = [2.2361; 1.0000];

f1(17) = 1.0842;
f2(17) = 3.5000;
f1(1) = 2.2361;
f2(1) = 1.0000;

figure(1);
hold on;
plot(a0(1), a0(2),'-.ok');
plot(b0(1), b0(2),'-.ok');

v0=[1.0842; 3.5000];

d = 0.22824375;

e = 2.2361;
a = b0;
i=1;
N=17;

while ((i<=N-2) && (norm(a0-a,1)>=d))
```

```
    v = fmincon('example7_constraint_obj',v0,[],[],[],[],[],[],...
     'example7_constraint_nlc',options,a,e,d);

    f1(i+1) = example7_f1(v(1),v(2));
    f2(i+1) = example7_f2(v(1),v(2));

    a = [f1(i+1); f2(i+1)];
    e = f1(i+1);
    v0 = v;

    figure(1);
    hold on;
    plot(f1(i+1), f2(i+1),'-.ok');

    i = i+1;

end

figure(1);
hold on;
plot(f1,f2,'-.ok');
```

```
example7_constraint_obj.m
function y = example7_constraint_obj(v,a,e,d)

x1 = v(1);
x2 = v(2);
```

```
y = example7_f2(x1,x2);
```

---

example7_constraint_nlc.m

```
function [c,ceq] = example7_constraint_nlc(v,a,e,d)


x1 = v(1);

x2 = v(2);


f = [example7_f1(x1,x2); example7_f2(x1,x2)];


c = [f(1)-e; d-(a(1)-f(1)+f(2)-a(2)); f(2)-3.5; -x1; -x2];


ceq = 0;
```

Appendix C Solution Points for Selected Examples

This appendix contains sample solution points for various test problems chosen from Chapter 6. All solution points given are with respect to the Pareto cone. Additionally, if the objective functions are equal to the decision variables or the negatives of the decision variables, then we have not included the values of the objective functions in the table.

x_1	x_2	f_1	f_2
1.0000	5.0000	-11.0000	-8.0000
2.6667	3.3333	-9.3333	-11.3333
1.8333	4.1667	-10.1667	-9.6667
3.3750	2.2500	-7.8750	-12.3750
1.4167	4.5833	-10.5833	-8.8333
2.2500	3.7500	-9.7500	-10.5000
3.0625	2.8750	-8.8125	-12.0625
3.6875	1.6250	-6.9375	-12.6875
1.2083	4.7917	-10.7917	-8.4167
1.6250	4.3750	-10.3750	-9.2500
2.0417	3.9583	-9.9583	-10.0833
2.4583	3.5417	-9.5417	-10.9167
2.8750	3.1250	-9.1250	-11.7500
3.2188	2.5625	-8.3438	-12.2188
3.5313	1.9375	-7.4063	-12.5313
3.8438	1.3125	-6.4688	-12.8438
4.0000	1.0000	-6.0000	-13.0000

Table C-1: Efficient and Pareto points for Example 1

x_1	x_2	f_1	f_2
3.5000	1.5000	-5.0000	-26.5000
2.5362	3.9276	6.4638	19.2138
2.9075	3.1849	6.0925	23.2175
2.2462	4.5076	6.7538	15.1288
3.1217	2.6348	5.7566	25.0691
2.7073	3.5854	6.2927	21.2302
2.3842	4.2315	6.6158	17.1783
2.1188	4.7623	6.8812	13.0687
3.2545	2.2365	5.4910	25.8973
3.0185	2.9445	5.9630	24.1817
2.8028	3.3943	6.1972	22.2284
2.6189	3.7622	6.3811	20.2249
2.4582	4.0836	6.5418	18.1980
2.3137	4.3725	6.6863	16.1550
2.1814	4.6373	6.8186	14.0999
2.0585	4.8831	6.9415	12.0353
2.0000	5.0000	-7.0000	-11.0000

Table C-2: Efficient and Pareto points for Example 2

x_1	x_2	x_3
3.0000	0.0000	0.0000
0.0000	3.0000	0.0000
0.0000	0.0000	3.0000
1.0000	1.0000	1.0000
1.5000	1.5000	0.0000
1.5000	0.0000	1.5000
0.0000	1.5000	1.5000
2.0000	0.5000	0.5000
0.5001	2.0000	0.4999
0.5000	0.5000	2.0000
2.2500	0.7500	0.0000
2.2500	0.0000	0.7500
1.5000	1.0000	0.5000
1.5000	0.5000	1.0000
0.7500	2.2500	0.0000
0.0000	2.2500	0.7500
1.0000	1.5000	0.5000
0.5000	1.5000	1.0000
0.5000	1.0000	1.5000
1.0000	0.5000	1.5000
0.0000	0.7500	2.2500
0.7500	0.0000	2.2500

Table C-3: Efficient points for Example 4 using the center technique

x_1	x_2	x_3
0.0000	0.0000	3.0000
0.7500	0.0000	2.2500
0.0000	0.7500	2.2500
0.3750	0.3750	2.2500
0.5625	0.1875	2.2500
0.1875	0.5625	2.2500
1.5000	0.0000	1.5000
0.0000	1.5000	1.5000
0.7500	0.7500	1.5000
1.1250	0.3750	1.5000
0.3750	1.1250	1.5000
1.3125	0.1875	1.5000
0.9375	0.5625	1.5000
0.5625	0.9375	1.5000
0.1875	1.3125	1.5000
2.2500	0.0000	0.7500
0.0000	2.2500	0.7500
1.1250	1.1250	0.7500
1.6875	0.5625	0.7500
0.5625	1.6875	0.7500
1.9688	0.2812	0.7500
1.4062	0.8437	0.7500
0.8438	1.4062	0.7500
0.2812	1.9687	0.7500
3.0000	0.0000	0.0000
0.0000	3.0000	0.0000
1.5000	1.5000	0.0000
2.2500	0.7500	0.0000
0.7500	2.2500	0.0000
2.6250	0.3750	0.0000
1.8750	1.1250	0.0000
1.1250	1.8750	0.0000
0.3750	2.6250	0.0000

Table C-4: Efficient points for Example 4 using the slicing technique

x_1	x_2	x_3	f_1	f_2	f_3
0.0000	0.0000	3.0000	-6.0000	3.0000	-12.0000
0.1667	0.1667	2.6667	-6.1667	2.0000	-10.6667
0.0000	0.4444	2.5556	-5.5556	1.2222	-10.6667
0.3333	0.3333	2.3333	-6.3333	1.0000	-9.3333
0.0000	0.8889	2.1111	-5.1111	-0.5556	-9.3333
0.1667	0.6111	2.2222	-5.7222	0.2222	-9.3333
0.2500	0.4722	2.2778	-6.0278	0.6111	-9.3333
0.0833	0.7500	2.1667	-5.4167	-0.1667	-9.3333
0.5000	0.5000	2.0000	-6.5000	0.0000	-8.0000
0.0000	1.1429	1.7143	-4.5714	-1.7143	-8.0000
0.2500	0.8214	1.8571	-5.5357	-0.8571	-8.0000
0.3750	0.6607	1.9286	-6.0179	-0.4286	-8.0000
0.1250	0.9821	1.7857	-5.0536	-1.2857	-8.0000
0.5833	0.5833	1.6667	-6.2500	-0.6667	-6.6667
0.0000	1.3333	1.3333	-4.0000	-2.6667	-6.6667
0.2917	0.9583	1.5000	-5.1250	-1.6667	-6.6667
0.4375	0.7708	1.5833	-5.6875	-1.1667	-6.6667
0.1458	1.1458	1.4167	-4.5625	-2.1667	-6.6667
0.6667	0.6667	1.3333	-6.0000	-1.3333	-5.3333
0.0000	1.5238	0.9524	-3.4286	-3.6190	-5.3333
0.3333	1.0952	1.1429	-4.7143	-2.4762	-5.3333
0.5000	0.8810	1.2381	-5.3571	-1.9048	-5.3333
0.1667	1.3095	1.0476	-4.0714	-3.0476	-5.3333
0.7500	0.7500	1.0000	-5.7500	-2.0000	-4.0000
0.0000	1.7143	0.5714	-2.8571	-4.5714	-4.0000
0.3750	1.2321	0.7857	-4.3036	-3.2857	-4.0000
0.5625	0.9911	0.8929	-5.0268	-2.6429	-4.0000
0.1875	1.4732	0.6786	-3.5804	-3.9286	-4.0000
0.6563	0.8705	0.9464	-5.3884	-2.3214	-4.0000
0.4687	1.1116	0.8393	-4.6652	-2.9643	-4.0000
0.2812	1.3527	0.7321	-3.9420	-3.6071	-4.0000
0.0937	1.5938	0.6250	-3.2187	-4.2500	-4.0000
0.8333	0.8333	0.6667	-5.5000	-2.6667	-2.6667
0.0000	1.9048	0.1905	-2.2857	-5.5238	-2.6667
0.4167	1.3690	0.4286	-3.8929	-4.0952	-2.6667
0.6250	1.1012	0.5476	-4.6964	-3.3810	-2.6667
0.2083	1.6369	0.3095	-3.0893	-4.8095	-2.6667
0.7292	0.9673	0.6071	-5.0982	-3.0238	-2.6667
0.5208	1.2351	0.4881	-4.2946	-3.7381	-2.6667
0.3125	1.5030	0.3690	-3.4911	-4.4524	-2.6667
0.1042	1.7708	0.2500	-2.6875	-5.1667	-2.6667

Table C-5: Efficient and Pareto points for Example 5 using the slicing technique

x_1	x_2	x_3	f_1	f_2	f_3
0.9167	0.9167	0.3333	-5.2500	-3.3333	-1.3333
0.3333	1.6667	0.0000	-3.0000	-5.3333	-1.3333
0.6250	1.2917	0.1667	-4.1250	-4.3333	-1.3333
0.7708	1.1042	0.2500	-4.6875	-3.8333	-1.3333
0.4792	1.4792	0.0833	-3.5625	-4.8333	-1.3333
1.0000	1.0000	0.0000	-5.0000	-4.0000	0.0000
1.0000	1.0000	0.0000	-5.0000	-4.0000	0.0000

Table C-6: Efficient and Pareto points for Example 5 using the slicing technique (continued)

x_1	x_2	x_3
1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000
0.5774	0.5774	0.5774
0.7071	0.7071	0.0007
0.7071	0.0007	0.7071
0.0283	0.7068	0.7068
0.8865	0.3669	0.2820
0.8865	0.2820	0.3669
0.2932	0.8872	0.3562
0.3669	0.8865	0.2820
0.3669	0.2820	0.8865
0.2932	0.3562	0.8872
0.9239	0.3827	0.0004
0.7168	0.6282	0.3027
0.3827	0.9239	0.0005
0.6282	0.7168	0.3027
0.3153	0.7256	0.6116
0.0420	0.9235	0.3814
0.3153	0.6116	0.7256
0.0417	0.3814	0.9235
0.3827	0.0005	0.9239
0.6282	0.3027	0.7168
0.7168	0.3027	0.6282
0.9239	0.0004	0.3827

Table C-7: Efficient points for Example 13 using the center technique

x_1	x_2	x_3
0.0000	0.0000	1.0000
0.5151	0.0000	0.8571
0.0338	0.5140	0.8571
0.3760	0.3521	0.8571
0.4790	0.1893	0.8571
0.2203	0.4656	0.8571
0.6999	0.0000	0.7143
0.0427	0.6986	0.7143
0.5097	0.4795	0.7143
0.6506	0.2579	0.7143
0.2971	0.6337	0.7143
0.6874	0.1313	0.7143
0.5907	0.3754	0.7143
0.4107	0.5667	0.7143
0.1730	0.6781	0.7143
0.8207	0.0000	0.5714
0.0451	0.8194	0.5714
0.5960	0.5641	0.5714
0.7624	0.3036	0.5714
0.3450	0.7446	0.5714
0.8060	0.1546	0.5714
0.6916	0.4418	0.5714
0.4791	0.6663	0.5714
0.1986	0.7963	0.5714
0.9035	0.0000	0.4286
0.0465	0.9023	0.4286
0.6551	0.6222	0.4286
0.8391	0.3350	0.4286
0.3778	0.8207	0.4286
0.8873	0.1706	0.4286
0.7608	0.4874	0.4286
0.5259	0.7347	0.4286
0.2160	0.8773	0.4286
0.9583	0.0000	0.2857
0.0473	0.9571	0.2857
0.6942	0.6607	0.2857
0.8898	0.3558	0.2857
0.3993	0.8712	0.2857
0.9410	0.1812	0.2857
0.8065	0.5176	0.2857
0.5568	0.7800	0.2857
0.2274	0.9309	0.2857

Table C-8: Efficient points for Example 6 using the slicing technique

x_1	x_2	x_3
0.9897	0.0000	0.1429
0.0480	0.9886	0.1429
0.7166	0.6827	0.1429
0.9189	0.3676	0.1429
0.4117	0.9000	0.1429
0.9719	0.1872	0.1429
0.8328	0.5348	0.1429
0.5746	0.8059	0.1429
0.2341	0.9617	0.1429
1.0000	0.0000	0.0000
0.0478	0.9989	0.0000
0.7238	0.6900	0.0000
0.9284	0.3716	0.0000
0.4156	0.9096	0.0000
0.9819	0.1892	0.0000
0.8413	0.5406	0.0000
0.5802	0.8145	0.0000
0.2360	0.9718	0.0000

Table C-9: Efficient points for Example 6 using the slicing technique (continued)

x_1	x_2	f_1	f_2
2.0000	0.0000	2.2361	1.0000
1.7912	0.0000	2.0514	1.0436
1.6326	0.0000	1.9146	1.1349
1.4992	0.0000	1.8021	1.2508
1.3816	0.0000	1.7055	1.3824
1.2751	0.0000	1.6204	1.5255
1.1768	0.0000	1.5443	1.6777
1.0850	0.0000	1.4756	1.8372
0.9985	0.0000	1.4132	2.0030
0.9163	0.0000	1.3563	2.1744
0.8377	0.0000	1.3045	2.3509
0.7623	0.0000	1.2574	2.5320
0.6895	0.0000	1.2147	2.7175
0.6190	0.0000	1.1761	2.9071
0.5506	0.0000	1.1415	3.1009
0.4839	0.0000	1.1109	3.2985
0.4189	0.0000	1.0842	3.5000

Table C-10: Efficient and Pareto points for Example 7 with $N = 17$

x_1	x_2	f_1	f_2
1.2500	1.2500	7.8906	16.1250
1.2913	1.2913	7.9263	15.8391
1.3270	1.3270	8.0065	15.5978
1.3597	1.3597	8.1115	15.3812
1.3906	1.3906	8.2321	15.1802
1.4205	1.4205	8.3632	14.9898
1.4498	1.4498	8.5015	14.8065
1.4788	1.4788	8.6444	14.6278
1.5080	1.5080	8.7901	14.4520
1.3070	1.8161	8.9275	14.2678
1.2545	2.0417	8.9463	13.9650
2.1170	1.2962	8.9853	13.6825
1.3468	2.1531	9.0748	13.4504
1.3985	2.1733	9.1940	13.2480
1.4515	2.1848	9.3298	13.0623
1.5071	2.1899	9.4742	12.8851
1.5672	2.1892	9.6208	12.7102
1.6357	2.1816	9.7634	12.5312
1.7205	2.1624	9.8925	12.3387
1.8520	2.1091	9.9868	12.1115
2.0498	2.0498	10.0026	11.8057
2.1262	2.1262	10.0453	11.5269
2.1881	2.1881	10.1582	11.3182
2.2365	2.2365	10.3273	11.1657
2.2754	2.2754	10.5331	11.0500
2.3078	2.3078	10.7629	10.9582
2.3356	2.3356	11.0092	10.8830
2.3599	2.3599	11.2674	10.8196
2.3815	2.3815	11.5344	10.7650
2.4011	2.4011	11.8083	10.7174
2.4190	2.4190	12.0876	10.6751
2.4355	2.4355	12.3714	10.6373
2.4508	2.4508	12.6588	10.6032
2.4652	2.4652	12.9493	10.5721

Table C-11: Efficient and Pareto points for Example 8

x_1	x_2
0.0417	1.0384
0.0906	1.0023
0.1337	0.9604
0.1882	0.9299
0.4464	0.9298
0.4750	0.8734
0.5021	0.8155
0.5501	0.7785
0.6297	0.7731
0.7052	0.7636
0.7534	0.7268
0.7720	0.6604
0.7740	0.5774
0.7977	0.5161
0.8510	0.4844
0.9099	0.4583
0.9290	0.1996
0.9546	0.1402
0.9963	0.0969
1.0384	0.0417

Table C-12: Efficient points for Example 9

x_1	x_2	f_1	f_2
1.0000	1.0000	-1.0000	1.0350
0.8994	0.8090	-0.8994	1.0045
0.8965	0.8037	-0.8965	0.8763
0.8931	0.7977	-0.8931	0.7485
0.8890	0.7904	-0.8890	0.6215
0.8823	0.7785	-0.8823	0.4971
0.7713	0.5949	-0.7713	0.4770
0.7679	0.5897	-0.7679	0.3493
0.7638	0.5834	-0.7638	0.2223
0.7571	0.5732	-0.7571	0.0979
0.6433	0.4138	-0.6433	0.0807
0.6393	0.4087	-0.6393	-0.0465
0.6335	0.4013	-0.6335	-0.1718
0.5163	0.2666	-0.5163	-0.1887
0.5119	0.2620	-0.5119	-0.3154
0.5021	0.2521	-0.5021	-0.4340

Table C-13: Efficient and Pareto points for Example 10

x_1	x_2	x_3
0.0000	0.0000	-10.0000
-0.5976	0.0000	-8.5714
-0.0092	-1.1951	-8.5714
-0.5806	-0.2830	-8.5714
-0.5340	-0.5364	-8.5714
-0.4602	-0.7626	-8.5714
-0.3567	-0.9590	-8.5714
-0.2138	-1.1162	-8.5714
-0.8452	0.0000	-7.1429
-0.0265	-1.6895	-7.1429
-0.8328	-0.2877	-7.1429
-0.7986	-0.5534	-7.1429
-0.7446	-0.7995	-7.1429
-0.6715	-1.0264	-7.1429
-0.5781	-1.2330	-7.1429
-0.4614	-1.4162	-7.1429
-0.3143	-1.5691	-7.1429
-0.1187	-1.6736	-7.1429
-1.0351	0.0000	-5.7143
-0.0170	-2.0699	-5.7143
-1.0249	-0.2898	-5.7143
-0.9963	-0.5612	-5.7143
-0.9513	-0.8162	-5.7143
-0.8905	-1.0554	-5.7143
-0.8140	-1.2789	-5.7143
-0.7208	-1.4857	-5.7143
-0.6090	-1.6739	-5.7143
-0.4747	-1.8396	-5.7143
-0.3102	-1.9751	-5.7143
-0.0963	-2.0612	-5.7143

Table C-14: Efficient points for Example 12

x_1	x_2	x_3
-1.1952	0.0000	-4.2857
-0.0110	-2.3904	-4.2857
-1.1863	-0.2911	-4.2857
-1.1612	-0.5660	-4.2857
-1.1215	-0.8263	-4.2857
-1.0681	-1.0729	-4.2857
-1.0011	-1.3059	-4.2857
-0.9204	-1.5251	-4.2857
-0.8250	-1.7297	-4.2857
-0.7133	-1.9181	-4.2857
-0.5825	-2.0873	-4.2857
-0.4275	-2.2323	-4.2857
-0.2379	-2.3426	-4.2857
-1.3363	0.0000	-2.8571
-0.0005	-2.6726	-2.8571
-1.3283	-0.2920	-2.8571
-1.3056	-0.5693	-2.8571
-1.2697	-0.8334	-2.8571
-1.2212	-1.0849	-2.8571
-1.1607	-1.3244	-2.8571
-1.0880	-1.5517	-2.8571
-1.0028	-1.7665	-2.8571
-0.9042	-1.9679	-2.8571
-0.7908	-2.1545	-2.8571
-0.6601	-2.3238	-2.8571
-0.5081	-2.4718	-2.8571
-0.3274	-2.5911	-2.8571
-0.1012	-2.6649	-2.8571

Table C-15: Efficient points for Example 12 (continued)

x_1	x_2	x_3
-1.4639	0.0000	-1.4286
-0.0072	-2.9277	-1.4286
-1.4565	-0.2927	-1.4286
-1.4357	-0.5718	-1.4286
-1.4025	-0.8387	-1.4286
-1.3578	-1.0940	-1.4286
-1.3020	-1.3381	-1.4286
-1.2352	-1.5713	-1.4286
-1.1571	-1.7933	-1.4286
-1.0674	-2.0035	-1.4286
-0.9651	-2.2013	-1.4286
-0.8489	-2.3851	-1.4286
-0.7167	-2.5528	-1.4286
-0.5648	-2.7010	-1.4286
-0.3872	-2.8234	-1.4286
-0.1714	-2.9076	-1.4286
-1.5811	0.0000	0.0000
-0.0003	-3.1623	0.0000
-1.5743	-0.2932	0.0000
-1.5549	-0.5738	0.0000
-1.5239	-0.8428	0.0000
-1.4822	-1.1011	0.0000
-1.4301	-1.3489	0.0000
-1.3677	-1.5866	0.0000
-1.2951	-1.8140	0.0000
-1.2120	-2.0308	0.0000
-1.1178	-2.2366	0.0000
-1.0116	-2.4304	0.0000
-0.8921	-2.6109	0.0000
-0.7572	-2.7761	0.0000
-0.6038	-2.9226	0.0000
-0.4263	-3.0452	0.0000
-0.2143	-3.1331	0.0000

Table C-16: Efficient points for Example 12 (continued)

x_1	x_2	x_3	f_3
0.0000	1.6094	1.2000	-1.4400
0.2808	1.4306	1.2000	-1.4400
0.0355	1.4306	1.2385	-1.5340
0.1490	1.4306	1.2281	-1.5082
0.4177	1.2518	1.2000	-1.4400
0.0349	1.2518	1.2854	-1.6522
0.3107	1.2518	1.2381	-1.5329
0.1870	1.2518	1.2686	-1.6092
0.5394	1.0730	1.2000	-1.4400
0.0347	1.0730	1.3414	-1.7993
0.4467	1.0730	1.2439	-1.5472
0.3454	1.0730	1.2829	-1.6459
0.2302	1.0730	1.3156	-1.7308
0.0896	1.0730	1.3380	-1.7902
0.6583	0.8941	1.2000	-1.4400
0.0315	0.8941	1.4085	-1.9838
0.5755	0.8941	1.2479	-1.5572
0.4881	0.8941	1.2922	-1.6698
0.3937	0.8941	1.3325	-1.7754
0.2886	0.8941	1.3676	-1.8703
0.1653	0.8941	1.3953	-1.9470
0.7799	0.7153	1.2000	-1.4400
0.0339	0.7153	1.4885	-2.2155
0.7046	0.7153	1.2509	-1.5647
0.6270	0.7153	1.2989	-1.6870
0.5457	0.7153	1.3438	-1.8058
0.4593	0.7153	1.3854	-1.9194
0.3652	0.7153	1.4231	-2.0252
0.2590	0.7153	1.4557	-2.1190
0.1315	0.7153	1.4804	-2.1916
0.9082	0.5365	1.2000	-1.4400
0.0336	0.5365	1.5842	-2.5098
0.8387	0.5365	1.2532	-1.5706
0.7683	0.5365	1.3039	-1.7001
0.6962	0.5365	1.3521	-1.8281
0.6217	0.5365	1.3977	-1.9535
0.5436	0.5365	1.4406	-2.0755
0.4605	0.5365	1.4806	-2.1923
0.3699	0.5365	1.5172	-2.3018
0.2679	0.5365	1.5491	-2.3998
0.1461	0.5365	1.5741	-2.4779

Table C-17: Efficient and Pareto points for Example 13

x_1	x_2	x_3	f_3
1.0464	0.3577	1.2000	-1.4400
0.0333	0.3577	1.6988	-2.8858
0.9816	0.3577	1.2550	-1.5751
0.9167	0.3577	1.3078	-1.7103
0.8513	0.3577	1.3583	-1.8449
0.7851	0.3577	1.4067	-1.9787
0.7173	0.3577	1.4529	-2.1109
0.6474	0.3577	1.4970	-2.2410
0.5744	0.3577	1.5388	-2.3680
0.4972	0.3577	1.5782	-2.4908
0.4140	0.3577	1.6148	-2.6076
0.3220	0.3577	1.6479	-2.7156
0.2158	0.3577	1.6761	-2.8094
0.0825	0.3577	1.6959	-2.8761
1.1985	0.1788	1.2000	-1.4400
0.0331	0.1788	1.8357	-3.3698
1.1371	0.1788	1.2564	-1.5786
1.0765	0.1788	1.3107	-1.7179
1.0161	0.1788	1.3629	-1.8576
0.9558	0.1788	1.4132	-1.9973
0.8951	0.1788	1.4617	-2.1366
0.8337	0.1788	1.5084	-2.2752
0.7712	0.1788	1.5533	-2.4127
0.7072	0.1788	1.5965	-2.5487
0.6409	0.1788	1.6378	-2.6824
0.5717	0.1788	1.6773	-2.8132
0.4984	0.1788	1.7146	-2.9399
0.4195	0.1788	1.7496	-3.0610
0.3323	0.1788	1.7815	-3.1738
0.2324	0.1788	1.8094	-3.2739
0.1087	0.1788	1.8304	-3.3502

Table C-18: Efficient and Pareto points for Example 13 (continued)

x_1	x_2	x_3	f_3
1.3694	0.0000	1.2000	-1.4400
0.0329	0.0000	1.9995	-3.9978
1.3105	0.0000	1.2574	-1.5810
1.2527	0.0000	1.3127	-1.7233
1.1959	0.0000	1.3662	-1.8665
1.1397	0.0000	1.4178	-2.0103
1.0840	0.0000	1.4678	-2.1545
1.0284	0.0000	1.5162	-2.2989
0.9727	0.0000	1.5631	-2.4432
0.9166	0.0000	1.6085	-2.5872
0.8600	0.0000	1.6524	-2.7306
0.8024	0.0000	1.6950	-2.8730
0.7435	0.0000	1.7361	-3.0141
0.6828	0.0000	1.7758	-3.1534
0.6199	0.0000	1.8140	-3.2904
0.5538	0.0000	1.8505	-3.4244
0.4837	0.0000	1.8853	-3.5543
0.4079	0.0000	1.9179	-3.6785
0.3240	0.0000	1.9480	-3.7946
0.2273	0.0000	1.9743	-3.8978
0.1067	0.0000	1.9943	-3.9773

Table C-19: Efficient and Pareto points for Example 13 (continued)

Bibliography

[1] L. J. Adams and Paul A. White. *Analytic Geometry and Calculus*. Oxford University Press, second edition, 1968.

[2] R. Armann. Solving multiobjective programming problems by discrete representation. *Optimization*, 20:483–492, 1989.

[3] Jonathan F. Bard. *Practical Bilevel Optimization: Algorithms and Applications*. Kluwer Academic Publishers, 1998.

[4] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Inc., 2nd edition, 1993.

[5] Harold P. Benson. An algorithm for optimizing over the weakly efficient set. *European Journal of Operational Research*, 25:192–199, 1986.

[6] Harold P. Benson. A finite, nonadjacent extreme-point search algorithm for optimization over the efficient set. *Journal of Optimization Theory and Applications*, 73(1):47–64, April 1992.

[7] Harold P. Benson. A bisection-extreme point search algorithm for optimizing over the efficient set in the linear dependence case. *Journal of Global Optimization*, 3:95–111, 1993.

[8] Harold P. Benson and D. Lee. Outcome-based algorithm for optimizing over the efficient set of a bicriteria linear programming problem. *Journal of Optimization Theory and Applications*, 88(1):77–105, January 1996.

[9] Harold P. Benson and Serpil Sayin. A face search heuristic algorithm for optimizing over the efficient set. *Naval Research Logistics*, 40:103–116, 1993.

[10] Harold P. Benson and Serpil Sayin. Towards finding global representations of the efficient set in multiple objective optimization. *Naval Research Logistics*, 44:47–67, 1997.

[11] V. E. Berezkin, G. K. Kamenev, and A. V. Lotov. Hybrid adaptive methods for approximating a nonconvex multidimensional Pareto frontier. *Computational Mathematics and Mathematical Physics*, 46(11):1918–1931, 2006.

[12] Rajendra Bhatia. *Positive Definite Matrices*. Princeton Series in Applied Mathematics. Princeton University Press, 2007.

[13] G. R. Bitran and T. L. Magnanti. The structure of admissible points with respect to cone dominance. *Journal of Optimization Theory and Applications*, 29(4):573–614, December 1979.

[14] H. Bonnel and J. Morgan. Semivectorial bilevel optimization problem: Penalty approach. *Journal of Optimization Theory and Applications*, 131(3):365–382, December 2006.

[15] Jürgen Branke and Kalyanmoy Deb. Integrating user preferences into evolutionary multi-objective optimization. In Yaochu Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, pages 461–478. Springer, October 2004.

[16] John Buchanan and Lorraine Gardiner. A comparison of two reference point methods in multiple objective mathematical programming. *European Journal of Operational Research*, 149:17–34, 2003.

[17] Vira Chankong and Yacov Y. Haimes. *Multiobjective Decision Making: Theory and Methodology*. North-Holland Series in System Science and Engineering. North-Holland, 1983.

[18] Yuh-Wen Chen, Chi-Hwang Wang, and Sain-Ju Lin. A multi-objective geographic information system for route selection of nuclear waste transport. *Omega: The International Journal of Management Science*, 36:363–372, 2008.

[19] S. Yu. Churkina. Search for efficient solutions of multi-criterion problems by target-level method. *Computational Mathematics and Modeling*, 13(2):208–213, 2002.

[20] Yann Collette and Patrick Siarry. Three new metrics to measure the convergence of metaheuristics towards the Pareto frontier and the aesthetic of a set of solution in biobjective optimization. *Computers & Operations Research*, 32:773–792, 2005.

[21] Piotr Czyżak and Andrzej Jaszkiewicz. Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7:34–47, 1998.

[22] Indraneel Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[23] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Technical report, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, 2000.

[24] Kalyanmoy Deb and Himanshu Gupta. Searching for robust Pareto-optimal solutions in multi-objective optimization. In *Evolutionary Multi-Criterion Optimization*, pages 150–164. Springer, 2005.

[25] Kalyanmoy Deb and Himanshu Gupta. Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14(4):463–494, 2006.

[26] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[27] Matthias Ehrgott. *Multicriteria Optimization.* Springer, second edition, 2005.

[28] Matthias Ehrgott and Margaret M. Wiecek. Multiobjective programming. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 667–722. Springer, 2005.

[29] Gabriele Eichfelder. An adaptive scalarization method in multiobjective optimization. *SIAM Journal on Optimization*, 19(4):1694–1718, 2009.

[30] Gabriele Eichfelder. A constraint method in nonlinear multi-objective optimization. In *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, pages 3–12. Springer, 2009.

[31] Alexander Engau. *Domination and Decomposition in Multiobjective Programming.* PhD thesis, Clemson University, 2007.

[32] Ali Farhang-Mehr and Shapour Azarm. An information-theoretic entropy metric for assessing multi-objective optimization solution set quality. *Journal of Mechanical Design*, 125:655–663, December 2003.

[33] Jörg Fliege. Gap-free computation of Pareto-points by quadratic scalarization. *Mathematical Methods of Operations Research*, 59:69–89, 2004.

[34] Yan Fu and Urmila M. Diwekar. An efficient sampling approach to multiobjective optimization. *Annals of Operations Research*, 132:109–134, 2004.

[35] J. Guddat, F. Guerra Vasquez, K. Tammer, and K. Wendler. *Multiobjective and Stochastic Optimization Based on Parametric Optimization.* Akademie-Verlag, 1985.

[36] Horst W. Hamacher, Christian Roed Pedersen, and Stefan Ruzika. Finding representative systems for discrete bicriterion optimization problems. *Operations Research Letters*, 35:336–344, 2007.

[37] Siegfried Helbig. On a constructive approximation of the efficient outcomes in bicriterion vector optimization. *Journal of Global Optimization*, 5:35–48, 1994.

[38] Deng Kui Huang, Huan Neng Chiu, Rue Huei Yen, and Jen Huei Chang. A fuzzy multi-criteria decision making approach for solving a bi-objective personnel assignment problem. *Computers & Industrial Engineering*, 56(1):1–10, February 2009.

[39] Brian J. Hunt. *Multiobjective Programming with Convex Cones: Methodology and Applications.* PhD thesis, Clemson University, 2004.

[40] J. Jahn and A. Merkel. Reference-point approximation method for the solution of bicriterial nonlinear optimization problems. *Journal of Optimization Theory and Applications*, 78:87–103, 1992.

[41] Johannes Jahn. Multiobjective search algorithm with subdivision technique. *Computational Optimization and Applications*, 35:161–175, 2006.

[42] Toni Jarimo and Ahti Salo. Multicriteria partner selection in virtual organizations with transportation costs and other network interdependencies. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 39(1):124–129, 2009.

[43] Esra Karasakal and Murat Köksalan. Generating a representative subset of the efficient frontier in multiple criteria decision making. *Operations Research*, 57(1):187–199, January-February 2009.

[44] Il Yong Kim and Olivier L. de Weck. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, 29:149–158, 2005.

[45] Il Yong Kim and Olivier L. de Weck. Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. *Structural and Multidisciplinary Optimization*, 31:105–116, 2006.

[46] Panis Kouvelis and Serpil Sayin. Algorithm robust for the bicriteria discrete optimization problem. *The Annals of Operations Research*, 147:71–85, 2006.

[47] Semën S. Kutateladze. Convex ϵ-programming. *Soviet Mathematics. Doklady*, 20(2):391–393, 1979.

[48] Yiu-Wing Leung and Yuping Wang. U-measure: A quality measure for multiobjective programming. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 33(3):337–343, May 2003.

[49] Alexander V. Lotov, Vladimir A. Bushenkov, and Georgy K. Kamenev. *Interactive Decision Maps: Approximation and Visualization of Pareto Frontier*. Kluwer Academic Publishers, 2004.

[50] Michael Masin and Yossi Bukchin. Diversity maximization approach for multiobjective optimization. *Operations Research*, 56(2):411–424, March-April 2008.

[51] Christopher A. Mattson, Anoop A. Mullur, and Achille Messac. Smart Pareto filter: Obtaining a minimal representation of multiobjective design space. *Engineering Optimization*, 36(6):721–740, December 2004.

[52] Hong-Yun Meng, Xiao-Hua Zhang, and San-Yang Liu. New quality measures for multiobjective programming. *Lecture Notes in Computer Science*, 3611:1044–1048, 2005.

[53] A. Messac, A. Ismail-Yahaya, and C. A. Mattson. The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, 25:86–98, 2003.

[54] Achille Messac and Christopher A. Mattson. Generating well-distributed sets of Pareto points for engineering design using physical programming. *Optimization and Engineering*, 3:431–450, 2002.

[55] Achille Messac and Christopher A. Mattson. Normal constraint method with guarantee of even representation of complete Pareto frontier. *AIAA Journal*, 42(10):2101–2111, October 2004.

[56] J. N. Morse. Reducing the size of the nondominated set: pruning by clustering. *Computers & Operations Research*, 7:55–66, 1980.

[57] Katta G. Murty, Layne T. Watson, and Leroy M. Kelly. A geometric problem in simplicial cones with applications to linear complementarity problems. Technical report, University of Michigan, 1986.

[58] Vladimir D. Noghin. Relative importance of criteria: a quantitative approach. *Journal of Multi-Criteria Decision Analysis*, 6(6):355–363, 1997.

[59] Włodzimierz Ogryczak, Adam Wierzbicki, and Marcin Milewski. A multi-criteria approach to fair and efficient bandwidth allocation. *Omega: The International Journal of Management Science*, 36:451–463, 2008.

[60] Christos H. Papadimitriou and Mihalis Yannakakis. On the approximability of trade-offs and optimal access of web sources (extended abstract). In *Proceedings of the* 41^{st} *Annual Symposium on Foundations of Computer Science*, pages 86–92. IEEE Computer Society, 2000.

[61] Adriano Pascoletti and Paolo Serafini. Scalarizing vector optimization problems. *Journal of Optimization Theory and Applications*, 42(4):499–524, April 1984.

[62] Rupesh Kumar Pati, Prem Vrat, and Pradeep Kumar. A goal programming model for paper recycling system. *Omega: The International Journal of Management Science*, 36:405–417, 2008.

[63] Sam Perlis. *Theory of Matrices*. Dover Publications, Inc., 1952.

[64] Berç Rustem. *Algorithms for Nonlinear Programming and Multiple-Objective Decisions*. John Wiley & Sons, 1998.

[65] Stefan Ruzika. *On Multiple Objective Combinatorial Optimization*. PhD thesis, University of Kaiserslautern, 2007.

[66] Stephan Ruzika and Margaret M. Wiecek. Approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3):473–501, September 2005.

[67] Serpil Sayin. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87:543–560, 2000.

[68] Serpil Sayin. A procedure to find discrete representations of the efficient set with specified coverage errors. *Operations Research*, 51(3):427–436, May-June 2003.

[69] Serpil Sayin and Panos Kouvelis. The multiobjective discrete optimization problem: A weighted min-max two-stage optimization approach and a bicriteria algorithm. *Management Science*, 51(10):1572–1581, October 2005.

[70] Bernd Schandl. *Norm-Based Evaluation and Approximation in Multicriteria Programming.* PhD thesis, Clemson University, 1999.

[71] Jason R. Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Massachusetts Institute of Technology, 1995.

[72] Lizhen Shao and Matthias Ehrgott. Finding representative nondominated points in multiobjective linear programming. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)*, pages 245–252, 2007.

[73] Kiyotaka Shimizu, Yo Ishizuka, and Jonathan F. Bard. *Nondifferentiable and Two-Level Mathematical Programming.* Kluwer Academic Publishers, 1997.

[74] R. E. Steuer and E. U. Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344, 1983.

[75] Ralph E. Steuer and Frederick W. Harris. Intra-set point generation and filtering in decision and criterion space. *Computers & Operations Research*, 7:41–53, 1980.

[76] John Sylva and Alejandro Crema. A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs. *European Journal of Operational Research*, 180:1011–1027, 2007.

[77] M. Tanaka, H. Watanabe, Y. Furnkawa, and T. Tanino. Ga-based decision support system for multicriteria optimization. In *Proceedings of the 1995 International Conference of Systems, Man, and Cybernetics*, volume 2, pages 1556–1561, 1995.

[78] Santosh Tiwari, Margaret M. Wiecek, and Georges Fadel. Inclusion of preferences in the design of evolutionary algorithms. In *Proceedings of 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization (MA&O) Conference*, September 2008.

[79] David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations.* PhD thesis, Air Force Institute of Technology, 1999.

[80] Margaret M. Wiecek. Advances in cone-based preference modeling for decision making with multiple criteria. *Decision Making in Manufacturing and Services*, 1(1-2):153–173, 2007.

[81] Jin Wu and Shapour Azarm. Metrics for quality assessment of a multiobjective design optimization solution set. *Transactions of the ASME*, 123:18–25, March 2001.

[82] Po-Lung Yu. *Multiple-Criteria Decision Making: Concepts, Techniques, and Extensions*, volume 30 of *Mathematical Concepts and Methods in Science and Engineering.* Springer, 1985.

[83] Wei Hong Zhang and Tong Gao. A min-max method with adaptive weightings for uniformly spaced Pareto optimum points. *Computers & Structures*, 84:1760–1769, 2006.

[84] Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology, 1999.

[85] Eckart Zitzler, Joshua Knowles, and Lothar Thiele. Quality assessment of Pareto set approximations. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 373–404. Springer, 2008.

[86] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *Parallel Problem Solving from Nature - PPSN V*, pages 292–301. Springer, September 1998.