Approximating the Pareto Set of Multiobjective Linear Programs via Robust Optimization

Gorissen, B.L.; den Hertog, D.

Publication date:
2012

Citation for published version (APA):
Gorissen, B. L., & den Hertog, D. (2012). *Approximating the Pareto Set of Multiobjective Linear Programs via Robust Optimization*. (CentER Discussion Paper; Vol. 2012-031). Operations research.

No. 2012-031

Approximating the Pareto set of Multiobjective Linear programs via Robust Optimization

By

Bram L. Gorissen, Dick den Hertog

April 12, 2012

ISSN 0924-7815

Approximating the Pareto set of multiobjective linear programs via Robust Optimization

Bram L. Gorissen, Dick den Hertog
Department of Econometrics and Operations Research/Center for Economic Research (CentER),
Tilburg University, PO Box 90153, 5000 LE Tilburg, The Netherlands
b.l.gorissen@uvt.nl

Abstract

The Pareto set of a multiobjective optimization problem consists of the solutions for which one or more objectives can not be improved without deteriorating one or more other objectives. We consider problems with linear objectives and linear constraints and use Adjustable Robust Optimization and Polynomial Optimization as tools to approximate the Pareto set with polynomials of arbitrarily large degree. The main difference with existing techniques is that we optimize a single (extended) optimization problem that provides a polynomial approximation whereas existing methods iteratively construct a piecewise linear approximation. The proposed method has several advantages, e.g. it is more useful for visualizing the Pareto set, it can give a local approximation of the Pareto set, and it can be used for determining the shape of the Pareto set.

keywords: Pareto set; multiobjective; polynomial inner approximation; robust optimization; polynomial optimization; SOS

JEL classification: C61

1 Introduction

Multiobjective optimization problems (MOPs) have a vector valued objective function $\boldsymbol{f} = [f_i]_{1 \leq i \leq k}$, where each f_i is a separate objective. Often it is not possible to have optimal values for all f_i simultaneously, e.g. in portfolio optimization it is not possible to have minimum risk and maximum return at the same time. Another example is intensity-modulated radiation therapy, where tumour coverage is balanced with sparing of surrounding organs [4, 15]. Optimization of a vector valued function involves a trade-off between two or more objectives f_i $(1 \leq i \leq k)$.

A simple way to deal with multiple objectives is by assigning an importance factor $w_i > 0$ to each objective and optimizing $\sum_{i=1}^{k} w_i f_i$ (we make the assumption that all f_i should be minimized w.l.o.g.). If such importance factors are not known a priori, a Pareto set (PS) allows the decision maker to make the trade-off after optimization. The set PS consists of all objective vectors $\boldsymbol{f}$ in which one or more objectives can not be improved without deteriorating one or more other objectives. Overviews of MOPs and approximation methods can be found in [3, 5, 13].

In practice often approximations of PS are used, since the exact PS can often not be found. In literature many different approximation methods are proposed. It is desirable to approximate PS with as few optimization runs as possible [15]. A well-known class of such approximation methods is sandwich methods. Sandwich methods [15] produce piecewise linear approximations in which between PS is located. In each iteration an optimization problem is solved, which leads to adding one or more facets to the approximations. All other methods (again see [3, 5, 13]) are also sequential, i.e. in each iteration one has to solve an optimization problem which leads to an improvement of the approximation.

In this paper, we focus on approximating PS for linear programming and propose a totally different way than those in the literature. The first difference is that our method is not sequential, but generates the approximation by solving one extended optimization problem. The second difference is that the final approximation is not piecewise linear but a polynomial. The way we construct this approximation is by using techniques from Adjustable Robust Optimization (ARO) [1] and Polynomial Optimization [9]. We first explain the link to ARO. The Pareto set is seen as a function of the *uncertain parameters* $f_1, ..., f_{k-1}$. The area of interest, i.e. the domain for $f_1, ..., f_{k-1}$ for which we would like to approximate the Pareto set, is considered as the *uncertainty region.* All variables in the linear program are made *adjustable* in the parameters $f_1, ..., f_{k-1}$. We use polynomials for the decision rule, and use Polynomial Optimization theory to reformulate the resulting robust counterpart into a Semi-Definite Programming (SDP) problem. Since the number of uncertain parameters (i.e. $k-1$) is often low, the sizes of the LMIs in the SDP are relatively small. Notice that in our approach ARO is merely used as a tool, uncertainty in the data is not considered.

The approach proposed in this paper has the following advantages:

The first advantage is that the final approximation is more tractable for navigating through PS. The polynomial representation is useful for the user to visualize PS for selecting the final solution. This is the reason why in [6], afer determining points in the feasible region close to or on PS, polynomial regression is used to obtain a tractable representation of PS (a so-called response surface). Our method finds such a tractable representation directly, with the additional advantage that it is guaranteed to lie in the feasible region. Sometimes the decision maker needs a local approximation of PS around a given solution. [18] formulate and test a method that gives a local quadratic approximation of (not necessarily convex) PS, but this approximation is neither an inner nor an outer approximation. Our method gives a polynomial of arbitrary degree and is guaranteed to give an inner approximation.

The second advantage is that our approach can be used to certify with a single optimization run that a given set V is dominated, i.e. that all elements of the set are dominated. If our method finds a feasible solution, then this solution is a certificate that the set V is dominated.

The third advantage is that the explicit polynomial approximation can be used in optimization problems with Pareto constraints. Such problems contain constraints that enforce that the solution should be (near) Pareto optimal for a certain multi-objective linear program. Examples of these problems can be found in e.g. [7]. A Pareto constraint can be replaced by the explicit polynomial approximation found by our method.

The fourth advantage is the possibility to quickly determine the shape of PS. [4], e.g., show that in IMRT the set of feasible objective vectors is often “long and narrow” and therefore a linear approximation of PS suffices. This linear approximation can be easily obtained by our approach. Finally, after an initial approximation, the most interesting subregion can be selected, followed by one or more iterations of approximation and selection. An interesting subregion can also be used as input for another algorithm that explores it more carefully.

This method also has five disadvantages. First, the resulting problem is often an SDP while the original problem is LP. Only in case the approximation is linear and the region of interest is polyhedral, the resulting problem is LP. Note that this is still an interesting case; see [4] that uses linear approximations in IMRT problems. Second, our method requires the region of interest to be known. Sandwich algorithms are capable of exploring the region of interest [15]. Third, it is difficult to approximate the Pareto set at its vertices, because polynomials are smooth functions. The further the vertex angle from 180°, the more difficult it is to approximate it. However, in some cases a smooth approximation is desirable, see e.g. [12]. Fourth, the method can not be extended to nonlinear multiobjective problems with current RO technology. Methods for approximating nonlinear MOPs can be found in [11, 16]. Fifth, while we show (Appendix A) that the method can also produce outer approximations, this is practically impossible due to computational issues.

2 Notation

We use the notation from [15] with some minor changes.

Throughout this paper, we use the following orderings of vectors. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ with $n \geq 2$. With x_i, we denote the i^{th} element of the vector $\boldsymbol{x}$. To enumerate different vectors, we use superscripts. When ordering two vectors, we use:

- $\boldsymbol{x} < \boldsymbol{y} \Leftrightarrow x_i < y_i$ for all $i = 1, \ldots, n$.
- $\boldsymbol{x} \lneqq \boldsymbol{y} \Leftrightarrow x_i \leq y_i$ for all $i = 1, \ldots, n$ and $\boldsymbol{x} \neq \boldsymbol{y}$.
- $\boldsymbol{x} \leq \boldsymbol{y} \Leftrightarrow x_i \leq y_i$ for all $i = 1, \ldots, n$.

The symbols $>$, $\gneqq$ and $\geq$ are defined accordingly. We furthermore define the set $\mathbb{R}^n_- = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} \leq 0\}$. If $X \subseteq \mathbb{R}^n$, then we define $X + \mathbb{R}^n_- = \{\boldsymbol{x} + \boldsymbol{y} : \boldsymbol{x} \in X, \boldsymbol{y} \in \mathbb{R}^n_-\}$. The sets $\mathbb{R}^n_+$ and $X + \mathbb{R}^n_+$ are defined accordingly.

In this paper, we consider the following multi-objective optimization problem:

$$\begin{array}{ll} \min & \boldsymbol{f}(\boldsymbol{x}) = [(\boldsymbol{c^1})^\top \boldsymbol{x}, \ldots, (\boldsymbol{c^k})^\top \boldsymbol{x}]^\top \\ & A\boldsymbol{x} \leq \boldsymbol{b}, \end{array}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the optimization variable, $\boldsymbol{c^i} \in \mathbb{R}^n$ are the objective vectors, and $A \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$.

As it is generally impossible to find a feasible x that minimizes all objectives at the same time, our aim is to find a set of so-called Pareto optimal solutions.

Definition 1
An objective vector $\boldsymbol{f}(\boldsymbol{x})$, for $\boldsymbol{x}$ such that $A\boldsymbol{x} \leq \boldsymbol{b}$, is (strongly) dominated if there exists an $\widetilde{\boldsymbol{x}}$ such that $A\widetilde{\boldsymbol{x}} \leq \boldsymbol{b}$ and $\boldsymbol{f}(\widetilde{\boldsymbol{x}}) < \boldsymbol{f}(\boldsymbol{x})$. If no such $\widetilde{\boldsymbol{x}}$ exists, the objective vector $\boldsymbol{f}(\boldsymbol{x})$ is weakly Pareto optimal.

An objective vector $\boldsymbol{f}(\boldsymbol{x})$*, for* $\boldsymbol{x}$ *such that* $A\boldsymbol{x} \leq \boldsymbol{b}$*, is weakly dominated if there exists an* $\widetilde{\boldsymbol{x}}$ *such that* $A\widetilde{\boldsymbol{x}} \leq \boldsymbol{b}$ *and* $\boldsymbol{f}(\widetilde{\boldsymbol{x}}) \lneqq \boldsymbol{f}(\boldsymbol{x})$*. If no such* $\widetilde{\boldsymbol{x}}$ *exists, the objective vector* $\boldsymbol{f}(\boldsymbol{x})$ *is (strongly) Pareto optimal.*

The set of Pareto optimal solutions is denoted by PS. An inner and outer approximation of the Pareto set are defined as follows:

Definition 2 *A set* $IPS \subseteq \mathbb{R}^k$ *is an inner approximation of* PS *if it satisfies* $IPS \subseteq PS + \mathbb{R}^k_+$*.*

Definition 3 *A set* $OPS \subseteq \mathbb{R}^k$ *is an outer approximation of* PS *if it satisfies* $PS \subseteq OPS + \mathbb{R}^k_+$*.*

We will approximate PS with polynomials on multidimensional sets. The following definitions are used to define the degree of a polynomial.

Definition 4 *A monomial of degree* d *in* $\boldsymbol{x} \in \mathbb{R}^n$ *with powers* $\boldsymbol{a} \in \mathbb{R}^n$ *is defined by* $\prod_{i=1}^n x_i^{a_i}$*.*

Definition 5 *A polynomial of degree* d *in* $\boldsymbol{x} \in \mathbb{R}^n$ *is defined as the sum of monomials in* $\boldsymbol{x}$ *of degree up to* d*. The degree of a polynomial* f *is denoted as* $deg(f)$*.*

3 Inner approximation

Let $U \subseteq \mathbb{R}^{k-1}$ be the domain of interest for $((\boldsymbol{c^1})^\top \boldsymbol{x}, (\boldsymbol{c^2})^\top \boldsymbol{x}, \ldots, (\boldsymbol{c^{k-1}})^\top \boldsymbol{x})$. For a fixed $\boldsymbol{u}$ in U, the following optimization problem determines a single weakly Pareto optimal solution [13, Thm. 3.2.1]:

$$\begin{aligned} \min_{x} \quad & (\boldsymbol{c^k})^\top \boldsymbol{x} & \\ & (\boldsymbol{c^i})^\top \boldsymbol{x} \leq u_i & i = 1, 2, \ldots, k-1 \\ & A\boldsymbol{x} \leq \boldsymbol{b}. & \end{aligned}$$

If the solution x is unique, it is (strongly) Pareto optimal [13, Thm. 3.2.4]. For every $\boldsymbol{u}$, there will be a different optimal $\boldsymbol{x}$. So, we want to solve for $\boldsymbol{x}$ as a function of $\boldsymbol{u}$. The constraints should hold for all $\boldsymbol{x}(\boldsymbol{u})$ for which $\boldsymbol{u}$ is in U, and the goal is e.g. to minimize the average objective:

$$\min_{\boldsymbol{x}(\boldsymbol{u})} \quad \int_U (\boldsymbol{c^k})^\top \boldsymbol{x}(\boldsymbol{u}) du \tag{1a}$$

$$(\boldsymbol{c^i})^\top \boldsymbol{x}(\boldsymbol{u}) \leq u_i \qquad \forall \boldsymbol{u} \in U, \quad i = 1, 2, \ldots, k-1 \tag{1b}$$

$$A\boldsymbol{x}(\boldsymbol{u}) \leq \boldsymbol{b} \qquad \forall \boldsymbol{u} \in U. \tag{1c}$$

This is an ARO problem, where $\boldsymbol{u}$ is the uncertain parameter, U is the uncertainty region, and x is an adjustable variable [1]. It is difficult to optimize over functions, therefore

ARO uses parameterized functions for adjustable variables. The adjustable variables then become expressions that are linear in the parameters. For instance, if we take a linear parameterization $\boldsymbol{x(u)} = \boldsymbol{\alpha^0} + \alpha^1 \boldsymbol{u}$, the parameters are $\boldsymbol{\alpha^0} \in \mathbb{R}^n$ and $\alpha^1 \in \mathbb{R}^{n\times(k-1)}$. After substituting $\boldsymbol{x(u)}$ in the problem (1a)-(1c), an ARO problem with constraints that are linear in $\boldsymbol{\alpha^0}$ and α^1 remains. In general, the tractability of (1a)-(1c) depends on the class of functions considered for x and the set U. Given a solution to this optimization problem, an inner approximation is given by
$\{((\boldsymbol{c^1})^\top \boldsymbol{x(u)}, (\boldsymbol{c^2})^\top \boldsymbol{x(u)}, ..., (\boldsymbol{c^k})^\top \boldsymbol{x(u)}) : \boldsymbol{u} \in U\}$. Constraint (1c) ensures that $\boldsymbol{x(u)}$ is feasible. So, the resulting inner approximation indeed lies in $PS + \mathbb{R}^k_+$. The objective (1a) minimizes the volume under the inner approximation if (1b) is tight for all $\boldsymbol{u}$. Note that a constant function $\boldsymbol{x(u)}$ may be feasible for this optimization problem. The reason why this problem returns an interesting inner approximation and not just a constant function x is because from the objective it follows that a smaller $(\boldsymbol{c^k})^\top \boldsymbol{x(u)}$ is better, and a smaller $(\boldsymbol{c^k})^\top \boldsymbol{x(u)}$ can only be obtained by increasing $(\boldsymbol{c^i})^\top \boldsymbol{x(u)}$ for $i = 1, 2, \ldots, k-1$. This increase as a function of $\boldsymbol{u}$ is constrained by (1b).

A functional description of the inner approximation may be more tractable for a specific purpose. Therefore, we define:

$$IPS = \{(\boldsymbol{u}, (\boldsymbol{c^k})^\top \boldsymbol{x(u)}) : \boldsymbol{u} \in U\}.$$

If constraint (1b) is tight for all $\boldsymbol{u}$, IPS is the same as the inner approximation given before. Otherwise, the given inner approximation dominates IPS.

The question arises for which regions U and functions x this formulation is tractable. When U is polyhedral, for instance a box ($\{\boldsymbol{u} \in \mathbb{R}^{k-1} : ||\boldsymbol{u}||_\infty \leq 1\}$), and x is linear in $\boldsymbol{u}$, this problem can be formulated as an LP. The optimization problem minimizes the volume enclosed between the linear approximation, the Pareto curve and the boundary of U. In case of two objectives ($k = 2$), it will find the line connecting the point on the Pareto curve where $(\boldsymbol{c^1})^\top \boldsymbol{x} = -1$ with the point on the Pareto curve where $(\boldsymbol{c^1})^\top \boldsymbol{x} = 1$. Hence, it finds two Pareto optimal points. This extends to larger k, where it finds a plane going through $k-1$ Pareto optimal points. In case U is a box, the instance size also grows linearly in k. This result of linear growth is new, because determining a linear inner approximation over a box would require determining the Pareto optimal points at the exponentially growing number of extreme points of U.

The inner approximation becomes more interesting when x is nonlinear in $\boldsymbol{u}$. When $k = 2$, a tractable choice is given by polynomials: for polynomials of arbitrarily large degree, the problem can be formulated as an SDP [1, Lemma 14.3.4], for which polynomial time solvers are available. When $k > 2$, the problem is tractable when U is ellipsoidal and x is quadratic in $\boldsymbol{u}$ [1, Lemma 14.3.7]. The resulting problem is an SDP with $m + k - 1$ variables matrices of size $k + 1$ and $m + k - 1$ constraints.

For $k > 2$, the problem can also be reformulated as an SDP when U is a semialgebraic set ($U = \{\boldsymbol{u} : p_i(\boldsymbol{u}) \leq 0 \quad (i \in I)\}$, where p_i are polynomials of arbitrary degree), and $\boldsymbol{x}$ is a polynomial in $\boldsymbol{u}$, but the reformulation is not always equivalent. This means that the resulting optimal solution for the SDP reformulation may not be optimal for (1a)-(1c). However, the solution to the reformulation is always an inner approximation, and the numerical results in Section 4.2 are promising. The reformulation is based on polynomials that are sums of squares (SOS) of other polynomials. An example of an SOS polynomial is $5x^2 + 2x + 65$, because it can be decomposed as $(2x + 4)^2 + (x - 7)^2$. Testing whether a polynomial is SOS is equivalent to solving an SDP [9, Lemma 3.8] with a matrix of size

Table 1: An overview of the tractability of inner approximation (1a)-(1c).

k	U	$\boldsymbol{x(u)}$	Tractability	Exact
≥ 2	box	linear	LP	$\surd$
≥ 2	polyhedral	linear	LP	$\surd$
≥ 2	ball	linear	CQP	$\surd$
2	interval	polynomial	SDP	$\surd$
> 2	ellipsoidal	quadratic	SDP	$\surd$
≥ 2	semialgebraic	polynomial	SDP	$-$

$\binom{k-1+d}{d}$, where $k-1$ is the number of variables and d is the degree of the polynomial. An SOS polynomial is obviously nonnegative. Let us focus on constraint (1b):

$$u_1 - (\boldsymbol{c^1})^\top \boldsymbol{x(u)} \geq 0 \qquad \forall \boldsymbol{u} : p_i(\boldsymbol{u}) \leq 0 \quad (i \in I).$$

By applying Putinar's Positivstellensatz [9, Thm. 3.20], which has been done before in ARO [2], we can obtain a sufficient condition under which this constraint holds:

$$u_1 - (\boldsymbol{c^1})^\top \boldsymbol{x(u)} = \sigma_0(\boldsymbol{u}) + \sum_{i \in I} p_i(\boldsymbol{u})\sigma_i(\boldsymbol{u}), \tag{2}$$

where σ_0 and σ_i $(i \in I)$ are SOS. Solving a problem with constraint (2) instead of (1b) is conservative for two reasons. First, (2) may not be a necessary condition when there are no σ_0, σ_i that are SOS for which the set $\{\boldsymbol{u} : \sigma_0(\boldsymbol{u}) + \sum_{i \in I} p_i(\boldsymbol{u})\sigma_i(\boldsymbol{u}) \geq 0\}$ is compact, or when (1b) is a not a strict inequality. Compactness can easily be guaranteed by including a constraint $\sum_{i=1}^{k-1} u_i^2 - R \leq 0$ to the description of U, which can be done without changing U because U is bounded [9, p. 186]. However, (1b) will in general not be a strict inequality. Second, solving a problem with constraint (2) as an SDP requires bounding the degree of σ_0 and σ_i.

We let the degree of x determine the complexity of the problem unless g is of higher degree, so we take the degree of σ_0 equal to $\max\{deg(x), \max_i\{deg(g_i)\}\}$, and the degree of σ_i equal to $\max\{0, deg(\sigma_0) - deg(g_i)\}$.

An overview of all tractable cases is given in Table 1.

For many uncertainty regions it may be difficult to reformulate the integral in the objective function as a simple linear function in the optimization variables. In that case the objective can be replaced with the average value of $(\boldsymbol{c^k})^\top \boldsymbol{x(u)}$ at well distributed sampling points $\boldsymbol{u}$ in U. For efficient sampling from a polytope, see e.g. [8].

The user has to specify the domain of interest U. If the specified region is too large, two things may occur. First, U may contain a vector with objective values that are too optimistic in the sense than they can not be met, in which case constraint (1b) is infeasible. Second, U may contain objective values that are not weakly dominated by $\boldsymbol{f(x)}$ for any feasible $\boldsymbol{x}$. In that case constraint (1b) will not be tight and also the objective is not fully related to the area of interest.

When the number of objectives is three or more, a weak parameterization of x can be another cause of infeasibility. If feasible solutions exist for all $\boldsymbol{u}$ in U, it is possible that these solutions can not be attained with the parameterization. An easy example is the case where U contains two different Pareto optimal solutions (projected on the first $k-1$ coordinates) while x is a constant function. When the optimization problem is infeasible, we do not see a possibility to detect whether the parameterization is too weak or U contains infeasible points.

It is known that PS is convex, and that it is nonincreasing. It may be the case that these properties do not hold for the inner approximation, which is problematic when the inner approximation is used in an algorithm that assumes these properties to hold. In case x is linear in $\boldsymbol{u}$, these conditions are automatically satisfied. In case $k = 2$, u is one-dimensional and convexity and nonincreasingness of $(\boldsymbol{c^2})^\top \boldsymbol{x(u)}$ can be enforced by constraining the first and second derivative w.r.t. u. The first and second derivative of a polynomial is again a polynomial, so constraining these to be negative and positive, respectively, for polynomial x does not increase the complexity class of the problem. In case $k > 2$, U is a polynomial and x is quadratic in $\boldsymbol{u}$, say $x(\boldsymbol{u})_i = \alpha_i^0 + (\boldsymbol{\alpha_i^1})^\top \boldsymbol{u} + \boldsymbol{u}^\top \Gamma_i \boldsymbol{u}$ where $\alpha_i^0 \in \mathbb{R}, \boldsymbol{\alpha_i^1} \in \mathbb{R}^{k-1}$ and $\Gamma_i \in \mathbb{R}^{(k-1)\times(k-1)}$ are decision variables. Nonincreasingness can easily be enforced by adding the constraints $\boldsymbol{\alpha_i^1} + 2\Gamma_i \boldsymbol{u} \leq 0$ for all $\boldsymbol{u}$ in U, which is a set of $n(k-1)$ linear constraints with ellipsoidal uncertainty, each of which can be reformulated as a conic quadratic constraint. For convexity it is required that Γ_i is positive semidefinite, which is an SDP constraint.

4 Numerical examples

4.1 Two objectives

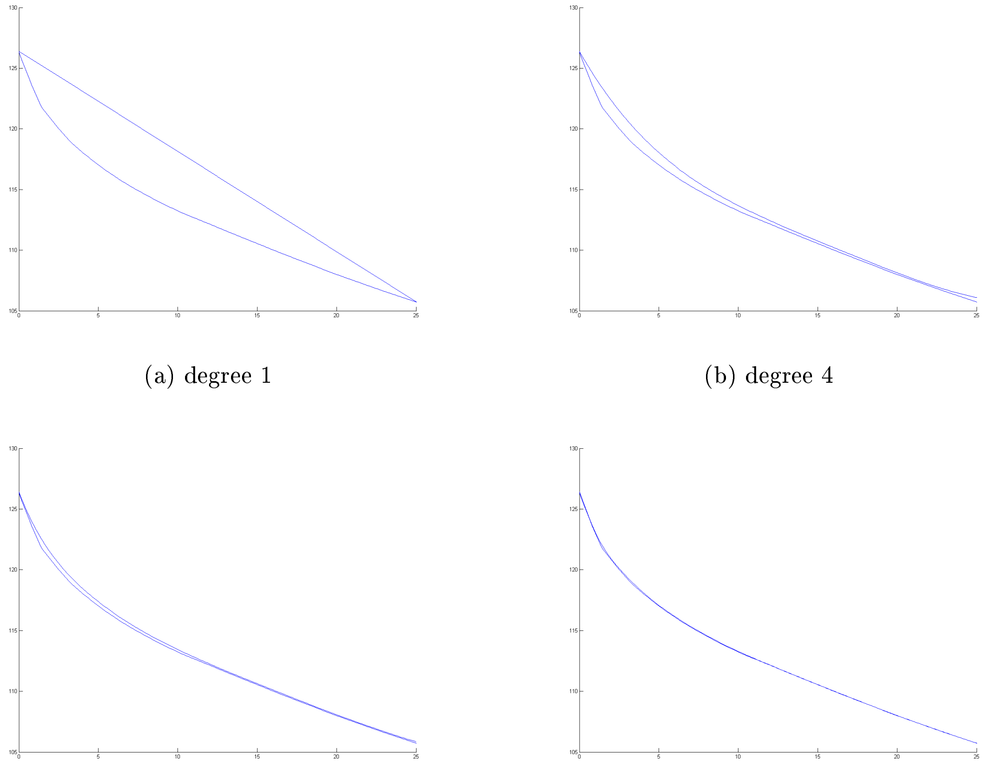
We construct a semi-random 150×170 matrix A, a 150-vector b, and two 170-vectors $\boldsymbol{c^1}$ and $\boldsymbol{c^2}$, such that the Pareto cuve is interesting on the interval $[0, 25]$. We compute a polynomial inner approximations of degree up to 16.

We solve linear programs with Matlab linprog. We enter linear constraints with LMI uncertainty into YALMIP [10]. YALMIP reformulates this problem as an SDP. In Appendix B we show how to do this reformulation by hand in case of a polynomial of degree 3. We let YALMIP export the resulting problem, then we reformulate free variables as the difference of two nonnegative variables using CSDP's convertf, and solve the problems with SDPA [17] (SDPA-DD [14] for the problem with a polynomial of degree 16). Figure 1 shows the resulting solutions. The solution time ranges from 21 seconds for the polynomial approximation of degree 4 to 4 minutes for degree 8 (with SDPA), and 45 minutes for degree 16 (with SDPA-DD).

4.2 Three objectives

We semi-randomly construct vectors $\boldsymbol{c^1} \in [0,1]^{10}$, $\boldsymbol{c^2} \in [0,1]^{10}$ and $\boldsymbol{c^3} \in [-1,0]^{10}$, and take $\mathbb{R}^{10}_+$ as the feasible region. Recall from Table 1 that we have an exact result for a quadratic inner approximation over an ellipsoidal set, and a conservative result for polynomial inner approximations of arbitrary degree over semialgebraic sets. Again we use YALMIP and SDPA to formulate the problem and solve the resulting SDPs. We use YALMIPs SOS module for constraining an expression to be SOS.

We take $\{\boldsymbol{u} : ||\boldsymbol{u} - \boldsymbol{5}||_2 \leq 5\}$ as the area of interest for $(\boldsymbol{c^1})^\top \boldsymbol{x}$ and $(\boldsymbol{c^2})^\top \boldsymbol{x}$, and approximate the Pareto set with a polynomial of degree 2 and with a polynomial of degree 4. For degree



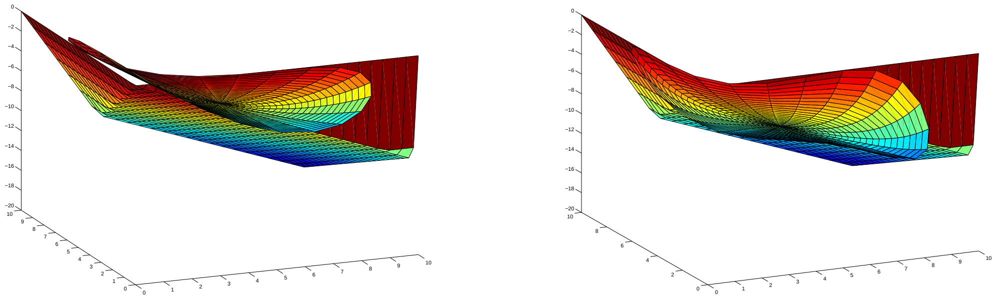
(a) degree 1

(b) degree 4

(c) degree 8

(d) degree 16

Figure 1: Numerical example with two objectives indicating the quality of the inner approximation with a polynomial decision rule. The lowest curve represents the set PS which is usually not known.



(a) degree 2

(b) degree 4

Figure 2: Numerical example with three objectives indicating the quality of the inner approximation with a polynomial decision rule. The lowest set represents the set PS which is usually not known.

2 we solve the exact robust counterpart, while for degree 4 we solve the SOS approximation. We also solve the SOS approximation for degree 2, and notice that the inner approximation is the same as with the exact robust counterpart. Figure 2 shows that the polynomial of degree 4 gives an approximation that is closer to PS than the polynomial of degree 2. The solution time is around 1.6 seconds for all three approximations.

A Outer approximation

In this appendix we show how ARO can be used to construct an outer approximation. While theoretically possible, it is practically not tractable to determine polynomial approximations of degree 2 or higher due to computational issues. For linear approximations, the results are trivial. We still mention the result because it also uses ARO to approximate PS.

Let $U = [a^1, b^1] \times [a^2, b^2] \times \cdots \times [a^{k-1}, b^{k-1}]$ with $[a^i, b^i]$ be the domain of interest for $(c^i)^\top \boldsymbol{x}$ $(i = 1, 2, \ldots, k-1)$. We construct the set OPS by creating a function function $\ell : \mathbb{R}^{k-1} \to \mathbb{R}$ for which $(u^1, u^2, \ldots, u^{k-1}, \ell(u^1, u^2, \ldots, u^{k-1}))$ is in $PS + \mathbb{R}^k_-$, and optimizing over this function:

$$\max_{\ell} \quad \int_U \ell(u_1, u_2, \ldots, u_{k-1}) du \tag{3a}$$

$$\ell((\boldsymbol{c^1})^\top \boldsymbol{x}, (\boldsymbol{c^2})^\top \boldsymbol{x}, \ldots, (c^{k-1})^\top \boldsymbol{x}) \leq (\boldsymbol{c^k})^\top \boldsymbol{x} \quad \forall \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, a^i \leq (\boldsymbol{c^i})^\top \boldsymbol{x} \leq b^i. \tag{3b}$$

Given a solution to this optimization problem, the outer approximation is $\{(u_1, u_2, \ldots, u_{k-1}, \ell(u_1, u_2, \ldots, u_{k-1})) : \boldsymbol{u} \in U\}$. The objective (3a) maximizes the volume under this approximation. Constraint (3b) ensures that the outer approximation as a function of $(\boldsymbol{c^1})^\top \boldsymbol{x}$ lies under $(\boldsymbol{c^2})^\top \boldsymbol{x}$ for every $\boldsymbol{x}$ in the domain of interest.

An optimal outer approximation is tangent to the Pareto curve at (at least) one point. This becomes clear from (3b): this constraint holds with equality for at least one x because otherwise we can add a constant to ℓ without losing feasibility, which contradicts optimality. Previous results force the decision maker to specify either this point of tangency or the derivative at this point a priori. Our formulation determines the point of tangency in such a way that the volume enclosed between this linear outer approximation and the Pareto curve over the set U, but in the linear case this turns out to give a trivial result.

When ℓ is linear, the problem (3b) can be reformulated as an LP using ARO. For the case $k = 2$ (two objectives) it can be shown that the optimal linear ℓ is a line tangent to PS at $\frac{a^1+b^1}{2}$, i.e. halfway the interval of interest. We conjecture that in higher dimensions the point of tangency is the barycenter of U. This would imply that the formulation for the outer approximation is not interesting because it is already known how to obtain an outer approximation that is tangent at a given point.

For nonlinear ℓ the SOS framework used in Section 3 can be used to reformulate the problem as an SDP when ℓ is a polynomial of arbitrary degree d. This is a polynomial in the vector $\boldsymbol{x}$, so the number of terms is $\binom{n+d}{d}$, which is also the order of the matrix in the SDP. Even for a quadratic function, the size of the SDP is often too large to solve.

The user has to specify the domain of interest $[u^1, u^2]$. Specifying the wrong domain does not lead to infeasibility. However, if the interval is too large, i.e. $(\boldsymbol{c^1})^\top \boldsymbol{x}$ does not range through the full interval, part of the outer approximation is meaningless because PS is inexistent for some $\boldsymbol{u}$.

B Derivation of the SDP formulation for a polynomial inner approximation with two objectives

We give a derivation of the SDP formulation of (1a)-(1c) in case $\boldsymbol{x(u)} = \boldsymbol{\alpha_0} + \boldsymbol{\alpha_1} u + \boldsymbol{\alpha_2} u^2 + \boldsymbol{\alpha_3} u^3$ (where $\boldsymbol{\alpha_i}$ in $\mathbb{R}^{170}$, i=1,2,3) for the numerical example of Section 4.1. Suppose the area of interest for $(\boldsymbol{c^1})^\top \boldsymbol{x}$ is $[0, 25]$, then u runs from 0 to 25. Because in the result by [1, Lemma 14.3.4] u runs from -1 to 1, we use the following linear transformation $\boldsymbol{x} \to D\boldsymbol{x} + \boldsymbol{d}$ to transform $\mathcal{Z} = \{(u, u^2, u^3) : -1 \leq u \leq 1\}$ into $\{(u, u^2, u^3) : 0 \leq u \leq 25\}$:

$$D = \begin{pmatrix} 12.5 & 0 & 0 \\ 312.5 & 156.25 & 0 \\ 5859.375 & 5859.375 & 1953.125 \end{pmatrix} \qquad \boldsymbol{d} = \begin{pmatrix} 12.5 \\ 156.25 \\ 1953.125 \end{pmatrix}$$

The problem can now be written as follows:

$$\begin{aligned} \min \quad & (\boldsymbol{c^2})^\top (25\boldsymbol{\alpha_0} + \frac{25^2}{2}\boldsymbol{\alpha_1} + \frac{25^3}{3}\boldsymbol{\alpha_2} + \frac{25^4}{4}\boldsymbol{\alpha_3}) && \text{(4a)} \\ & (\boldsymbol{c^1})^\top (\boldsymbol{\alpha_0} + [\boldsymbol{\alpha_1} \quad \boldsymbol{\alpha_2} \quad \boldsymbol{\alpha_3}](D\boldsymbol{\zeta} + \boldsymbol{d})) \leq (D\boldsymbol{\zeta} + \boldsymbol{d})_1 \qquad \forall \boldsymbol{\zeta} \in \mathcal{Z} && \text{(4b)} \\ & A(\boldsymbol{\alpha_0} + [\boldsymbol{\alpha_1} \quad \boldsymbol{\alpha_2} \quad \boldsymbol{\alpha_3}](D\boldsymbol{\zeta} + \boldsymbol{d})) \leq \boldsymbol{b} \qquad \forall \boldsymbol{\zeta} \in \mathcal{Z}, && \text{(4c)} \end{aligned}$$

where:

$$\mathcal{Z} = \{\boldsymbol{\zeta} \in \mathbb{R}^3 : \begin{pmatrix} 1 \\ \boldsymbol{\zeta} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 3 & 0 & 4 & 0 \\ 0 & 4 & 0 & 8 \\ 3 & 0 & 4 & 0 \\ 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}^\top \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{pmatrix}, \quad \begin{pmatrix} \lambda_0 & \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \\ \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 \\ \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 \end{pmatrix} \succeq 0\}.$$

Constraints (4b) and (4c) are a total of 151 semi-infinite constraints that have to hold for an infinite number of $\boldsymbol{\zeta}$. Let $\boldsymbol{A_j}$ denote the j^{th} row of A. In order to allow for shorter notation, we define the linear function $\ell : \mathbb{R}^{170}\times\mathbb{R}^{170}\times\mathbb{R}^{170} \to \mathbb{R}^3$ vector $\boldsymbol{\ell(\alpha)} := A_j[\boldsymbol{\alpha_1} \quad \boldsymbol{\alpha_2} \quad \boldsymbol{\alpha_3}]D$. The j^{th} constraint of (4c) can be rearranged to:

$$\boldsymbol{A_j\alpha_0} - \boldsymbol{b} + [\boldsymbol{\alpha_1} \quad \boldsymbol{\alpha_2} \quad \boldsymbol{\alpha_3}]\boldsymbol{d} + \boldsymbol{\ell(\alpha)\zeta} \leq 0 \qquad\qquad \forall\boldsymbol{\zeta} \in \mathcal{Z},$$

which is equivalent to:

$$\boldsymbol{A_j\alpha_0} - \boldsymbol{b} + [\boldsymbol{\alpha_1} \quad \boldsymbol{\alpha_2} \quad \boldsymbol{\alpha_3}]\boldsymbol{d} + \max_{\boldsymbol{\zeta}\in\mathcal{Z}} \{\boldsymbol{\ell(\alpha)\zeta}\} \leq 0. \tag{5}$$

The maximization problem is an SDP. Replacing this problem with its SDP dual and omitting the min operator is a well-known method to transform a semi-infinite constraint into a single constraint. We show how to do this. In the following SDP, we take the 4×4 matrix with λ_i's in the description of $\mathcal{Z}$ as our variable X. Let $\ell_i(\alpha)$ denote the i^{th} component of $\boldsymbol{\ell(\alpha)}$.

The optimization problem in (5) is:

$$\begin{aligned} \max \quad & \langle C, X\rangle \\ \text{s.t.} \quad & \langle A_i, X\rangle = b_i \quad (i = 1, 2, 3, 4) \\ & X \succeq 0, \end{aligned}$$

where $\langle\cdot,\cdot\rangle$ denotes the trace inner product, $b_1 = 1$, $b_2 = b_3 = b_4 = 0$, and:

$$C = \begin{pmatrix} \ell_3(\alpha) & \ell_2(\alpha) & \frac{4}{3}\ell_1(\alpha) + \ell_3(\alpha) & \ell_2(\alpha) \\ \ell_2(\alpha) & \frac{4}{3}\ell_1(\alpha) + \ell_3(\alpha) & \ell_2(\alpha) & \frac{4}{3}\ell_1(\alpha) + \ell_3(\alpha) \\ \frac{4}{3}\ell_1(\alpha) + \ell_3(\alpha) & \ell_2(\alpha) & \frac{4}{3}\ell_1(\alpha) + \ell_3(\alpha) & \ell_2(\alpha) \\ \ell_2(\alpha) & \frac{4}{3}\ell_1(\alpha) + \ell_3(\alpha) & \ell_2(\alpha) & \ell_3(\alpha) \end{pmatrix},$$

$$A_1 = \begin{pmatrix} 0 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}, \qquad A_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 0 & 0 \end{pmatrix}.$$

By formulating the dual and putting this into constraint (5), we get the following robust counterpart:

$$\begin{aligned} \boldsymbol{A_j\alpha_0} - \boldsymbol{b} + [\boldsymbol{\alpha_1} \quad \boldsymbol{\alpha_2} \quad \boldsymbol{\alpha_3}]\boldsymbol{d} + y_1 &\leq 0 \\ \sum_{i=1}^{4} y_i A_i - C &\succeq 0. \end{aligned}$$

Constraint (4b) and the other constraints (4c) can be transformed in a smilar way.

Acknowledgments

We would like to thank R. Sotirov from Tilburg University (The Netherlands) for her input on solving SDPs.

References

[1] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2009.

[2] D. Bertsimas, D. A. Iancu, and P. A. Parrilo. A hierarchy of near-optimal policies for multi-stage adaptive optimization. *IEEE Transactions on Automatic Control*, 56(12):2809–2824, 2011.

[3] J. Branke, K. Deb, K. Miettinen, and R. Slowinski. *Multiobjective Optimization: Interactive and evolutionary approaches*, volume 5252 of *Lecture Notes in Computer Science*. Springer, 2008.

[4] D. Craft and T. Bortfeld. How many plans are needed in an imrt multi-objective plan database? *Physics in Medicine and Biology*, 53(11):2785–2796, 2008.

[5] M. Ehrgott. *Multicriteria Optimization*. Lecture Notes in Economics and Mathematical Systems. Springer, 2005.

[6] T. Goel, R. Vaidyanathan, R. Haftka, W. Shyy, N. Queipo, and K. Tucker. Response surface approximation of Pareto optimal front in multi-objective optimization. *Computer Methods in Applied Mechanics and Engineering*, 196(4-6):879–893, 2007.

[7] S. T. Hackman and U. Passy. Maximizing a linear fractional function on a pareto efficient frontier. *Journal of Optimization Theory and Applications*, 113:83–103, 2002.

[8] R. Kannan and H. Narayanan. Random walks on polytopes and an affine interior point method for linear programming. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 561–570, 2009.

[9] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In Mihai Putinar and Seth Sullivant, editors, *Emerging Applications of Algebraic Geometry*, volume 149 of *The IMA Volumes in Mathematics and its Applications*. Springer New York, 2009.

[10] J. Löfberg. Automatic robust convex programming. *Optimization Methods and Software*, 2010. Accepted for publication.

[11] M. Luque, K. Miettinen, A. B. Ruiz, and F. Ruiz. A two-slope achievement scalarizing function for interactive multiobjective optimization. *Computers & Operations Research*, 39(7):1673–1681, 2012.

[12] J. C. C. B. S. De Mello, M. P. E. Lins, and E. G. Gomes. Construction of a smoothed dea frontier. *Pesquisa Operacional*, 22(2):183–201, 2002.

[13] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Dordrecht, 1999.

[14] M. Nakata. A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: Sdpa-gmp, -qd and -dd. In *2010 IEEE Multi-Conference on Systems and Control*, volume 25/I, pages 29–34, 2010.

[15] G. Rennen, E. R. van Dam, and D. den Hertog. Enhancement of sandwich algorithms for approximating higher-dimensional convex pareto sets. *INFORMS Journal on Computing*, 23(4):493–517, 2011.

[16] S. V. Utyuzhnikov, P. Fantini, and M. D. Guenov. A method for generating a well-distributed pareto set in nonlinear multiobjective optimization. *Journal of Computational and Applied Mathematics*, 223(2):820–841, 2009.

[17] M. Yamashita, K. Fujisawa, K. Nakata, M. Nakata, M. Fukuda, K. Kobayashi, and K. Goto. A high-performance software package for semidefinite programs: Sdpa7. Technical report, Dept. of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan, 2010.

[18] J. Zhang, M. M. Wiecek, and W. Chen. Local approximation of the efficient frontier in robust design. *Journal of Mechanical Design*, 122(2):232–236, 2000.