Succinct Approximate Convex Pareto Curves
(Extended Abstract)

Ilias Diakonikolas* Mihalis Yannakakis*

Abstract

We study the succinct approximation of convex Pareto curves of multiobjective optimization problems. We propose the concept of ϵ-convex Pareto (ϵ-CP) set as the appropriate one for the convex setting, and observe that it can offer arbitrarily more compact representations than ϵ-Pareto sets in this context. We characterize when an ϵ-CP can be constructed in polynomial time in terms of an efficient routine Comb for optimizing (exactly or approximately) monotone linear combinations of the objectives. We investigate the problem of computing minimum size ϵ-convex Pareto sets, both for discrete (combinatorial) and continuous (convex) problems, and present general algorithms using a Comb routine. For bi-objective problems, we show that if we have an exact Comb optimization routine, then we can compute the minimum ϵ-CP for continuous problems (this applies for example to bi-objective Linear Programming and Markov Decision Processes), and factor 2 approximation to the minimum ϵ-CP for discrete problems (this applies for example to bi-objective versions of polynomial-time solvable combinatorial problems such as Shortest Paths, Spanning Tree, etc.). If we have an approximate Comb routine, then we can compute factor 3 and 6 approximations respectively to the minimum ϵ-CP for continuous and discrete bi-objective problems. We consider also the case of three and more objectives and present some upper and lower bounds.

1 Introduction

Decision making involves the evaluation of different alternative solutions from a design space, and the selection of a solution that is "best" according to the criteria of interest. In most situations there are usually more than one criteria that matter. For example, in network design we are concerned with its cost, capacity, reliability; in investments we care about return and risk; in radiation therapy we care about the effects on the tumor on the one hand, and healthy organs on the other; and so forth. Such *multicriteria* (or *multiobjective*) problems are pervasive across many diverse disciplines, in economics and management, engineering, manufacturing, healthcare, etc. The area of multiobjective optimization has been (and continues to be) extensively investigated with many papers, conferences and books (see e.g. [Cli, Ehr, EG, FGE, Miet]).

In multiobjective problems, there is typically no solution that is uniformly best in all the objectives; rather, there is a trade-off between the different objectives, which is captured by the *trade-off* or *Pareto* curve (surface), the set of all feasible solutions whose vector of values for the objective functions is not dominated by any other solution. Usually the Pareto curve (set) has exponential size for discrete optimization problems (even for two objectives), and is infinite for continuous problems (and typically there is no closed form expression for it). Thus, we cannot compute the full Pareto curve and have to contend with approximation. We want to compute efficiently a small set of solutions (as small as possible) that provides a "good enough" representation (as good as possible) of the whole design space, i.e. the full Pareto curve. In fact, even in cases where the Pareto set has polynomial size, we may still want a very small number of solutions that provide the best approximation. Typically the representative set of solutions are investigated more thoroughly by the decision makers to assess the different choices and pick a suitable one based on factors that are perhaps not even formalized or quantifiable. For example, in radiotherapy planning, different plans will be assessed by the physician to select one that provides the best balance [CHSB]. Obviously, there is a small limit on the number of plans that can be examined. As another example, when we plan a trip we want to examine just a few possible routes (in terms of time, distance, cost), not a polynomial number in the size of the map. Thus, ideally we want to compute the smallest set that achieves a desired approximation. Indeed, this is the underlying goal of much of the research in the multiobjective optimization area, where

*Department of Computer Science, Columbia University, New York, NY 10027, USA. Email: {ilias,mihalis}@cs.columbia.edu. Research supported in part by NSF grants CCF-04-30946, CCF-07-28736, and by an Alexander S. Onassis Foundation Fellowship to the first author.

many heuristics have been proposed for this purpose, usually however without any performance guarantees or complexity analysis, as we do in TCS.

In recent years, we have initiated a more systematic investigation [PY1, VY, DY] to develop a theory of *multiobjective approximation* along similar rigorous lines as the approximation of single objective problems. The approximation to the Pareto curve is captured by the concept of an ϵ-Pareto set, a set P_ϵ of solutions that approximately dominates every other solution, i.e., for every other solution s, the set P_ϵ contains a solution s' that is within a factor $1+\epsilon$ or better in all the objectives. This concept was studied earlier for certain problems, e.g. multiobjective shortest paths, in [Han, Wa]. It is investigated more systematically in a general framework with respect to the size and polynomial time computability in [PY1]. The problem of computing a minimum ϵ-Pareto set was raised and studied in [VY, DY].

In several problems, the set of solution values in the objective space (and/or decision space) is convex, i.e. if v, v' are the vectors of objective values for two solutions, then every convex combination of v, v' is also the vector of values for some solution. This is the case for example for *Multiobjective Linear Programming* (MOLP): minimize (or maximize) a collection of linear functions subject to a set of linear constraints. In this case, the Pareto curve is a polygonal line for two objectives (a polyhedral surface for more). Although there is an infinite set of Pareto points, a finite set of points, namely the vertices of the curve, suffice to represent the curve; every other Pareto point is a convex combination of vertices. Indeed, MOLP has been studied thoroughly and several algorithms (e.g. Multiobjective Simplex) have been developed to generate all the vertices of the Pareto set, see e.g. [Ehr, Ze]. Another example is the design of optimal policies for Markov decision processes with multiple objectives corresponding to the probabilities of satisfaction of a set of properties of the execution or to given discounted rewards [EKVY, CMH]; in effect, these problems can be reduced in both cases to MOLP problems.

Convexity can arise in various other ways, in both continuous and discrete problems, even if it is not present originally. In several applications, solutions that are dominated by convex combinations of other solutions may be regarded as inferior and thus not desirable. The multicriteria literature uses sometimes the term "efficient" for a Pareto solution and "supported efficient" for a solution that is not strictly dominated by the convex combination of other solutions. Thus, sometimes only supported efficient solutions are sought. These are the solutions whose values lie on the Pareto set of the convex hull of all solution points. This "boundary" set can be represented by its extreme points which we call the *convex Pareto set* of the instance. Of course, if the objective space is convex, then the boundary set coincides with the Pareto set. Second, we note that many popular methods for generating Pareto points (called weighted-sum methods) are based on assigning weights $w_i \geq 0$ to the different objectives f_i (assume for simplicity they are all minimization objectives), and minimize the linear combining function $\sum_i w_i f_i$. This is done for a sequence of weight tuples and then the 'dots are connected' to form a representation of the boundary (lower envelope) of the objective space. The optimal solutions of monotone linear combining functions are indeed Pareto solutions, though in general not all Pareto solutions can be so generated, in particular those that are not supported efficient solutions. Thus, the representation that is obtained is actually a representation of the Pareto set of the convex hull. In some applications we may actually want to depict this convex trade-off curve; for example [VV] studies network routing with multiple QoS criteria (these are essentially multiobjective shortest path problems) and the associated trade-off curves, both the exact Pareto curve and the convex (called there relaxed) curve. Third, another case of convexification is when the decision is randomized, not deterministic (pure), i.e. the decision is a probability distribution over the set of solutions, and the figures of merit are the expected values of the objective function. Randomization has the effect of taking the convex hull of the set of solution points. Note that this holds for all types of objectives (both linear and nonlinear).

In the case of combinatorial optimization problems with linear objectives the convex Pareto set is closely related to *parametric optimization* [Gus, Meg1, Meg2]. For example, consider the parametric $s-t$ shortest path problem where each edge e has cost $c_e + \lambda d_e$ that depends on the parameter λ. The length of the shortest path is a piecewise linear concave function of λ whose pieces correspond to the vertices of the convex Pareto curve for the bi-objective shortest path problem with cost vectors c, d on the edges. The size (number of vertices) of the convex Pareto set may be *much* smaller than the size of the Pareto set. For example, the Pareto set for the bi-objective shortest path problem has exponential size in the worst-case, and the same is true for essentially all common combinatorial problems (bi-objective knapsack, spanning trees, etc.) In contrast, the convex Pareto set for bi-objective shortest paths has quasi-polynomial size, $n^{\Theta(\log n)}$ (upper and lower bound) [Gus, Car]; for bi-objective spanning trees it has polynomial size [Chan, Gus]. On the other hand, for bi-objective minimum cost flows (and for LP) the convex Pareto set, which coincides with the vertices of the Pareto set, has exponential size in the worst-case [Ru].

Regarding the approximation of the convex Pareto set, the notion of ϵ-Pareto set is *not* the right one, as it can be very wasteful. For example consider the case of two objectives, and suppose that the Pareto set P is just a straight line segment $\overline{ab}$. The two vertices a, b are not enough to form an ϵ-Pareto set P_ϵ: we may need to add many points along (or close to) the segment $\overline{ab}$ so that every point of P is almost dominated by a point of P_ϵ. This is obviously redundant and ignores the convexity of the setting.

A natural definition of approximation in the convex setting is the following: A set S of solution points is a *ϵ-convex Pareto set* if for every solution point p there is a convex combination of members of S that almost dominates p in all objectives (i.e., is within a factor $1+\epsilon$ of p or better). Such a set of points S can be arbitrarily smaller than the smallest ϵ-Pareto set. Note that for every instance there is a unique convex Pareto set, but not a unique ϵ-convex Pareto set; there are usually many different (nonredundant) such sets and they could have very different sizes. We want to compute one that has the smallest possible size. *This is the problem we address in this paper.*

There is extensive literature in the optimization and management areas on the approximation of (convex) Pareto curves, both in terms of general methods and for specific (continuous and combinatorial) problems; see e.g. [EG, RW] for some references. The bulk of the literature concerns bi-objective problems; only a very small fraction considers 3 or more objectives. There are several methods proposed in the multicriteria literature which compute a sequence of solution points by optimizing weighted linear combinations of the objectives, and then 'connect the dots'. The underlying goal of the methods is basically the same, to obtain a good approximation of the convex Pareto curve with few points. Usually however there are no quantitative guarantees on the approximation error that is achieved by the methods and the size of the computed set. Also in many cases the methods try explicitly to get an even distribution of the points on the surface. However, for the purposes of minimizing the approximation error with a limited number of points, it is better to have an uneven distribution, with a denser representation in the areas of high curvature, and fewer points in the flat areas. The problem is of course that we are not given explicitly the Pareto surface (the whole point is to construct a good, succinct representation), but can only access it indirectly.

The problem of computing an approximate convex Pareto curve of minimum cardinality has some similarities to the problem of computing minimal approximations to polytopes and convex surfaces (which has been studied in computational geometry, e.g. [ABRSY, Cl, MitS]), but some important differences also, the main ones being (i) the reference surface (the convex Pareto curve) is given implicitly, not explicitly, (ii) we have an asymmetric dominance relation here, as opposed to proximity. Also the metrics of proximity used in geometry are usually different than the ratio measure here (which is common in the analysis of approximation algorithms). Some of the techniques developed there however are still useful in our context.

We summarize now the results and the organization of the rest of the paper. In Section 2 we give basic additional definitions and background. In Section 3 we provide a simple necessary and sufficient condition for the polynomial-time constructibility of an ϵ-convex Pareto set (anyone, not necessarily a small one), in terms of the approximate optimization of monotone linear combining functions of the objectives. Sections 4-6 concern the computation/approximation of the smallest ϵ-convex Pareto set CP^*_ϵ. There are two variants of this problem, depending on whether the (objective) space is continuous (convex) or discrete. It turns out that the corresponding problems are related, but not identical. In Section 4 we address the bi-objective problem when the Pareto set is given explicitly in the input. We show that, in both the continuous and the discrete case, a minimum ϵ-convex Pareto set can be computed efficiently. In Section 5 we consider the bi-objective problem when the Pareto set is not given explicitly, but is only accessible through a routine Comb that optimizes (exactly or approximately) monotone linear combinations of the objectives; we seek general-purpose algorithms that use Comb as a black box and achieve guaranteed performance in terms of the approximation error of the convex Pareto set and its size. Note that the Comb-based model is the one used typically in the multicriteria literature, it is consistent with the characterization of Section 3, and applies to a broad class of continuous and discrete multiobjective problems: Linear Programming, Markov Decision Processes, Shortest Paths, Spanning Trees, etc.; all of these have an exact Comb routine. If the Comb routine is exact, we show that in the continuous case we can compute a minimum ϵ-convex Pareto set (with a polynomial number of calls to the Comb routine). In the important special case of Bi-objective Linear Programming, our algorithm uses roughly 2 LP calls per generated point of the minimum ϵ-convex Pareto set. In the discrete case, we can compute a factor 2 approximation, and the factor 2 is intrinsic, in the sense that no general-purpose algorithm can improve it; we show also NP-hardness specifically for the bi-objective shortest path problem. If we have an approximate Comb routine, we present an approximation algorithm for the construction of the minimum ϵ-convex Pareto set that achieves factor 3 in the continuous case and factor 6 in the discrete case; for both cases we show a lower bound of 2. In Section 6 we discuss the problem for $d \geq 3$ objectives. We present upper and lower bounds, both for explicitly given point sets, and for implicitly specified instances. For explicitly given points we provide a constant factor approximation for $d = 3$, and an approximation with logarithmic factor for fixed $d \geq 4$; for arbitrary (unbounded) number d of objectives, the approximation problem is at least as hard as Set Cover (thus is not approximable better than $\Omega(\log n)$). For implicitly given instances we show that no bounded factor can be achieved for the same ϵ; but if we relax the allowed error to any $\epsilon' > \epsilon$, we can compute an ϵ'-convex Pareto set which achieves the same approximation factors (with respect to the minimum ϵ-CP) as the explicit point case. *Due to space constraints, most proofs are omitted.*

2 Preliminaries

A multiobjective optimization problem (MOP) Π has a set $\mathcal{I}_\Pi$ of *valid instances*, and every instance $I \in \mathcal{I}_\Pi$ has an associated set of *feasible solutions* $\mathcal{S}(I)$, usually called the *solution* or *decision space*. There are d objective functions, $f_1, f_2, \ldots, f_d$, each of which maps every instance $I \in \mathcal{I}_\Pi$ and solution $s \in \mathcal{S}(I)$ to a real number $f_j(I, s)$. The problem specifies for each objective whether it is to be maximized or minimized. We denote by $\mathbf{f} = [f_1, f_2, \ldots, f_d]$ the d-vector of objective values, and by $\mathfrak{X}(I)$ the set $\bigcup_{s\in\mathcal{S}(I)} \mathbf{f}(s)$ of the solution values, usually called the *objective space*. As usual in approximation, we assume that the objective functions have positive values. For combinatorial problems, we assume also as usual that the values are rational and can be computed in polynomial time, and we use m to denote the maximum number of bits in the numerator and denominator of a solution value; thus, all values are between 2^{-m} and 2^m. We will consider in this paper also continuous problems, and for these we assume that the objective space $\mathfrak{X}(I)$ is a closed set that is bounded also between 2^{-m} and 2^m in all the objectives.

We say that a d-vector u *dominates* another d-vector v if it is at least as good in all the objectives, i.e. $u_j \geq v_j$ if f_j is to be maximized ($u_j \leq v_j$ if f_j is to be minimized); the domination is strict if at least one of the inequalities is strict. Given a (not necessarily finite) set of points $A \subseteq \mathbf{R}^d_+$, the *Pareto set* of A, denoted by $P(A)$, is the set of undominated points in A. We will use the term *envelope* of A for the Pareto set of the convex hull $\mathcal{CH}(A)$ of A. The *convex Pareto set* of A, denoted by $CP(A)$, is the set of extreme points of the envelope of A, i.e., the points of A that are not dominated by convex combinations of other points. For example, for a discrete problem, the envelope of A is a polyhedral surface and $CP(A)$ is its set of vertices.

We say that a d-vector u *c-covers* another d-vector v if u is at least as good as v up to a factor of c in all the objectives, i.e. $u_j \geq v_j/c$ if f_j is to be maximized ($u_j \leq cv_j$ if f_j is to be minimized). Given $A \subseteq \mathbf{R}^d_+$ and $\epsilon > 0$, an *ϵ-Pareto set* of A, $P_\epsilon(A)$, is a set of points in A that $(1+\epsilon)$-cover all vectors in A. An *ϵ-convex Pareto set* of A, $CP_\epsilon(A)$, is a set of points in A, whose *convex combinations* $(1+\epsilon)$-cover all points in A. Clearly, an ϵ-Pareto set is also an ϵ-convex Pareto set, but the inverse does not generally hold. For a given set A, there may exist many ϵ-(convex) Pareto sets, and they may have very different sizes.

The above definitions apply to any set $A \subseteq \mathbf{R}^d_+$. For a multiobjective optimization problem, A is the objective space $\mathfrak{X}(I)$ of an instance I. We stress that A is not given explicitly, but rather implicitly through the instance I. For example in Multiobjective Linear programming, I is a set of linear constraints and linear objective functions, and A is the (infinite) set of values of all feasible solutions. In the Bi-objective Shortest Path problem, I is a graph G with two (positive) cost functions c, d on its edges (e.g. cost, delay), source node s, target t; the set $\mathcal{S}(I)$ of solutions is the set of $s-t$ paths, and A is the set of the corresponding costs and delays of all the $s-t$ paths. Of course, the objective functions do not have to be linear, for example the function c could represent the capacity of the edges, and the capacity of a path (which we want to maximize) is the minimum capacity of an edge along the path.

It is shown in [PY1] that for every MOP in the aforementioned framework, for every instance I and $\epsilon > 0$, there exists an ϵ-Pareto set (thus, also an ϵ-convex Pareto set) of size $O((4m/\epsilon)^{d-1})$, i.e. polynomial for fixed d. An approximate (convex) Pareto set always exists, but it may not be constructible in polynomial time. We say that the problem of computing an ϵ-(convex) Pareto set for a multiobjective problem Π has a PTAS (resp. FPTAS) if there is an algorithm that for every instance I and $\epsilon > 0$ constructs an ϵ-(convex) Pareto set in time polynomial in the size $|I|$ of the instance I (resp. in time polynomial in $|I|$, the representation size $|\epsilon|$ of ϵ, and in $1/\epsilon$). As shown in [PY1], there is a PTAS or FPTAS for constructing an ϵ-Pareto set iff there is one for the following *GAP Problem*: given an instance I of Π, a (positive rational) d-vector b, and a rational $\delta > 0$, either return a solution whose vector dominates b or report that there does not exist any solution whose vector is better than b by at least a $(1+\delta)$ factor in all of the coordinates.

The problem of computing efficiently not just any ϵ-Pareto set, but one of approximately minimum size was investigated in [VY, DY], and both algorithms and lower bounds were given for the general class of problems that have a GAP routine, as well as for subclasses that include many important well-studied problems like shortest paths and others.

As noted in the introduction, for convex problems the appropriate notion of approximation is the ϵ-convex Pareto set, which is the subject of this paper. A first question is, when can we construct such a set in (fully) polynomial time? Since every ϵ-Pareto set is also an ϵ-convex Pareto set, an (F)PTAS for the GAP problem is a sufficient condition. However, as shown in Section 3, it is not a necessary condition. In that section, we give a necessary and sufficient condition when all the objectives are of the same type (minimization or maximization), in terms of the approximate optimization of positive linear combinations of the objectives. Second, we do not want just any ϵ-convex Pareto set, but we would like to compute the smallest one if possible. We consider both continuous and discrete (combinatorial) problems. For combinatorial problems there are two variants, depending on whether the ϵ-convex Pareto set is allowed to include only the discrete solution points in $A = \mathfrak{X}(I)$ or also their convex combinations, which in effect makes the problem continuous. We denote by $\mathcal{Q}_{\mathrm{D}}(A, \epsilon)$ the former variant, and by $\mathcal{Q}_{\mathrm{C}}(A, \epsilon)$ the latter variant, and denote by $CP^*_{\mathrm{D}}(A, \epsilon)$, $CP^*_{\mathrm{C}}(A, \epsilon)$ respectively the minimum size solutions. Thus, for example if in the bi-objective shortest path problem we may use a probability distribution on the paths, then in effect the problem at hand is the continuous version $\mathcal{Q}_{\mathrm{C}}(A, \epsilon)$.

Notation: We will often assume for concreteness and simplicity that the objectives at hand are minimization objectives, and use the term *lower envelope* of the set A, denoted by $\mathrm{LE}(A)$, for the Pareto set of the convex hull of A, i.e. $\mathrm{LE}(A) = P(\mathcal{CH}(A))$. For two points $p, q \in \mathbf{R}^d_+$ the *ratio distance* between p and q is defined by: $\mathcal{RD}(p, q) = \max\{\max_i(p_i/q_i), 1\}$. The ratio distance between p and q is the minimum value $\rho^* = 1 + \epsilon^*$ of the ratio ρ such that p ρ-covers q. In the bi-objective case, we will use x and y to denote the coordinates of the plane corresponding to the two objectives. Every solution is mapped to a point p on this plane and we use $x(p)$, $y(p)$ to denote its coordinates; that is, $p = \big(x(p), y(p)\big)$.

3 Efficient Computability

Let Π be an optimization problem with d minimization objectives. We define the following associated problem *Comb*: Given an instance I of Π and (a weight-vector) $\mathbf{w} \in \mathbf{R}^d_+$, minimize the combined objective $v = \mathbf{w} \cdot \mathbf{f}$. Analogously, if the f_j's are all maximization objectives, the problem is defined in a similar way, the only difference being that v is to be maximized. (The case of mixed objectives requires more care - see below.) We say that the problem Comb has a PTAS (resp. FPTAS) if there is an algorithm, which given $I \in \mathcal{I}_\Pi$, $\mathbf{w}$ and $\delta > 0$, computes a $(1+\delta)$-approximate optimum and runs in time polynomial in $|I|$ and $|\mathbf{w}|$ (resp. $|I|$, $|\mathbf{w}|$, $|\delta|$ and $1/\delta$). Note that the convex Pareto set $CP(I)$ is the set of all optima for all (infinitely many) possible positive weight-vectors; thus, if we could solve Comb exactly for all such $\mathbf{w}$, we would obtain $CP(I)$. On the other hand, it is easy to see that any ϵ-convex Pareto set contains an $(1+\epsilon)$-approximate optimum for any $\mathbf{w} \in \mathbf{R}^d_+$. We show that the converse is also true, and a polynomial number of calls to an approximate Comb_δ routine for a suitable set of weight-vectors suffices to obtain an ϵ-convex Pareto set. The algorithm is essentially the same as an algorithm given in [PY1], albeit in a more restricted context there where all the objectives are linear. We show however that the algorithm works in general, giving a different (more general) proof of correctness, which is also much simpler.

THEOREM 3.1. *Let the number of objectives d be fixed and of the same type. There is a (F)PTAS for constructing an ϵ-convex Pareto set* iff *the problem Comb admits a (F)PTAS.*

An important corollary of Theorem 3.1 is that the class of problems for which an *ϵ-convex Pareto set* is efficiently constructible is *broader* than the corresponding class for the *ϵ-Pareto set*. Consider for example the multi-objective $s-t$ min-cut problem. Given a graph with a d-vector of weights on each edge and a pair of nodes (s,t), find an $s-t$ cut such that the total weight of the edges crossing the cut (for each of the objectives) is minimized. By Theorem 3.1, for any fixed d there is an FPTAS for constructing an *ϵ-convex Pareto set* for the problem. However, as shown in [PY1], even for $d = 2$, there is no FPTAS for constructing an *ϵ-Pareto set* (unless P=NP).

We comment on the case of mixed objectives: In this case the weights in the Comb problem must be positive for the one type of objectives and negative for the other. If we have an exact algorithm for Comb then we can construct again in polynomial time an ϵ-convex Pareto set; also, all the results in the following sections that use an exact Comb routine hold. However, as far as approximate Comb is concerned, note that the weighted linear combination for mixed objectives may take negative values, and technically speaking, approximation ratios are defined only for positive functions. We can circumvent this by using absolute values, and requiring that the absolute difference between the value of the computed solution and OPT be bounded by $\delta|\text{OPT}|$. Such an approximate Comb routine is also sufficient for the polynomial time construction of an ϵ-CP, and for the relevant algorithms of the following sections.

4 Two Objectives - Explicitly Given Points

We are a given a set A of (rational) points in objective space and $\epsilon > 0$, and we want to compute a minimum ϵ-convex Pareto set CP^*_ϵ of A. Recall that there are two versions of the problem, the continuous version $\mathcal{Q}_\text{C}(A, \epsilon)$, where we may include in CP^*_ϵ convex combinations of points in A, and the discrete version $\mathcal{Q}_\text{D}(A, \epsilon)$, where we can only include points in A. We examine the two versions separately and show:

THEOREM 4.1. *Given a set of points in the plane and a rational parameter $\epsilon > 0$, the problem of computing the smallest ϵ-convex Pareto set can be solved in polynomial time in both the continuous and the discrete case.*

4.1 Convex (Objective) Space - Problem $\mathcal{Q}_\text{C}$.

Assume for concreteness that we have minimization objectives (the same methods apply in the other cases). Let $CP(A) = \{p_1, \ldots, p_n\}$ be the convex Pareto set of A with the points ordered left to right. Thus, the lower envelope of A is a polygonal chain denoted as $\text{LE} = \langle p_1, \ldots, p_n \rangle$. Viewed as a function of x, LE is strictly decreasing and convex. We also consider the curve LE'_ϵ obtained from LE by scaling its points by a factor of $(1 + \epsilon)$ in each coordinate, i.e. $\text{LE}'_\epsilon \doteq \{p' \mid p'/(1+\epsilon) \in \text{LE}\}$. We say that a point q lies *between* the curves LE and LE'_ϵ if q is dominated by some point of LE and is not strictly dominated by any point of LE'_ϵ. As a first observation, note that there always exists an optimal solution to the problem that uses points only from LE. We now proceed to show an equivalent reformulation of the problem $\mathcal{Q}_\text{C}$ that forms the basis for the algorithm.

LEMMA 4.1. *The problem $\mathcal{Q}_\text{C}(A, \epsilon)$ is equivalent to the following: Compute a convex polygonal chain C with minimum number of vertices in* LE *having the following properties: (i) its leftmost (resp. rightmost) vertex $(1 + \epsilon)$-covers the leftmost (resp. rightmost) point of $P(A)$ (ii) the curve C lies* between LE *and* LE'_ϵ.

We say that a point p *ϵ-sees* a point q (or q is *ϵ-visible* from p) if no point of the line segment $\overline{pq}$ is strictly dominated by a point of LE'_ϵ. (In other words, this means that $\overline{pq}$ does not intersect the region in $\mathbf{R}^2$ that is strictly "above and to the right" of LE'_ϵ.) Lemma 4.1 motivates the following simple algorithm: Given $CP(A)$ and ϵ, construct the polygonal chain LE'_ϵ and compute a set of points $Q = \{q_1, q_2, \ldots, q_k\}$ as follows: If p_n has $x(p_n) \leq (1 + \epsilon)x(p_1)$, select p_n and halt. Otherwise, the *leftmost* point $q_1 \in Q$ is the point of LE having x-coordinate $x(q_1) = (1 + \epsilon)x(p_1)$. Point q_{i+1} is the *rightmost* point of LE that is ϵ-visible from q_i. The algorithm terminates when the point $p_n \in \text{LE}$ is $(1 + \epsilon)$-covered by the current point $q_k \in Q$.

The optimality of this algorithm can be shown by induction. It is not hard to see that the algorithm uses a linear number of arithmetic operations (i.e. has time complexity $O(n)$ in the real RAM model). This is not enough however: we need to bound also the bit precision of the computed points because n operations can generate in principle numbers with an exponential number of bits. We show that this does not happen here: the computed points have rational coordinates with polynomially bounded bit complexity.

LEMMA 4.2. *The solution set Q computed by the above algorithm has* total *bit complexity $O(k^2 m)$, where $k = |Q|$ and m is the maximum number of bits required to describe* any *vertex $p \in CP(A)$ and the error ϵ.*

4.2 Discrete (Objective) Space - Problem $\mathcal{Q}_\text{D}$.

In this case, one needs to consider only points of $P(A) = \{p_1, \ldots, p_n\}$ for inclusion in the ϵ-CP, but the optimal solution may need points that are not on the lower envelope (if we ignore such points, we may lose at most a factor of 2 - this holds for more general reasons, see Section 6.) It can be shown that a structural lemma parallel to Lemma 4.1 holds in this setting also. In view of this similarity, a naive approach would involve selecting q_{i+1} to be the *rightmost* point of $P(A)$ ϵ-visible from q_i. It is not hard to construct examples for which this approach is suboptimal.

We now describe a modified method that works. We restrict attention to points of $P(A)$ that are not strictly dominated by LE'_ϵ. For two such points $p, q \in P(A)$ we say that p is *at least as good* as q ($p \succeq q$) if either (i) the *rightmost* vertex $v_p \in \text{LE}'_\epsilon$ ϵ-visible from p, is equal to or lies to the right of the corresponding vertex v_q for q, or (ii) $v_p = v_q = v$ and the line $\overline{pv}$ is equal to or lies above the line $\overline{qv}$ to the *right* of v. Given a set of points S, we say that the point $p \in S$ is *a best* point in the set if for any $q \in S$ it holds $p \succeq q$. (Note that there may exist more than one points with this property; in such a case we can arbitrarily pick one of them.)

The algorithm selects a set of points $Q \subseteq P(A)$ as follows: For the computation of the leftmost point $q_1 \in Q$, consider the set of eligible points $E_1 = \{\sigma \in P(A) \mid x(\sigma) \leq (1 + \epsilon)x(p_1)\}$. If there exists a point in E_1 that $(1 + \epsilon)$-covers p_n, select it and halt. Otherwise, select a best point in E_1. For each $i \geq 2$, select q_i

from the set (of eligible points) $E_i = \{\sigma \in P(A) \mid x(\sigma) > x(q_{i-1})$ and σ is $\epsilon -$ visible from $q_{i-1}\}$. If one of the points in E_i $(1+\epsilon)$-covers p_n, select it and halt. Otherwise, select a best point in E_i and iterate. This modified algorithm is easily seen to run in $O(n^2)$ time and its optimality can be shown by induction.

5 Two Objectives - General Results

In Section 5.1, we consider the case that an exact Comb routine is available and Section 5.2 considers the case of an approximate Comb_δ routine. We design *generic* algorithms, i.e. algorithms which use Comb as a black box.

5.1 Exact Comb routine. We consider separately the discrete and continuous cases.

5.1.1 Discrete Space. We first give a lower bound showing that no generic algorithm can guarantee a ratio better than 2. We then give a matching upper bound (an efficient 2-competitive algorithm).

Computing the smallest ϵ-convex Pareto set is typically NP-hard for common problems even for two objectives. The following proposition is an illustration of this fact. Consider the bi-objective Shortest Path (BSP) problem: Given a graph, "costs" and "delays" for each edge and two specified nodes s and t, compute an $s-t$ path trying to minimize both cost and delay.

Proposition 5.1. *For the BSP problem, for any $k \geq 1$, it is NP-hard to distinguish the case that the minimum size of the optimal ϵ-convex Pareto set is k from the case that it is $k+1$.*

The NP-hardness also holds for the bi-objective spanning tree and many other common combinatorial problems. For generic algorithms, factor 2 is the best we can hope for:

Theorem 5.1. *For two objectives, no generic algorithm having oracle access to Comb for constructing a small ϵ-convex Pareto set can be better than 2-competitive.*

We now sketch a generic algorithm that computes an ϵ-convex Pareto set of size at most twice the optimal. The algorithm applies to all *discrete* bi-objective problems with an exact Comb routine. The idea is to appropriately use the routine so as to "simulate" the optimal algorithm of Section 4 for problem $\mathcal{Q}_{\text{D}}$. Because of the lower bound, this is not exactly possible, since, using the given routine as a black box, we cannot access the solution points that do not lie in the convex Pareto set. However, if we ignore such points we do not lose more than a factor of 2. With that in mind, the generic algorithm outputs a set Q of solution points as follows: We first compute the leftmost and rightmost points of the convex Pareto set (p_{left} and p_{right} respectively). If these two points do not constitute an ϵ-convex Pareto set, we select as q_1 the rightmost solution point in $CP(I)$ that lies at most $(1+\epsilon)$ to the right of p_{left}. The remaining points are selected by the following iterative procedure: The point q_{i+1} is the rightmost point of $CP(I)$ that is ϵ-visible from q_i. To find q_{i+1}, the algorithm has an operation $Next(q_i)$ that simulates visibility by using a "binary search procedure on the slopes" with an application of the given routine at each step of the search. At every step of the search, it calls $\text{Comb}(\lambda, 1)$ to compute the point p_λ minimizing the linear objective $(\lambda, 1)$. Let λ' be the (absolute value of the) slope of the line segment $\overline{q_i p_\lambda}$. Then, $r = \text{Comb}(\lambda', 1)$ is the solution point whose *ratio distance* from $\overline{q_i p_\lambda}$ is *maximum* among all solution points (with x-value) between q_i and p_λ. That is, p_λ is ϵ-visible from q_i iff $\mathcal{RD}(\overline{q_i p_\lambda}, r) \leq 1+\epsilon$. This criterion guides the binary search which terminates the first time we find two adjacent vertices $q_{\text{YES}}, q_{\text{NO}} \in CP(I)$ such that q_{YES} is ϵ-visible from q_i and q_{NO} is not. At this point, we set $q_{i+1} = q_{\text{YES}}$. The discreteness of the space guarantees that this happens after a polynomial number of calls to the Comb routine. Therefore,

Theorem 5.2. *For any discrete bi-objective problem possessing an exact Comb routine, for any $\epsilon > 0$, we can compute a 2-approximation to the optimal ϵ - convex Pareto set in polynomial time.*

5.1.2 Continuous Space. If the objective space is convex, in particular a *convex polytope*, then we can compute in polynomial time an ϵ-convex Pareto set of minimum cardinality by an adaptation of the above scheme. The generic algorithm is similar in spirit to the previous case, the difference being that it simulates the optimal algorithm for $\mathcal{Q}_{\text{C}}$. For the simulation we use a similar binary search procedure. In this case, after we compute the adjacent vertices q_{YES} and q_{NO} such that q_{YES} is ϵ-visible from q_i and q_{NO} is not (as above), we compute the rightmost convex combination q^* of q_{YES} and q_{NO} that is ϵ-visible from q_i. By convexity, we know that q^* corresponds to a solution point and we set $q_{i+1} = q^*$. Lemma 4.2 guarantees that the points q^* selected in this manner have polynomial bit complexity (in $|I|$ and $1/\epsilon$) given that the size of the smallest ϵ-convex Pareto set is polynomial in $|I|$ and $1/\epsilon$. We thus have the following:

Theorem 5.3. *For any bi-objective problem with polyhedral objective space and an exact Comb routine, for any $\epsilon > 0$, we can compute the optimal ϵ - convex Pareto* *set in polynomial time.*

5.1.3 Bi-objective Linear Programming.

An important special case of the continuous space setting is Bi-objective Linear Programming. Even though there has been extensive work on this problem, we are not aware of any optimal approximation algorithm. Many existing algorithms use variations of an intuitive approach which starts with the segment connecting the leftmost and rightmost points of the Pareto curve (which can be found by LP), and then iteratively refines the current polygonal line by introducing additional points that optimize judiciously chosen linear combinations of the objectives, until the desired approximation error ϵ is achieved. Various versions have been introduced under various names (ES, sandwich method, chord rule). Such a method has been studied analytically in [RF] in the context of the bi-objective min cost flow problem (a special case of LP). They use the same multiplicative metric for the approximation error ϵ as we do, and show that the number of points generated is at most pseudopolynomial in the size of the instance and $1/\epsilon$; they do not compare it with the size of the minimum ϵ-CP. With a slightly more careful analysis one can show actually that the number of points is polynomial, however it is suboptimal and it is unclear in fact if it is within any constant factor of OPT. (We can show that by post processing, we can extract a subset of size at most $2 \cdot OPT$.)

Theorem 5.3 implies that we can compute the optimal ϵ-convex Pareto set for bi-objective LP in polynomial time. We show now that we can in fact do it using essentially 2 LP calls per generated point.

THEOREM 5.4. *For bi-objective LP, we can efficiently compute the optimal ϵ-convex Pareto set by solving a* linear *number of LP's. In particular, if k is the size of the optimal ϵ-convex Pareto set, our algorithm solves $2k + 3$ LP's whose size is of the same order as the size of the initial LP.*

SKETCH: We have a decision space $\mathcal{Z} = \{z \in \mathbf{R}^{n\times 1} \mid A \cdot z \geq b, z \geq \mathbf{0}_{n\times 1}\}$ where $A \in \mathbf{Q}^{m\times n}$ and $b \in \mathbf{Q}^{1\times m}$, and two minimization objectives $c, d \in \mathbf{R}^{1\times n}$. The objective space is $\mathfrak{X} = \{(x, y) \in \mathbf{R}^2_+ \mid x = c \cdot z, y = d \cdot z, z \in \mathcal{Z}\}$. The main ingredient of the algorithm is the efficient implementation of the $Next$ operation: Given a point p on $P(\mathfrak{X})$, the Pareto curve of $\mathfrak{X}$, and $\epsilon > 0$, compute the rightmost point $q \in P(\mathfrak{X})$ that is ϵ-visible from p. Let $r^*x + y = t^*$, $r^*, t^* > 0$ be the equation of the line $\overline{pq}$. Since p belongs to this line, it follows that $r^*x(p) + y(p) = t^*$ ($x(p)$, $y(p)$ are constants and r^*, t^* variables in this equation). By construction, r^* is the minimum absolute value of the slope such that for all solution points $(x, y) \in \mathfrak{X}$ it holds $r^*x + y \geq t^*/(1 + \epsilon)$. In other words, we want the following implication to hold: $z \in \mathcal{Z} \Rightarrow r^*(c \cdot z) + (d \cdot z) \geq t^*/(1 + \epsilon)$. By duality, this implication holds iff there exists a vector $w \in \mathbf{R}^{1\times m}_+$ (the dual variables corresponding to the rows of A) such that $r^*c + d \geq wA$ and $w \cdot b \geq t^*/(1 + \epsilon)$. Therefore, we first solve the LP: $\min r^*$ s.t. $\{r^*x(p) + y(p) = t^*, r^*c + d \geq wA, w \cdot b \geq t^*/(1 + \epsilon), w \geq \mathbf{0}_{1\times m}, r^* \geq 0\}$. The solution gives the equation of the line $\overline{pq}$. The point q is the solution point on this line with minimum y-value. To compute it, we solve the following LP: $\min y$ s.t. $\{r^*x + y = t^*, x = c \cdot z, y = d \cdot z, z \in \mathcal{Z}\}$. (Note that the x and y are now variables and r^*, t^* are the parameters of the line computed by the previous LP.) The solution of this LP gives the coordinates of the point q. ■

5.2 Approximate Comb routine.

We first point out that no generic algorithm can be better than 2-competitive in this setting, even if it has access to the GAP_δ routine. Our main result is a generic algorithm that efficiently computes a 6-approximation to the optimal ϵ-convex Pareto set in the discrete case (3 in the continuous) and is applicable to all bi-objective problems (in our general framework) that have a polynomial approximate Comb_δ routine. (We remark that if the GAP_δ routine is available, we can get a 3-approximation for the discrete case as well.)

By adapting the argument in Theorem 5.1, we show the following:

THEOREM 5.5. *For two objectives, no generic algorithm having oracle access to GAP_δ for constructing a small ϵ-convex Pareto set can be better than* 2*-competitive.*

To compute an approximation to the optimal set we will proceed in two phases. In the first phase we compute a δ-convex Pareto set for a particular $\delta < \epsilon$. In the second phase, we delete points from this set until we are left with a small ϵ-convex Pareto set. In particular, the algorithm is the following:

1. Compute a δ-convex Pareto set R_δ by using the generic algorithm of Section 3 for $\delta = \sqrt[4]{1 + \epsilon} - 1$ ($\approx \epsilon/4$ for small ϵ). If $\sqrt[4]{1 + \epsilon}$ is not rational, then we pick δ to be a rational satisfying $(1 + \delta)^4 \leq 1 + \epsilon$ and has bit representation $O(|\epsilon|)$.
2. Use the optimal algorithm for problem $\mathcal{Q}_{\text{D}}$ to compute the smallest ϵ''-convex Pareto set Q of R_δ for $1 + \epsilon'' = (1 + \epsilon)/(1 + \delta)$. Output Q.

The above described algorithm runs in polynomial time and it is not difficult to see that the set Q is an

ϵ-convex Pareto set for the given instance. To prove the desired guarantee we proceed in two phases: In the first phase, we show that the cardinality of the set Q is at most twice the cardinality of the optimal ϵ'-convex Pareto set for the given instance, where $1+\epsilon' = (1+\epsilon)/(1+\delta)^2$. The second and main part of the argument involves showing that the optimal ϵ'-convex Pareto set has size at most 3 times the size of the optimal ϵ-convex Pareto set. Intuitively, this means that, as ϵ decreases, the size of the optimal ϵ-convex Pareto set (for the same instance) does not increase too fast. This holds because of the following geometric lemma that applies to both versions of the problem (i.e. both $\mathcal{Q}_{\mathrm{C}}$ and $\mathcal{Q}_{\mathrm{D}}$):

LEMMA 5.1. *Let $A \subseteq \mathbf{R}_+^2$ be a discrete set of points. For any $\epsilon > 0$ and any $\epsilon' > 0$ satisfying $1+\epsilon' \geq \sqrt{1+\epsilon}$, we have: $|CP^*(A,\epsilon')| \leq 3 \cdot |CP^*(A,\epsilon)|$.*

Unfortunately, the above lemma is dimension-specific; there is no analogue for three or more dimensions, which makes it impossible to guarantee a constant factor in this case, unless we relax ϵ (Theorem 6.3).

If the objective space is convex, we can instead use the algorithm for problem $\mathcal{Q}_{\mathrm{C}}$ in the second step and save a factor of 2 in the approximation.

THEOREM 5.6. *For any bi-objective problem possessing a (fully) polynomial Comb$_\delta$ routine, for any $\epsilon > 0$, we can compute a 6-approximation in the discrete case (3 in the continuous case) to the smallest ϵ - convex Pareto set in (fully) polynomial time.*

If a GAP$_\delta$ routine is available, we can save a factor of 2 by computing a δ-Pareto set in Step 1.

THEOREM 5.7. *For any bi-objective problem possessing a (fully) polynomial GAP$_\delta$ routine, for any $\epsilon > 0$, we can compute a 3-approximation to the smallest ϵ - convex Pareto set in (fully) polynomial time.*

6 d Objectives

In Section 6.1 we analyze the case of explicitly given points. In Section 6.2 we give our generic results.

6.1 Explicitly Given Points. Our main result for this section is the following theorem that applies to both the discrete and continuous versions $\mathcal{Q}_{\mathrm{D}}$ and $\mathcal{Q}_{\mathrm{C}}$. We use the notation $O_d()$ below to indicate that the hidden constant of the big-Oh depends on d.

THEOREM 6.1. *a. For any* fixed d, *we can efficiently approximate the size* OPT_ϵ *of the optimal ϵ-convex Pareto set within a factor of $O_d(\log \mathrm{OPT}_\epsilon)$.*
b. For $d = 3$, we can efficiently approximate OPT_ϵ *within a constant factor.*

First note that $\mathcal{Q}_{\mathrm{C}}$ is a *continuous* problem and it is not clear that, for $d > 2$, it is even in NP. However, if we only consider points in $CP(A)$, we do not lose more than a factor of d in the approximation. To phrase this claim formally, we define the (intermediate) problem $\mathcal{Q}_{\mathrm{C},R}(A,\epsilon)$ (a restriction to both $\mathcal{Q}_{\mathrm{C}}$ and $\mathcal{Q}_{\mathrm{D}}$) as follows: Given $A \subseteq \mathbf{R}_+^d$ and $\epsilon > 0$, compute the smallest ϵ-convex Pareto set of A that is allowed to use points *only* from $CP(A)$. Denote the optimal solution to this problem by $CP^*_{\mathrm{C},R}(A,\epsilon)$. It is not hard to argue that $|CP^*_{\mathrm{C},R}(A,\epsilon)| \leq d \cdot |CP^*_{\mathrm{C}}(A,\epsilon)|$, which in turn implies $|CP^*_{\mathrm{C},R}(A,\epsilon)| \leq d \cdot |CP^*_{\mathrm{C}}(A,\epsilon)| \leq d \cdot |CP^*_{\mathrm{D}}(A,\epsilon)|$. Hence, an r-approximation algorithm for $\mathcal{Q}_{\mathrm{C},R}$ implies a dr-approximation for $\mathcal{Q}_{\mathrm{C}}$.

To prove Theorem 6.1 we formulate the problem $\mathcal{Q}_{\mathrm{D}}$ (and as a consequence $\mathcal{Q}_{\mathrm{C},R}$) as a hitting set problem on a set system $\mathcal{F}$ of bounded VC-dimension. The reduction goes as follows. We consider the class of "supporting" hyperplanes of $(1+\epsilon)\cdot\mathrm{LE}(A)$ with *positive coefficients* and partition it into equivalence classes according to the points of $P(A)$ that lie below each class. $\mathcal{F}$ has an "element" for each point in $P(A)$ and a subset of $P(A)$ for each equivalence class; in particular, the subset that lies below it. (We remark that $\mathcal{F}$ is of size $O(|A|^d)$, and is not given explicitly. It is therefore crucial for the dimension d to be fixed so that it is constructible in polynomial time.) The described set system has VC-dimension at most d. This implies part (a) of Theorem 6.1 by [BG]. In particular, for $d = 3$, $\mathcal{F}$, being a special case of half-spaces in $\mathbf{R}^d$, admits a $1/r$-net of size $O(r)$ that is efficiently constructible [MSW], which in conjunction with [BG] implies part (b) of the theorem.

The following theorem indicates that the problems $\mathcal{Q}_{\mathrm{D}}$ and $\mathcal{Q}_{\mathrm{C}}$ become very hard in high dimensions.

THEOREM 6.2. *If d is unbounded, even if all the solution points are given explicitly in the input, for any $\epsilon > 0$, we cannot approximate the smallest* ϵ-convex Pareto set *on d objectives in polynomial time to a factor better than $\Omega(\log n)$, unless P=NP.*

6.2 General Results. In this section we consider multiobjective problems possessing either a Comb$_\delta$ routine or a GAP$_\delta$ routine. We first show that, for $d \geq 3$ objectives, we are forced to compute an ϵ'-convex Pareto set, where $\epsilon' > \epsilon$, if we are to have a guarantee on its size.

THEOREM 6.3. *For $d \geq 3$, any polynomial generic algorithm having oracle access to GAP$_\delta$ (or Comb$_\delta$) cannot be c-competitive for any c.*

As our main positive result in this section, we show that for any $\epsilon' > \epsilon$, we can get a constant factor approximation to OPT_ϵ for $d = 3$ and a logarithmic approximation for any fixed d, if we spend time polynomial in $1/(\epsilon' - \epsilon)$.

THEOREM 6.4. *a. For any $\epsilon' > \epsilon$ there exists a polynomial generic algorithm that computes an ϵ'-convex Pareto set Q such that $|Q| \leq O_d(\log \mathrm{OPT}_\epsilon) \cdot \mathrm{OPT}_\epsilon$.*
b. For $d = 3$, we can efficiently compute a constant factor approximation to OPT_ϵ.

The following lemma relates the approximability of problem $\mathcal{Q}_{\mathrm{C},R}$ with the problem in hand. Let $\epsilon > 0$ be a given rational number. For any $\epsilon' > \epsilon$, we can find a $\delta > 0$ such that $1/\ \delta = O(1/(\epsilon' - \epsilon))$ satisfying $1 + \epsilon' \geq (1 + \epsilon)(1 + \delta)^2$.

LEMMA 6.1. *Suppose that there exists an r-factor approximation algorithm for $\mathcal{Q}_{\mathrm{C},R}$. Then, for any $\epsilon' > \epsilon$, we can compute an ϵ'-convex Pareto set Q, such that $|Q| \leq dr\mathrm{OPT}_\epsilon$ using $O((m/\delta)^d)$ $Comb_\delta$ calls.*

To prove the lemma, consider the following two-phase generic algorithm: In the first-phase, compute a δ-convex Pareto set using the generic algorithm of Section 3 and in the second-phase use the r-approximation algorithm for $\mathcal{Q}_{\mathrm{C},R}$ to $(1 + \epsilon)(1 + \delta)$-cover this set. It can be shown that this scheme produces an ϵ'-convex Pareto set of the desired cardinality. By Theorem 6.1, problem $\mathcal{Q}_{\mathrm{C},R}$ can be approximated within a factor of $O_d(\log(\mathrm{OPT}_\epsilon))$ for general d and within a constant factor for $d = 3$. Theorem 6.4 follows by combining this fact with Lemma 6.1.

References

[ABRSY] A. Aggarwal, H. Booth, J. O'Rourke, S. Suri, C.K. Yap. Finding minimal convex nested polygons. *Inform. and Control*, 83, pp. 98-110, 1989.

[BG] H. Brönnimann, M.T. Goodrich Almost Optimal Set Covers in Finite VC-Dimension. *Discrete and Computational Geometry*, 14(4): 463-479 (1995).

[Car] P. Carstensen. The complexity of some problems in parametric linear and combinatorial programming. *PhD Thesis, University of Michigan*, 1983.

[Chan] R. Chandrasekaran. Minimal ratio spanning tree. *Networks* 7, pp. 335–342, 1977.

[CHSB] D. L. Craft, T. F. Halabi, H. A. Shih, T. R. Bortfeld. Approximating convex Pareto surfaces in multiobjective radiotherapy planning. *Med. Phys.*, 33, pp. 3399-3407, 2006.

[Cli] J. Climacao, Ed. *Multicriteria Analysis.* Springer-Verlag, 1997.

[Cl] K. Clarkson. Algorithms for polytope covering and approximation. *WADS*, 246-252, 1993.

[CMH] K. Chatterjee, R. Majumdar, T.A. Henzinger. Markov Decision Processes with Multiple Objectives. *STACS*, 325-336, 2006.

[DY] I. Diakonikolas, M. Yannakakis. Small Approximate Pareto Sets for Bi-objective Shortest Paths and Other Problems. In *Proc. APPROX*, 2007.

[Ehr] M. Ehrgott. *Multicriteria optimization*, 2nd edition, Springer-Verlag, 2005.

[EG] M. Ehrgott, X. Gandibleux. An annotated bibliography of multiobjective combinatorial optimization problems. *OR Spectrum* 42, pp. 425-460, 2000.

[EKVY] K. Etessami, M. Kwiatkowska, M. Y. Vardi, and M. Yannakakis. Multi-Objective Model Checking of Markov Decision Processes. In *Proc. TACAS*, 2007.

[FGE] J. Figueira, S. Greco, M. Ehrgott, eds. *Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer, 2005.

[Gus] D. Gusfield. Sensitivity Analysis for Combinatorial Optimization. *PhD Thesis, UC Berkeley*, 1980.

[Han] P. Hansen. Bicriterion Path Problems. *Proc. 3rd Conf. Multiple Criteria Decision Making Theory and Application*, pp. 109–127, Springer Verlag LNEMS 177, 1979.

[KP] V. Koltun, C. H. Papadimitirou. Approximately dominating representatives. *Theoretical Computer Science*, 371, pp. 148-154, 2007.

[Meg1] N. Meggido. Combinatorial Optimization with Rational Objective Functions. *Math of OR*, pp. 414–424, 1979.

[Meg2] N. Meggido. Applying Parallel Computation Algorithms in the Design of Serial Algoritmhs. *JACM*, pp. 852–865, 1983.

[Miet] K. M. Miettinen. *Nonlinear Multiobjective Optimization*, Kluwer, 1999.

[MitS] J. Mitchell and S. Suri. Separation and Approximation of Polyhedral Objects. *SODA*, pp. 296–306, 1992.

[MSW] J.Matoušek, R, Seidel, E. Welzl. How to net a lot with little: small ε-nets for disks and halfspaces. *SoCG*, pp. 16–22, 1990.

[PY1] C.H. Papadimitriou, M. Yannakakis. On the Approximability of Trade-offs and Optimal Access of Web Sources. In *Proceedings 41st IEEE Symp. on Foundations of Computer Science*, 2000.

[PY2] C.H. Papadimitriou, M. Yannakakis. Multiobjective Query Optimization. In *Proc. PODS* 2001.

[RF] G. Ruhe and B. Fruhwirth. Epsilon-optimality for biciteria programs and its application to minimum cost flows. *Computing* , 44, pp. 21–34, 1990.

[Ru] G. Ruhe. Complexity results for multicriteria and parametric network flows using a pathological graph of Zadeh. *Zeitschrift fur Operations Research* , 32, pp. 9–27, 1988.

[RW] S. Ruzika, M. M. Wiecek. Approximation Methods in Multiobjective Programming (Survey paper). *J. Opt. Th. and Appl.*, 126(3), pp. 473-501, 2005.

[VV] P. Van Mieghen, L. Vandenberghe. Trade-off Curves for QoS Routing. In *Proc. INFOCOM*, 2006.

[VY] S. Vassilvitskii, and M. Yannakakis. Efficiently computing succinct trade-off curves. *Theoretical Computer Science* 348, pp. 334-356, 2005.

[Wa] A. Warburton. Approximation of Pareto Optima in Multiple-Objective Shortest Path Problems. *Operations Research*, 35, pp. 70-79, 1987.

[Ze] M. Zeleny. *Linear Multiobjective Programming*, Springer, 1974.