Small Approximate Pareto Sets for Bi-objective Shortest Paths and Other Problems

Ilias Diakonikolas* and Mihalis Yannakakis**

Department of Computer Science
Columbia University
New York, NY 10027, USA
{ilias, mihalis}@cs.columbia.edu

Abstract. We investigate the problem of computing a minimum set of solutions that approximates within a specified accuracy ϵ the Pareto curve of a multiobjective optimization problem. We show that for a broad class of bi-objective problems (containing many important widely studied problems such as shortest paths, spanning tree, and many others), we can compute in polynomial time an ϵ-Pareto set that contains at most twice as many solutions as the minimum such set. Furthermore we show that the factor of 2 is tight for these problems, i.e., it is NP-hard to do better. We present further results for three or more objectives, as well as for the dual problem of computing a specified number k of solutions which provide a good approximation to the Pareto curve.

1 Introduction

In many decision making situations it is typically the case that more than one criteria come into play. For example, when purchasing a product (car, tv, etc.) we care about its cost, quality, etc. When choosing a route we may care about the time it takes, the distance travelled, etc. When designing a network we may care about its cost, its capacity (the load it can carry), its coverage. This type of *multicriteria* or *multiobjective* problems arise across many diverse disciplines, in engineering, in economics and business, healthcare, and others. The area of multiobjective optimization has been (and continues to be) extensively investigated in the management science and optimization communities with many papers, conferences and books (see e.g. [Cli,Ehr,EG,FGE,Miet]).

In multiobjective problems there is typically no uniformly best solution in all objectives, but rather a trade-off between the different objectives. This is captured by the *trade-off* or *Pareto curve*, the set of all solutions whose vector of objective values is not dominated by any other solution. The trade-off curve represents the range of reasonable "optimal" choices in the design space; they are precisely the optimal solutions for all possible global "utility" functions that depend monotonically on the different objectives. A decision maker, presented with the trade-off curve, can select a solution that corresponds best to his/her preferences; of course different users generally may have

* Research partially supported by NSF grant CCF-04-30946 and an Alexander S. Onassis Foundation Fellowship.
** Research partially supported by NSF grant CCF-04-30946.

different preferences and select different solutions. The problem is that the trade-off curve has typically exponential size (for discrete problems) or is infinite (for continuous problems), and hence we cannot construct the full curve. Thus, we have to contend with an approximation of the curve: We want to compute efficiently and present to the decision makers a small set of solutions (as small as possible) that represents as well as possible the whole range of choices, i.e. that provides a good approximation to the Pareto curve. Indeed this is the underlying goal in much of the research in the multiobjective area, with many heuristics proposed, usually however without any performance guarantees or complexity analysis, as we do in theoretical computer science.

In recent years we initiated a systematic investigation [PY1,VY] to develop the 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; that is, for every solution s, the set P_ϵ contains a solution s' that is within a factor $1+\epsilon$ of s, or better, in all the objectives. (As usual in approximation, it is assumed that all objective functions take positive values.) Such an approximation was studied before for certain problems, e.g. multiobjective shortest paths, for which Hansen [Han] and Warburton [Wa] showed how to construct an ϵ-Pareto set in polynomial time (for fixed number of objectives). Note that typically in most real-life multiobjective problems the number of objectives is small. In fact, the great majority of the multiobjective literature concerns the case of two objectives.

Consider a multiobjective problem with d objectives, for example shortest path with cost and time objectives. For a given instance, and error tolerance ϵ, we would like to compute a smallest set of solutions that form an ϵ-Pareto set. Can we do it in polynomial time? If not, how well can we approximate the smallest ϵ-Pareto set? Note that an ϵ-Pareto set is not unique: in general there are many such sets, some of which can be very small and some very large. First, to have any hope we must ensure that there exists at least a polynomial size ϵ-Pareto set. Indeed, in [PY1] it was shown that this is the case for every multiobjective problem with a fixed number of polynomially computable objectives. Second we must be able to construct at least one such set in polynomial time. This is not always possible. A necessary and sufficient condition for polynomial computability for all $\epsilon > 0$ is the existence of a polynomial algorithm for the following *Gap problem*: Given a vector of values b, either compute a solution that dominates b, or determine that no solution dominates b by at least a factor $1+\epsilon$ (in all the objectives). Many multiobjective problems were shown to have such a routine for the Gap problem (and many others have been shown subsequently).

Construction of a polynomial-size approximate Pareto set is useful, but not good enough in itself: For example, if we plan a trip, we want to examine just a few possible routes, not a polynomial number in the size of the map. More generally, in typical multicriteria situations, the selected representative solutions are investigated more thoroughly by the decision maker (designer, physician, corporation, etc.) to assess the different choices and pick the most preferable one, based possibly on additional factors that are perhaps not formalized or not even quantifiable. We thus want to select as small a set as possible that achieves a desired approximation. In [VY] the problem of constructing a minimum ϵ-Pareto set was raised formally and investigated in a general framework. It was shown that for all bi-objective problems with a polynomial-time Gap routine, one can construct an ϵ-Pareto set that contains at most 3 times the number of points of the smallest such set; furthermore, the factor 3 is best possible in the sense that for some problems it is NP-hard to do better. Further results were shown for 3 and more objectives, and for other related questions. Note that although the factor 3 of [VY] is best possible in general for two objectives, one may be able to do better for specific problems.

We show in this paper, that for an important class of bi-objective problems (containing many widely studied natural ones such as shortest paths, spanning tree, knapsack, scheduling problems and others) we can obtain a 2-approximation, and furthermore the factor of 2 is tight for them, i.e., it is NP-hard to do better. Our algorithm is a general algorithm that relies on a routine for a stronger version of the Gap problem, namely a routine that solves approximately the following *Restricted problem*: Given a (hard) bound b_1 for one objective, compute a solution that optimizes approximately the second objective subject to the bound. Many problems (e.g. shortest paths, etc.) have a polynomial time approximation scheme for the Restricted problem. For all such problems, a 2-approximation to the minimum ϵ-Pareto set can be computed in polynomial time. Furthermore, the number of calls to the Restricted routine (and an associated equivalent dual routine) is linear in the size OPT_ϵ of the optimal ϵ-Pareto set.

The bi-objective shortest path problem is probably the most well-studied multiobjective problem. It is the paradigmatic problem for dynamic programming (thus can express a variety of problems), and arises itself directly in many contexts. One area is network routing with various QoS criteria (see e.g. [CX2,ESZ,GR+,VV]). For example, an interesting proposal in a recent paper by Van Mieghen and Vandenberghe [VV] is to have the network operator advertise a portfolio of offered QoS solutions for their network (a trade-off curve), and then users can select the solutions that best fit their applications. Obviously, the portfolio cannot include every single possible route, and it would make sense to select carefully an "optimal" set of solutions that cover well the whole range. Other applications include the transportation of hazardous materials (to minimize risk of accident, and population exposure) [EV], and many others; we refer to the references, e.g. [EG] contains pointers to the extensive literature on shortest paths, spanning trees, knapsack, and the other problems. Our algorithm applies not only to the above standard combinatorial problems, but more generally to any bi-objective problem for which we have available a routine for the Restricted problem; the objective functions and the routine itself could be complex pieces of software without a simple mathematical expression.

After giving the basic definitions and background in Sect. 2, we present in Sect. 3 our general lower and upper bound results for bi-objective problems, as well as applications to specific problems. In Sect. 4 we present some results for $d = 3$ and more objectives. Here we assume only a Gap routine; i.e. these results apply to all problems with a polynomial time constructible ϵ-Pareto set. It was shown in [VY] that for $d = 3$ it is in general impossible to get a constant factor approximation to the optimal ϵ-Pareto set, but one has to relax ϵ. Combining results from [VY] and [KP] we show that for any $\epsilon' > \epsilon$ we can construct an ϵ'-Pareto set of size $cOPT_\epsilon$, i.e. within a (large) constant factor c of the size OPT_ϵ of the optimal ϵ-Pareto set. For general d, the problem can be reduced to a Set Cover problem whose VC dimension and codimension are at most d, and we can construct an ϵ'-Pareto set of size $O(d\log(dOPT_\epsilon))OPT_\epsilon$.

We discuss also the *Dual* problem: For a specified number k of points, find k points that provide the best approximation to the Pareto curve, i.e. that form an ϵ-Pareto set with the minimum possible ϵ. In [VY] it was shown that for $d = 2$ objectives the problem is NP-hard, but we can approximate arbitrarily well (i.e. there is a PTAS) the minimum approximation ratio $\rho^* = 1 + \epsilon^*$. As we'll see, for $d = 3$ this is not possible, in fact one cannot get any multiplicative approximation (unless P=NP). We use a relationship of the Dual problem to the asymmetric k-center problem and techniques from the latter problem to show that the Dual problem can be approximated (for $d = 3$) within a constant power, i.e. we can compute k points that cover every point on the Pareto curve within a factor $\rho' = (\rho^*)^c$ or better in all objectives, for some constant c. For small ρ^*, i.e. when there is a set of k points that provides a good approximation to the Pareto curve, constant factor and constant power are related, but in general of course they are not.

2 Definitions and Background

A multiobjective optimization problem Π has a set $\mathcal{I}_\Pi$ of *valid instances*, every instance $I \in \mathcal{I}_\Pi$ has a set of solutions $\mathcal{S}(I)$. There are d objective functions, $f_1, \ldots, f_d$, each of which maps every instance I and solution $s \in \mathcal{S}(I)$ to a value $f_j(I, s)$. The problem specifies for each objective whether it is to be maximized or minimized. We assume as usual in approximation that the objective functions have positive rational values, and that they are polynomial-time computable. We use m to denote the maximum number of bits in numerator and denominator of the objective function values.

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). Similarly, we define domination between any solutions according to the d-vectors of their objective values. Given an instance I, the *Pareto set* $P(I)$ is the set of undominated d-vectors of values of the solutions in $\mathcal{S}(I)$. Note that for any instance, the Pareto set is unique. (As usual we are also interested in solutions that realize these values, but we will often blur the distinction and refer to the Pareto set also as a set of solutions that achieve these values. If there is more than one undominated solution with the same objective values, $P(I)$ contains one of them.)

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 an instance I and $\epsilon > 0$, an *ϵ-Pareto set* $P_\epsilon(I)$ is a set of d-vectors of values of solutions that $(1 + \epsilon)$-cover all vectors in $P(I)$. For a given instance, there may exist many ϵ-Pareto sets, and they may have very different sizes. It is shown in [PY1] that for every multiobjective optimization problem in the aforementioned framework, for every instance I and $\epsilon > 0$, there exists an ϵ-Pareto set of size $O((4m/\epsilon)^{d-1})$, i.e. polynomial for fixed d.

An approximate Pareto set always exists, but it may not be constructible in polynomial time. We say that a multiobjective problem Π has a polynomial time approximation scheme (respectively a fully polynomial time approximation scheme) if there is an algorithm, which, given instance I and a rational number $\epsilon > 0$, constructs an

ϵ-Pareto set $P_\epsilon(I)$ in time polynomial in the size $|I|$ of the instance I (respectively, in time polynomial in $|I|$, the representation size $|\epsilon|$ of ϵ, and in $1/\epsilon$). Let MPTAS (resp. MFPTAS) denote the corresponding class of problems. There is a simple necessary and sufficient condition [PY1], which relates the efficient computability of an ϵ-Pareto set for a multi-objective problem Π to 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. As shown in [PY1], a problem is in MPTAS (resp. MFPTAS) if and only if there is a subroutine GAP that solves the GAP problem for Π in time polynomial in $|I|$ and $|b|$ (resp. in $|I|$, $|b|$, $|\delta|$ and $1/\delta$).

We say that an algorithm that uses a routine as a black box to access the solutions of the multiobjective problem is *generic*, as it is not geared to a particular problem, but applies to all of the problems for which the particular routine is available. All that such an algorithm needs to know about the input instance is bounds on the minimum and maximum possible values of the objective functions. (For example, if the objective functions are positive rational numbers whose numerators and denominators have at most m bits, then an obvious lower bound on the objective values is 2^{-m} and an obvious upper bound is 2^m; however, for specific problems better bounds may be available.) Based on the bounds, the algorithm calls the given routine for certain values of its parameters, and uses the returned results to compute an approximate Pareto set.

For a given instance, there may exist many ϵ-Pareto sets, and they may have very different sizes. We want to compute one with the smallest possible size, which we'll denote OPT_ϵ. [VY] gives generic algorithms that compute small ϵ-Pareto sets and are applicable to all multiobjective problems in M(F)PTAS, i.e. all problems possessing a (fully) polynomial GAP routine. They consider the following "dual" problems: Given an instance and an $\epsilon > 0$, construct an ϵ-Pareto set of as small size as possible. And dually, given a bound k, compute an ϵ-Pareto set with at most k points that has as small an ϵ value as possible. In the case of two objectives, they give an algorithm that computes an ϵ-Pareto set of size at most $3 \cdot OPT_\epsilon$; they show that no algorithm can be better than 3-competitive in this setting. For the dual problem, they show that the optimal ϵ-value can be approximated arbitrarily closely. For three objectives, they show that no algorithm can be c-competitive for any constant c, unless it is allowed to use a larger ϵ value. They also give an algorithm that constructs an ϵ'-Pareto set of cardinality at most $4 \cdot OPT_\epsilon$, for any $\epsilon' > (1+\epsilon)^2 - 1$.

In a general multiobjective problem we may have both minimization and maximization objectives. In the remainder, we will assume for convenience that all objectives are minimization objectives; this is without loss of generality, since we can simply take the reciprocals of maximization objectives.

Due to space constraints, most proofs are deferred to the full version of this paper.

3 Two Objectives

We use the following notation in this section. Consider the plane whose coordinates correspond to the two objectives. Every solution is mapped to a point on this plane. We use x and y as the two coordinates of the plane. If p is a point, we use $x(p)$, $y(p)$ to denote its coordinates; that is, $p = \big(x(p), y(p)\big)$.

We consider the class of bi-objective problems Π for which we can approximately minimize one objective (say the y-coordinate) subject to a "hard" constraint on the other (the x-coordinate). Our basic primitive is a (fully) polynomial time routine for the following *Restricted problem* (for the y-objective): Given an instance $I \in \mathcal{I}_\Pi$, a (positive rational) bound C and a parameter $\delta > 0$, either return a solution point $\tilde{s}$ satisfying $x(\tilde{s}) \leq C$ and $y(\tilde{s}) \leq (1+\delta) \cdot \min\{y$ over all solutions $s \in \mathcal{S}(I)$ having $x(s) \leq C\}$ or report that there does not exist any solution s such that $x(s) \leq C$. For simplicity, we will drop the instance from the notation and use Restrict$_\delta$ $(y, x \leq C)$ to denote the solution returned by the corresponding routine. If the routine does not return a solution, we will say that it returns NO. We say that a routine Restrict$_\delta$ $(y, x \leq C)$ runs in polynomial time (resp. fully polynomial time) if its running time is polynomial in $|I|$ and $|C|$ (resp. $|I|$, $|C|$, $|\delta|$ and $1/\delta$). The Restricted problem for the x-objective is defined analogously. We will also use the Restricted routines with strict inequality bounds; it is easy to see that they are polynomially equivalent.

Note that in general the two objectives could be nonlinear and completely unrelated. Moreover, it is possible that a bi-objective problem possesses a (fully) polynomial Restricted routine for the one objective, but not for the other.

The considered class of bi-objective problems is quite broad and contains many well-studied natural ones. Applications include the shortest path problem [Han,Wa] and generalizations [EV,GR+,CX2,VV], cost-time trade-offs in query evaluation [PY2], spanning tree [GR,HL] and related problems [CX1]. The aforementioned problems possess a polynomial Restricted routine for *both* objectives. For several other problems [ABK1,ABK2,CJK,DJSS], the Restricted routine is available for one objective *only*, and it is NP-hard to even separately optimize the other objective. An example is the following scheduling problem: We are given a set of n jobs and a fixed number m of machines. Executing job j on machine i requires time p_{ij} and incurs cost c_{ij}. We are interested in the trade-off between makespan and cost. Minimizing the makespan is NP-hard even for $m = 2$, but there is an FPTAS for the Restricted problem for the makespan objective [ABK1].

In Sect. 3.1, we show that, even if the given bi-objective problem possesses a (fully) polynomial Restricted routine *for both objectives*, no generic algorithm can guarantee an approximation ratio better than 2. (This lower bound applies *a fortiori* if the Restricted routine is available for one objective only.) Furthermore, we show that for two such natural problems, namely, the bi-objective shortest path and spanning tree problems, it is NP-hard to do better than 2. In Sect. 3.2 we give a matching upper bound: we present an algorithm that is 2-competitive and applies to all of the problems that possess a polynomial Restricted routine for one of the two objectives.

3.1 Lower bound

To prove a lower bound for a generic procedure, we present two Pareto sets which are indistinguishable from each other using the Restricted routine as a black box, yet whose smallest ϵ-Pareto sets are of different sizes. We omit the proof.

Proposition 1. *Consider the class of bi-objective problems that possess a fully polynomial Restricted routine for both objectives. Then, for any $\epsilon > 0$, there is no polynomial* *time generic algorithm that approximates the size of the smallest ϵ-Pareto set P^*_ϵ to a factor better than 2.*

In fact, we can prove something stronger (assuming P $\neq$ NP) for the bi-objective shortest path (*BSP*) and spanning tree (*BST*) problems. In the *BSP* problem, we are given a graph, positive rational "costs" and "delays" for each edge and two specified nodes s and t. The set of feasible solutions is the set of $s-t$ paths. The *BST* problem is defined analogously. These problems are well-known to possess polynomial Restricted routines for *both* objectives [LR,GR].

Theorem 1. *a. For the bi-objective Shortest Path problem, for any k from $k = 1$ to a polynomial, it is NP-hard to distinguish the case that the minimum size OPT_ϵ of the optimal ϵ-Pareto set is k from the case that it is $2k - 1$.*
b. The same holds for the bi-objective Spanning Tree problem for any fixed k.

The proof is omitted, due to lack of space. For $k = 1$ the theorem says that it is NP-hard to tell if one point suffices or we need at least 2 points for an ϵ approximation. We proved that the theorem holds also for more general k to rule out additive and asymptotic approximations as well. Similar hardness results can be shown for several other related problems.

3.2 Two Objectives Algorithm

Before we give the algorithm, we remark that if we have *exact* (not approximate) Restricted routines for both objectives then we can compute the *optimal* ϵ-Pareto set by a simple greedy algorithm. The algorithm is similar to the one given in [KP,VY] for the (special) case where all the solution points are given explicitly in the input. The Greedy algorithm proceeds by iteratively selecting points $q_1, \ldots, q_k$ in decreasing x (increasing y) as follows: We start by computing a point q'_1 having minimum y coordinate (among all feasible solutions); q_1 is then selected to be the *leftmost* solution point satisfying $y(q_1) \leq (1+\epsilon)y(q'_1)$. During the jth iteration ($j \geq 2$) we initially compute the point q'_j with minimum y-coordinate among all solution points s having $x(s) < x(q_{j-1})/(1+\epsilon)$ and select as q_j the leftmost point which satisfies $y(q_j) \leq (1+\epsilon)y(q'_j)$. The algorithm terminates when the last point selected $(1+\epsilon)$-covers the leftmost solution point. It follows by an easy induction that the set $\{q_1, q_2, \ldots, q_k\}$ is an ϵ-Pareto set of minimum cardinality. This exact algorithm is applicable to biobjective linear programming (and all problems reducible to it, for example biobjective flows), the biobjective *global* mincut problem [AZ] and several scheduling problems [CJK]. For these problems we can compute an ϵ-Pareto set of minimum cardinality.

If we have approximate Restricted routines, one may try to modify the Greedy algorithm in a straightforward way to take into account the fact that the routines are not exact. However, it can be shown that this modified Greedy algorithm is suboptimal, in particular it does not improve on the factor 3 that can be obtained from the general GAP routine (details omitted). More care is required to achieve a factor 2, matching the lower bound. We will describe now how to accomplish this.

Assume that we have an approximate Restricted routine for the y-objective. Our generic algorithm will also use a polynomial routine for the following *Dual Restricted* *problem* (for the x-objective): Given an instance, a bound D and $\delta > 0$, either return a solution $\tilde{s}$ satisfying $y(\tilde{s}) \le (1+\delta) D$ and $x(\tilde{s}) \le \min\{x(s)$ over all solutions s having $y(s) \le D\}$ or report that there does not exist any solution s such that $y(s) \le D$. We use the notation $\text{DualRestrict}_\delta(x, y \le D)$ to denote the solution returned by the corresponding routine. The following proposition establishes the fact that any bi-objective problem that possesses a (fully) polynomial Restricted routine for the one objective, also possesses a (fully) polynomial *Dual Restricted* routine for the other.

Proposition 2. *For any bi-objective optimization problem, the problems Restrict*$_\delta(y,\cdot)$ *and DualRestrict*$_\delta(x,\cdot)$ *are polynomially equivalent.*

Algorithm Description: We first give a high-level overview of the 2-competitive algorithm. The algorithm iteratively selects a set of solution points $\{q_1, \ldots, q_r\}$ (in decreasing x) by judiciously combining the two routines. The idea is, in addition to the Restricted routine (for the y-coordinate), to use the Dual Restricted routine (for the x-coordinate) in a way that circumvents the problems previously identified for the greedy algorithm. More specifically, after computing the point q'_i in essentially the same way as the greedy algorithm, we proceed as follows: We select as q_i a point that: (i) has y-coordinate at most $(1+\epsilon)y(q'_i)/(1+\delta)$ and (ii) has x-coordinate *at most* the minimum x over all solutions s with $y(s) \le (1+\epsilon)y(q'_i)/(1+\delta)^2$ for a suitable δ. This can be done by a call to the Dual Restricted routine for the x-objective. Intuitively this selection means that we give some "slack" in the y-coordinate to "gain" some slack in the x-coordinate. Also notice that, by selecting the point q_i in this manner, there may exist solution points with y-values in the interval $((1+\epsilon)y(q'_i)/(1+\delta)^2, (1+\epsilon)y(q'_i)/(1+\delta)]$ whose x-coordinate is *arbitrarily* smaller than $x(q_i)$. In fact, the optimal point $(1+\epsilon)$-covering q_i can be such a point. However, it turns out that this is sufficient for our purposes and, if δ is chosen appropriately, this scheme can guarantee that the point q_{2i} lies to the left (or has the same x-value) of the i-th rightmost point of the optimal solution. We now proceed with the formal description of the algorithm. In what follows, the error tolerance is set to $\delta \doteq \sqrt[3]{1+\epsilon} - 1$ ($\approx \epsilon/3$ for small ϵ). If $\sqrt[3]{1+\epsilon}$ is not rational, we let δ be a rational that approximates $\sqrt[3]{1+\epsilon} - 1$ from below, i.e. $(1+\delta)^3 \le (1+\epsilon)$, and which has representation size (number of bits) $|\delta| = O(|\epsilon|)$.

Algorithm 2-Competitive
If $\text{Restrict}_{\delta_0 \leftarrow 1}(y, x \le 2^m) = \text{NO}$ **then** halt.
$q'_1 = \text{Restrict}_\delta(y, x \le 2^m)$;
$q_{\text{left}} = \text{DualRestrict}_{\delta_0 \leftarrow 1}(x, y \le 2^m)$; $\quad x_{\min} = x(q_{\text{left}})$;
$\bar{y}_1 = y(q'_1)(1+\delta)$;
$q_1 = \text{DualRestrict}_\delta(x, y \le \bar{y}_1)$;
$\bar{x}_1 = x(q_1)/(1+\epsilon)$;
$Q = \{q_1\}$; $i = 1$;
While ($\bar{x}_i > x_{\min}$) **do**
{ $q'_{i+1} = \text{Restrict}_\delta(y, x < \bar{x}_i)$;
$\bar{y}_{i+1} = [(1+\epsilon)/(1+\delta)] \cdot \max\{\bar{y}_i, y(q'_{i+1})/(1+\delta)\}$;
$q_{i+1} = \text{DualRestrict}_\delta(x, y \le \bar{y}_{i+1})$;
$\bar{x}_{i+1} = x(q_{i+1})/(1+\epsilon)$;
$Q = Q \cup \{q_{i+1}\}$;

$i = i + 1;$ }
Return Q.

Theorem 2. *The above algorithm computes a 2-approximation to the smallest ϵ-Pareto set in time $O(OPT_\epsilon)$ subroutine calls, where $1/\delta = O(1/\epsilon)$.*

Sketch of Proof: We can assume that the solution set is non-empty. In this case, (*i*) the solution point q_{left} has minimum x-value among all feasible solutions and (*ii*) q'_1 has y-value *at most* $(1 + \delta)y_{\min}$. (We denote by $x_{\min}$, $y_{\min}$ the minimum values of the objectives in each dimension.) It is also easy to see that each subroutine call returns a point; so, all the points are well-defined. Let $Q = \{q_1, q_2, \ldots, q_r\}$ be the set of solution points produced by the algorithm. We will prove that the set Q is an ϵ-Pareto set of size at most $2OPT_\epsilon$. We note the following simple properties.

Fact. 1. For each $i \in [r - 1]$ it holds (i) $x(q'_{i+1}) < x(q_i)/(1 + \epsilon)$ and (ii) for each solution point t with $x(t) < x(q_i)/(1 + \epsilon)$, we have $y(t) \geq y(q'_{i+1})/(1 + \delta)$.
2. For each $i \in [r]$ it holds (i) $y(q_i) \leq (1 + \delta)\bar{y}_i$ and (ii) for each solution point t with $y(t) \leq \bar{y}_i$ we have $x(t) \geq x(q_i)$.

We first show that Q is an ϵ-Pareto set. It is not hard to see that the x coordinates of the points $q_1, q_2, \ldots, q_r$ of Q form a strictly decreasing sequence, while this is not necessarily the case with the y coordinates. We claim that the point q_1 $(1 + \epsilon)$-covers all of the solution points that have x-coordinate at least $x(q_1)/(1 + \epsilon)$. To wit, let t be a solution point with $x(t) \geq x(q_1)/(1 + \epsilon)$. It suffices to argue that $y(t) \geq y(q_1)/(1 + \epsilon)$. By property 2-(*ii*) we have $y(q_1) \leq (1 + \delta)\bar{y}_1 = (1 + \delta)^2 y(q'_1)$ and the definition of q'_1 implies that $y(t) \geq y(q'_1)/(1 + \delta)$, *for any solution point* t. By combining these facts we get that for any solution point t it holds $y(t) \geq y(q_1)/(1 + \delta)^3 \geq y(q_1)/(1 + \epsilon)$. Moreover, for each $i \in [r] \setminus \{1\}$ the point q_i $(1 + \epsilon)$-covers all of the solution points that have their x-coordinate in the interval $\big[x(q_i)/(1 + \epsilon), x(q_{i-1})/(1 + \epsilon)\big)$. Let t be a solution point satisfying $x(q_i)/(1+\epsilon) \leq x(t) < x(q_{i-1})/(1+\epsilon)$. Suppose (for the sake of contradiction) that there exists such a point t with $y(t) < y(q_i)/(1 + \epsilon)$. By property 2-(*i*) and the definition of $\bar{y}_i$ this implies $y(t) < \max\{\bar{y}_{i-1}, y(q'_i)/(1 + \delta)\}$. Now since $x(t) < x(q_{i-1})/(1 + \epsilon)$, property 1-(*ii*) gives $y(t) \geq y(q'_i)/(1 + \delta)$. Furthermore, since $x(t) < x(q_{i-1})$, by property 2-(*ii*) it follows that $y(t) > \bar{y}_{i-1}$. Finally note that there are no solution points with x-coordinate smaller than $x(q_r)/(1 + \epsilon)$.

We now bound the size of the set of points Q in terms of the size of the optimal ϵ-Pareto set. Let $P^*_\epsilon = \{p^*_1, p^*_2, \ldots, p^*_k\}$ be the optimal ϵ-Pareto set, where its points p^*_i, $i \in [k]$, are ordered in increasing order of their y- and decreasing order of their x-coordinate. We will show that $|Q| = r \leq 2k$. This follows from the following claim, whose proof is omitted.

Claim. If the algorithm selects a solution point q_{2i-1} (i.e. if $2i - 1 \leq r$), then there must exist a point p^*_i in P^*_ϵ (i.e. it holds $i \leq k$) and if the algorithm selects a point q_{2i}, then $x(p^*_i) \geq x(q_{2i})$.

We now analyze the running time of the algorithm. Let k be the number of points in the smallest ϵ-Pareto set, $k = OPT_\epsilon$. The algorithm involves $r \leq 2k$ iterations of the while loop; each iteration involves two calls to the subroutines. Therefore, the total running time is bounded by $4k$ subroutine calls. □

In the case of the bi-objective shortest path problem, the Restricted problem can be solved in time $O(en/\epsilon)$ for acyclic (directed) graphs [ESZ], and in time $O(en(\log\log n + 1/\epsilon))$ for general graphs [LR] with n nodes and e edges. The Dual Restricted problem can be solved with the same complexity. Thus, our algorithm runs in time $O(en(\log\log n + 1/\epsilon)OPT_\epsilon)$ for general graphs and $O(enOPT_\epsilon/\epsilon)$ for acyclic graphs. The time complexity is comparable or better than previous algorithms, which furthermore do not provide any guarantees on the size.

4 *d* Objectives

The results in this section use the GAP routine and thus apply to all problems in M(F)PTAS.

4.1 Approximation of the optimal ϵ-Pareto set.

Recall that for $d \geq 3$ objectives we are forced to compute an ϵ'-Pareto set, where $\epsilon' > \epsilon$, if we are to have a guarantee on its size [VY]. For any $\epsilon' > \epsilon$, a logarithmic approximation for the problem is given in [VY], by a simple reduction to the Set Cover problem. We can sharpen this result, by exploiting additional properties of the corresponding set system.

Theorem 3. *1. For any $\epsilon' > \epsilon$ there exists a polynomial time generic algorithm that computes an ϵ'-Pareto set Q such that $|Q| \leq O\big(d\log(dOPT_\epsilon)\big) \cdot OPT_\epsilon$.*
2. For $d = 3$, the algorithm outputs an ϵ'-Pareto set Q satisfying $|Q| \leq cOPT_\epsilon$, where c is a constant.

Consider the following problem $\mathcal{Q}(P, \epsilon)$: Given a set of n points $P \subseteq \mathbf{R}^d_+$ as input and $\epsilon > 0$, compute the smallest ϵ-Pareto set of P. It should be stressed that, by definition, the set of points P is given *explicitly* in the input. (Note the major difference with our setting: for a typical multiobjective problem there are exponentially many solution points and they are not given explicitly.) This problem can be solved in linear time for $d = 2$ by a simple greedy algorithm. For $d = 3$ it is NP-hard and can be approximated within some (large) constant factor c [KP]. If d is arbitrary (i.e. part of the input, e.g. $d = n$), the problem is hard to approximate better than within a $\Omega(\log n)$ factor [VY].

The following fact relates the approximability of $\mathcal{Q}$ with the problem of computing a small ϵ'-Pareto set for a multiobjective problem Π, given the GAP primitive. 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 $1 + \epsilon' \geq (1 + \epsilon)(1 + \delta)^2$.

Proposition 3 (implicit in [VY]). *Suppose that there exists an r-factor approximation algorithm for $\mathcal{Q}$. Then, for any $\epsilon' > \epsilon$, we can compute an ϵ'-Pareto set Q, such that $|Q| \leq rOPT_\epsilon$ using $O((m/\delta)^d)$ GAP_δ calls.*

Sketch of Proof: First compute a δ-Pareto set R, by using the original algorithm of [PY1]. Then apply the r-approximation algorithm for $\mathcal{Q}$ to $(1+\epsilon)(1+\delta)$-cover R. It is easy to see that the computed set of points is an ϵ'-Pareto set of the desired cardinality. □

Part 2 of Theorem 3 follows immediately from the fact that $\mathcal{Q}$ is constant factor approximable for $d = 3$ [KP] and the above proposition. We consider the case of general d in the remainder. The problem $\mathcal{Q}(P, \epsilon)$ can be phrased as a set cover problem as follows: For each input point $q \in P$ and $\epsilon > 0$, define $S_{q,\epsilon} = \{x \in \mathbf{R}^d \mid q \leq (1+\epsilon) \cdot x\}$ (the subset of $\mathbf{R}^d$ $(1+\epsilon)$-covered by q). For each point $r \in P$, r is $(1+\epsilon)$-covered by q iff $r \in S_{q,\epsilon}$. Now consider the set system $\mathcal{F}(P, \epsilon) = (P, \mathcal{S}(P, \epsilon))$, where $\mathcal{S}(P, \epsilon) = \{P_{q,\epsilon} \equiv P \cap S_{q,\epsilon} \mid q \in P\}$. Clearly, there is a bijection between set covers of $\mathcal{F}(P, \epsilon)$ and ϵ-Pareto sets of P. For $q \in P$ and $\epsilon > 0$, define $S^D_{q,\epsilon} = \{x \in \mathbf{R}^d \mid x \leq (1+\epsilon) \cdot q\}$. A point r $(1+\epsilon)$-covers q iff $r \in S^D_{q,\epsilon}$. The "dual" set system of $\mathcal{F}(P, \epsilon)$ is defined as $\mathcal{F}^D(P, \epsilon) = (P, \mathcal{S}^D(P, \epsilon))$, where $\mathcal{S}^D(P, \epsilon) = \{P^D_{q,\epsilon} \equiv P \cap S^D_{q,\epsilon} \mid q \in P\}$. In words, the elements are the points of P and for each point $q \in P$ we have a set consisting of the points $r \in P$ that $(1+\epsilon)$-cover q. An ϵ-Pareto set of P is equivalent to a hitting set of $\mathcal{F}^D$.

For a set system $(U, \mathcal{R})$, we say that $X \subseteq U$ is *shattered* by $\mathcal{R}$ if for any $Y \subseteq X$, there exists a set $R \in \mathcal{R}$ with $X \cap R = Y$. The *VC-dimension* of the set system is the maximum size of any set shattered by $\mathcal{R}$. We can show the following:

Proposition 4. *The VC-dimension of the set systems $\mathcal{F}(P, \epsilon)$ and $\mathcal{F}^D(P, \epsilon)$ is upper bounded by d.*

As shown in [BG], for any (finite) set system with VC-dimension d, there exists a polynomial time $O(d \log(dOPT))$-factor approximation algorithm for the minimum *hitting set* problem, where OPT is the cost of the optimal solution. If we apply this result to the dual set system $\mathcal{F}^D(P, \epsilon)$, we conclude:

Proposition 5. *Problem $\mathcal{Q}$ can be approximated within a factor of $O(d \log(dOPT_\epsilon))$.*

Part 1 of Theorem 3 follows by combining Propositions 3 and 5.

4.2 The Dual Problem

For a given number k, we want to find k points that provide the best approximation to the Pareto curve, i.e. such that every Pareto point is ρ^*-covered by one of the k selected points for the minimum possible ratio $\rho^* = 1 + \epsilon^*$. It was shown in [VY] that for $d = 2$ the problem is NP-hard but has a PTAS. We show now that for $d = 3$ any multiplicative factor for the dual problem is impossible, even for explicitly given points; we can only hope for a constant power, and only above a certain constant. We omit the proof.

Theorem 4. *Consider the Dual problem for $d = 3$ objectives and explicitly given points.*
1. It is NP-hard to approximate the minimum ratio ρ^* within any polynomial multiplicative factor.
2. It is NP-hard to compute k points that approximate the Pareto curve with ratio better than $(\rho^*)^{3/2}$.

Of course if ρ^* is upperbounded by a constant, i.e. if k points suffice to provide a good approximation to the Pareto curve, then a constant power of ρ^* is also bounded by a constant. In [VY] the Dual problem was related to the Asymmetric k-center problem, and this was used to show that (i) for any d, a set of k points can be computed that approximates the Pareto curve with ratio $(\rho^*)^{O(\log^* k)}$, and (ii) for unbounded d and explicitly given points, it is hard to do much better. Since the metric ρ for the dual problem is a ratio (multiplicative coverage) versus distance (additive coverage) in the k-center problem, in some sense the analogue of constant factor approximation for the Dual problem is constant power.

Can we achieve a constant power $(\rho^*)^c$ for all problems in MPTAS with a fixed number d of objectives? We show that the answer is Yes for $d = 3$ and provide a conjecture that implies it for general d.

Consider the following *generalization* $\mathcal{Q}'(A, P, \epsilon)$ of problem $\mathcal{Q}$: Given a set of n points $P \subseteq \mathbf{R}_+^d$, a subset $A \subseteq P$ and $\epsilon > 0$, compute the smallest subset $P_\epsilon^*(A) \subseteq P$ that $(1+\epsilon)$-covers A. It is easy to see that for $d = 3$ the arguments of [KP] for $\mathcal{Q}$ can be applied to $\mathcal{Q}'$ as well showing that it admits a constant factor approximation. We believe that in fact for all fixed d there may well be a constant factor approximation. Proving (or disproving) this for $d > 3$ seems quite challenging.

The following weaker statement seems more manageable:

Conjecture 1. For any fixed d, there exists a polynomial time *bicriterion* approximation algorithm for $\mathcal{Q}'(A, P, \epsilon)$, that outputs an $(1+\epsilon)^{f(d)}$-cover $C \subseteq P$ of A, satisfying $|C| \leq g(d) \cdot |P_\epsilon^*(A)|$, for some functions $f, g : \mathbf{N} \to \mathbf{N}$.

For $d = 3$, conjecture 1 holds with $f(3) \leq 2$ and $g(3) \leq 4$. This can be shown by a technical adaptation of the 3-objectives algorithm in [VY], that is omitted here.

For general multiobjective problems with a polynomial GAP routine, we formulate the following conjecture:

Conjecture 2. For any fixed d, there exists a polynomial time generic algorithm, that outputs an $(1+\epsilon)^{f(d)}$-cover C, whose cardinality is $|C| \leq g(d) \cdot OPT_\epsilon$, for some functions $f, g : \mathbf{N} \to \mathbf{N}$.

The case of $d = 3$ is proved in [VY] with $f(3) =$ any constant $c' > 2$ and $g(3) = 4$. Note that, by (a variant of) Proposition 3, Conjecture 1 implies Conjecture 2. The converse is also partially true: Conjecture 2 implies Conjecture 1, if in the statement of the latter, problem $\mathcal{Q}'$ is substituted with problem $\mathcal{Q}$.

In the following theorem, we show that a constant factor bicriterion approximation for $\mathcal{Q}'$ implies a constant power approximation for the problem of computing k solutions that cover the feasible set with minimum ratio, given the GAP routine.

Theorem 5. *Consider a (implicitly represented) d-objective problem in MPTAS and suppose that the minimum achievable ratio with k points is ρ^*.*
1. For $d = 3$ objectives we can compute k points which approximate the Pareto set with ratio $O((\rho^*)^c)$ for some constant c.
2. If Conjecture 1 holds, then the same result holds for any fixed number d of objectives.

Sketch of Proof: Part 1 follows from 2 since Conjecture 1 holds for $d = 3$. To show 2, consider first the following ("dual") problem $\mathcal{D}(P, k)$: We are given *explicitly* a set P of n points in $\mathbf{R}_+^d$ and a positive integer k and we want to compute a subset of P of cardinality (at most) k that ρ-covers P with minimum ratio ρ. Let $\rho^* = 1 + \epsilon^*$ denote the optimal value of the ratio. As shown in [VY], this problem can be reduced to the asymmetric k-center problem. A c-factor approximation algorithm for the asymmetric k-center problem implies a $(\rho^*)^c$-factor approximation algorithm for the problem at hand.

We claim that, if problem $\mathcal{Q}'(A, P, \epsilon)$ admits a $((1 + \epsilon)^{f(d)}, g(d))$-bicriterion approximation, then problem $\mathcal{D}(P, k)$ admits a $(\rho^*)^{h(d)}$ approximation for some function h. This is implied by the above reduction and the following more general fact: If we have an instance of the asymmetric k-center problem (problem $\mathcal{D}(P, k)$ in our setting) such that a certain collection of associated set cover subproblems (which are instances of problem $\mathcal{Q}'(A, P, \epsilon)$ here) admits a constant factor bicriterion approximation (an algorithm that blows up both criteria by a constant factor), then this instance admits a constant factor *unicriterion* approximation. This implication is not stated (or proved) in [PV], but follows easily from their work. One way to prove it is to apply Lemma 5 of [PV] in a recursive manner. This is the approach we follow. We defer the technical details to the full version. Aaron Archer recently informed us [Ar2] that he has an alternative method that yields better constants.

For a general multiobjective problem where the solution points are not given explicitly, we impose a geometric $\sqrt{1 + \delta}$ grid for a suitable δ, call GAP at the grid points, and then apply the above algorithm to the set of points returned. Then the set of k points computed by the algorithm provide a $(1 + \epsilon')^{h(d)}$-cover of the Pareto curve, where $1 + \epsilon' = (1 + \epsilon)(1 + \delta)^2$. □

We should remark that the algorithms of this section are less satisfactory that the bi-objective algorithm of the previous section (and the 2d and 3d algorithms of [VY]) in several respects. One weakness is that the constants c obtained (for $d = 3$) are quite large: in the case of Theorem 3, c is the same constant as in [KP] (which is around 200), and in the case of Theorem 5 the constant that comes out of the recursive method applied to the [VY] algorithm is around 100 (using Archer's new technique instead would reduce it to around 20). A second weakness of the algorithms is that they start by applying the general method of [PY1] calling the GAP routine on a grid, and thus incur always the worst-case time complexity even if there is a very small ϵ-Pareto set. Thus, we view our algorithms in this section mainly as theoretical proofs of principle, i.e. that certain (constant) approximations can be computed in polynomial time, but it would be very desirable and important to improve both the constants and the time.

5 Conclusions

We investigated the problem of computing a minimum set of solutions for a multiobjective optimization problem that represents approximately the whole Pareto curve within a desired accuracy ϵ. We developed tight approximation algorithms for the bi-objective shortest path problem, spanning tree, and a host of other bi-objective problems. Our algorithms compute efficiently an approximate Pareto set that contains at most twice as many solutions as the minimum one; furthermore improving on the factor 2 for these specific problems is NP-Hard. The algorithm works in general for all bi-objective problems for which we have a routine for the Restricted problem of approximating one objective subject to a (hard) bound on the other. The algorithm calls this Restricted routine and a dual one as black boxes and makes quite effective use of them: for every instance, the number of calls is linear (at most 4 times) in the number of points in the optimal solution for that instance.

We presented also results for three and more objectives, both for the problem of computing an optimal ϵ-Pareto set and for the dual problem of selecting a specified number k of points that provide the best approximation of the full Pareto curve. As we indicated at the end of the last section, there is still a lot of room for improvement both in the time complexity and the constants of the approximations achieved. We would like especially to resolve Conjecture 2, hopefully positively. It would be great to have a general efficient method for any (small) fixed number d of objectives that computes for every instance a succinct approximate Pareto set with small constant loss in accuracy and in the number of points, and do it in time proportional to the number of computed points, i.e., the optimal approximate Pareto set for the instance in hand.

References

[ABK1] E. Angel, E. Bampis, A. Kononov. A FPTAS for Approximating the Unrelated Parallel Machines Scheduling Problem with Costs. *Proc. ESA*, pp. 194–205, 2001.

[ABK2] E. Angel, E. Bampis, A. Kononov. On the approximate trade-off for bicriteria batching and parallel machine scheduling problems. *Theoretical Computer Science*, 306(1-3), pp. 319-338 (2003).

[ABV1] H. Aissi, C. Bazgan, D. Vanderpooten. Approximation Complexity of Min-Max (Regret) Versions of Shortest Path, Spanning Tree and Knapsack. *Proc. ESA*, 2005.

[ABV2] H. Aissi, C. Bazgan, D. Vanderpooten. Complexity of Min-Max (Regret) Versions of Cut problems. *Proc. ISAAC*, 2005.

[AN+] H. Ackermann, A. Newman, H. Röglin, and B. Vöcking. Decision Making Based on Approximate and Smoothed Pareto Curves. *Proc. ISAAC*, 2005.

[Ar1] A. F. Archer. Two $O(\log^* k)$-approximation algorithms for the asymmetric k-center problem. *Proc. 8th Conf. on Integer programming and Combinatorial Optimization*, pp. 1-14, 2001.

[Ar2] A. F. Archer. Personal communication, 2007.

[AZ] A. Armon, U. Zwick. Multicriteria Global Minimum Cuts. *Algorithmica*, 46, pp. 15-26, 2006.

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

[CG+] J. Chuzhoy, S. Guha, E. Halperin, S. Khanna, G. Kortsartz, S. Naor. Asymmetric k-center is $\log^* n$-hard to Approximate. *Proc. 36th ACM STOC*, pp. 21-27, 2004.

[CJK] T. C. E. Cheng, A. Janiak, and M. Y. Kovalyov. Bicriterion Single Machine Scheduling with Resource Dependent Processing Times. *SIAM J. Optimization*, 8(2), pp. 617–630, 1998.

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

[CX1] G. Chen, G. Xue. A PTAS for weight constrained Steiner trees in series-parallel graphs. *Theoretical Computer Science*, 1-3(304), pp. 237-247, 2003.

[CX2] G. Chen, G. Xue. K-pair delay constrained minimum cost routing in undirected networks. *Proc. SODA*, pp. 230-231, 2001.

[DJSS] J. Dongarra, E. Jeannot, Erik Saule, Zhiao Shi. Bi-objective Scheduling Algorithms for Optimizing Makespan and Reliability on Heterogeneous Systems. *Proc. SPAA*, 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.

[ESZ] F. Ergun, R. Sinha, and L. Zhang. An improved FPTAS for Restricted Shortest Path. *Information Processing Letters* 83(5):237-239, 2002.

[EV] E. Erkut, V. Verter. Modeling of transport risk for hazardous materials. *Operations Research*, 46, pp. 625-642, 1998.

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

[GJ] M. R. Garey, D. S. Johnson. *Computers and Intractability*. W. H. Freeman, 1979.

[GR] M.X. Goemans, R. Ravi. The Constrained Minimum Spanning Tree Problem. *Proc. SWAT*, 1996, pp.66-75.

[GR+] A. Goel, K. G. Ramakrishnan, D. Kataria, and D. Logothetis. Efficient computation of delay-sensitive routes from one source to all destinations. *Proc. IEEE INFOCOM*, 2001.

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

[Has] R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1), pp. 36–42, 1992.

[HL] R. Hassin, A. Levin. An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem. *SIAM J. Comput.*, 33(2): 261-268 (2004).

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

[LR] D. H. Lorenz, D. Raz. A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 28(5), pp. 213–219, 2001.

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

[PV] R. Panigrahy, S. Vishwanathan. An $O(\log^* n)$ approximation algorithm for the asymmetric p-center problem. *J. of Algorithms*, 27(2), pp. 259-268, 1998.

[PY1] C.H. Papadimitriou, M. Yannakakis. On the Approximability of Trade-offs and Optimal Access of Web Sources. *Proc.41st IEEE FOCS*, 2000.

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

[TZ] G. Tsaggouris, and C.D. Zaroliagis. Multiobjective Optimization: Improved FPTAS for Shortest Paths and Non-linear Objectives with Applications. *Proc. ISAAC*, pp. 389-398, 2006.

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