On the Approximability of Trade-offs and Optimal Access of Web Sources

(Extended Abstract)

CHRISTOS H. PAPADIMITRIOU
Computer Science Dept.
U.C. Berkeley
christos@cs.berkeley.edu

and

MIHALIS YANNAKAKIS
Computing Sciences Research
Bell Labs
mihalis@research.bell-labs.com

Abstract

We study problems in multiobjective optimization, *in which solutions to a combinatorial optimization problem are evaluated with respect to several cost criteria, and we are interested in the trade-off between these objectives (the so-called* Pareto curve*). We point out that, under very general conditions, there is a polynomially succinct curve that ϵ-approximates the Pareto curve, for any $\epsilon > 0$. We give a necessary and sufficient condition under which this approximate Pareto curve can be constructed in time polynomial in the size of the instance and $1/\epsilon$. In the case of multiple linear objectives, we distinguish between two cases: When the underlying feasible region is convex, then we show that approximating the multi-objective problem is equivalent to approximating the single-objective problem. If, however, the feasible region is discrete, then we point out that the question reduces to an old and recurrent one: How does the complexity of a combinatorial optimization problem change when its feasible region is intresected with a hyperplane with small coefficients; we report some interesting new findings in this domain. Finally, we apply these concepts and techniques to formulate and solve approximately a cost-time-quality trade-off for optimizing access to the world-wide web, in a model first studied by Etzioni et al [EHJ+] (which was actually the original motivation for this work).*

1. Introduction

Suppose that you want to retrieve a list of records from the world-wide web. The desired records reside in n known sites, but each site contains a random subset of them. Each site has associated with it a cost c_i for accessing the records, a time delay t_i for obtaining them, and a quality q_i (intuitively, the probability that one particular record will be found in site i). We will end up accessing a subset $S \subseteq \{1, 2, \ldots, n\}$ of the sites. The total cost will be

$$\sum_{i \in S} c_i,$$

the total delay will be

$$\max_{i \in S} t_i,$$

and the overall quality of the result will be

$$1 - \prod_{i \in S} (1 - q_i).$$

Which subset S should we choose? We call this the WEB ACCESS PROBLEM.

The WEB ACCESS PROBLEM is a problem of *multi-objective optimization,* a research area in the interface between Operations Research and Microeconomics that has been under intense study since the 1950s. (The particular model is a variant of that proposed, and solved in various alternative single-objective formulations, in [EHJ+].) In multiobjective optimization we are interested not in a single optimal solution, but in a complicated object capturing the notion of a "trade-off" and called the *Pareto curve,* the set of all feasible solutions whose vector of the various objectives is not dominated by the vector of another solution. The Pareto curve captures the intuitive notion of a "trade-off." Unfortunately, it is typically exponential in size. Because of this difficulty, computational approaches to multiobjective optimization are usually concerned with less demanding goals, such as optimizing lexicographically the various criteria.

In this paper we study an interesting *approximate version* of this concept, the *ϵ-approximate Pareto curve*. Informally, it is a set of solutions which are not dominated by any other *by a ratio of more than* $1+\epsilon$. We show (Theorem 1) that, under very general conditions, for all $\epsilon > 0$ there is a polynomially (in the size of the instance and $\frac{1}{\epsilon}$) succinct ϵ-approximate Pareto curve. ϵ-approximate Pareto curves are not new; there had been sporadic work in the more computationally-oriented multiobjective optimization literature on constructing the ϵ-approximate Pareto curve for specific problems, starting with Hansen's pioneering result on bicriteria shortest paths [Han]. However, with the exception of certain straight-forward results in [CJK], the complexity issues raised had not been addressed systematically —apparently, neither had the simple existence proof of Theorem 1 been noted.

A polynomially small ϵ-approximate Pareto curve always exists, but it is hard to construct in general. We give a necessary and sufficient condition for its efficient (polynomial in the size of the instance and $\frac{1}{\epsilon}$) construction in terms of the complexity of an appropriately formulated computational problem, which turns out to be precisely the multiobjective analog of the gap problem used in the study of the complexity of ordinary approximation (Theorem 2).

The above two results apply to any multi-objective optimization problem. Of particular interest, however, are the problems with multiple *linear* objectives. We seek general results which relate the complexity of constructing the ϵ-approximate Pareto curve in problems with multiple linear objectives to the complexity of the single-objective problem. If the underlying feasible region is convex (typical example of such problem is MULTI-OBJECTIVE MIN-COST FLOW, the convex closures of discrete problems, etc.), we show that the ϵ-approximate Pareto curve can be constructed efficiently iff the single-objective problem can be so approximated (Theorem 3). Thus, in this case multiple objectives add little to the complexity of the problem. We have two proofs of this theorem. One uses the ellipsoid method and duality, the second is a simple exhaustive (and slower, yet polynomial) algorithm with a rather subtle proof of correctness.

Discrete optimization problems, with even two linear objectives, (shortest path, maximum weight matching, minimum spanning tree, minimum cut) tend to be NP-hard (it is NP-hard even to maximize one objective while keeping the other fixed, because of KNAPSACK). Yet, there seems to be an interesting dichotomy: The three first problems above have an FPTAS for calculating the ϵ-approximate Pareto curve (some had been known, some are easy consequences of algorithmic ideas in the literature combined with our reduction in Theorem 4 explained below; for matching, the approximation scheme is not polynomial but RNC [MVV]), whereas the corresponding problem for BI-OBJECTIVE MIN-CUT is strongly NP-hard (Theorem 5). We show (Theorem 4) that this phenomenon is related to a familiar if somewhat intricate issue in combinatorial optimization, first raised in [PY]: There is an FPTAS for constructing the ϵ-approximate Pareto curve for a given discrete optimization problem A if the following version of A, called EXACT A, is pseudopolynomial-time solvable: Given an instance of A, is there a solution with cost *exactly* K? (Seen in this light, our positive results for convex problems seem intuitive: The exact version of a convex problem is no harder than the original problem, because of convexity.) Of the four discrete problems we mentioned above (shortest path, maximum weight matching, minimum spanning tree, minimum cut) the first three have polynomial exact versions. Consequently, for these problems we have FPTAS's for constructing the ϵ-approximate Pareto curve, for any fixed number of objectives. (As we mentioned in the previous paragraph, multi-objective min-cost flow also has an FPTAS, because of convexity.) In contrast, BI-OBJECTIVE MIN-CUT is strongly NP-hard, and thus no such FPTAS is possible unless P = NP (in fact, since the reduction is from BISECTION WIDTH, and is approximation-preserving, no approximation is forthcoming).

Finally, we apply this theory to the WEB ACCESS PROBLEM introduced above (which was our original motivation for studying this area). We show that for this problem the ϵ-approximate Pareto curve (that is, a small collection of sets of sites S such that no other set of sites dominates any set in the collection by a factor of more than $(1+\epsilon)$ in all criteria) can indeed be constructed in polynomial time; we give a simple algorithm in Section 4 (Theorem 7).

2. Approximate Pareto Curves

The basic ingredients of an *optimization problem* are its set of *instances*, *solutions* and *objective function*. An optimization problem has a set of *instances*, and every instance x has a set of feasible solutions $F(x)$. As usual, for a computational context, instances and solutions are represented by strings, and we assume that solutions are polynomially bounded and polynomially recognizable in the size of the instance. The objective function is given by a polynomial time algorithm f, which given an instance x and a feasible solution s, computes its value $f(x,s)$, a positive rational number. We seek, given x to find $\arg\max f(x,s)$.[1]

We shall assume that our optimization problems are such that if $f(x,s) > 0$ then $f(x,s)$ is between $2^{-p(|x|)}$ and $2^{p(|x|)}$ for some polynomial p; this is a consequence of the polynomial nature of the solutions and the objective function f.

In a *multi-objective optimization problem* we have instead $k \geq 1$ objective functions $f_i, i = 1, \ldots, k$ (all defined on the same set of feasible solutions). Typically the number of objectives is a small constant number k; we will usually take k to be fixed for complexity condsideration in the remainder. It is not immediately obvious what a computational solution of such a problem should entail. In Theoretical CS, in the past such problems were dealt with by bounding all but one objective and optimizing the other (see, e.g., [EHJ+]). In the multi-objective optimization literature on the other hand (eg. [Cli, Har]), there is general agreement that the right solution concept is that of the *Pareto curve*.

Given an instance x of a multi-objective optimization problem, its *Pareto curve* $P(x)$ is the set of all k-vector of values such that for each $v \in P(x)$, (1) there is a feasible solution s such that $f_i(s) = v_i$ for all i, and (2) there is no feasible solution s' such that $f_i(x,s') \geq v_i$ for all i, with a strict inequality for some i. As in ordinary (single-objective) optimization problems, we are not interested only in the values, but also in solutions realizing these values. We will often blur this distinction and refer usually to the Pareto curve $P(x)$ as a set of solutions which achieve these values. (If there is more than one solution with the same f_i values, $P(x)$ contains one of them.)

Intuitively, $P(x)$ contains all undominated solutions, what in Theoretical CS we usually call "trade-offs." This seems conceptually the right idea, but it is often computationally problematic, since $P(x)$ will typically be exponentially large for many reasonable problems. Furthermore, for even the simplest problems (matching, minimum spanning tree, shortest path) and even for two objectives, determining whether a point belongs to the Pareto curve $P(x)$ is NP-hard (one can reduce KNAPSACK to the problem of finding the shortest spanning tree of a graph consisting of pairs of parallel edges in tandem, such that the weight of the spanning tree in another set of weights is bounded by a constant).

It turns out that a good way to define a meaningful computational problem related to multi-objective optimization involves *approximation:*

Given an instance x of a multi-objective optimization problem and an $\epsilon > 0$, an *ϵ-approximate Pareto curve*, denoted $P_\epsilon(x)$, is a set of solutions s such that there is no other solution s' such that, for all $s \in P_\epsilon(x)$ $f_i(x,s') \geq (1+\epsilon) f_i(x,s)$ for some i.

That is, every other solution is alsmost dominated by some solution in $P_\epsilon(x)$, i.e. there is a solution in $P_\epsilon(x)$ that is within a factor of ϵ in all objectives. This is a rather attractive notion. If a succinct such set exists, it is obviously a reasonable answer (if you wish, interface to present to a customer), if the optimization problem being solved has many objectives. Rather surprisingly, there is always an ϵ-approximate Pareto curve which is polynomial in size:

Theorem 1 *For any multi-objective optimization problem and any ϵ there is a $P_\epsilon(x)$ consisting of a number of solutions that is polynomial in $|X|$ and $\frac{1}{\epsilon}$ (but exponential in the number of objectives).*

Sketch: Consider the k-dimensional space of all objectives. Their values range from $1/2^{p(|x|)}$ to $2^{p(|x|)}$ for some polynomial p. Consider now a subdivision of this cube into hyperrectangles, such that, in each dimension, the ratio of the larger to the smaller coordinate is $1+\epsilon$. Obviously, there are $O(\frac{(2p(|x|))^k}{\epsilon^k})$ such subdivisions. We define $P_\epsilon(x)$ by choosing one point of $P(x)$ in each hyperrectangle that contains such a point. It is easy to see that $P_\epsilon(x)$ is indeed an ϵ-approximate Pareto curve. □

The problem, however, is whether $P_\epsilon(x)$ can be constructed in polynomial time or not. It was first observed very early [Han] that, in the case of bicriteria shortest paths, this is possible for all ϵ —a multi-objective FPTAS. This is done by a pseudopolynomial dynamic programming generalization of Dijkstra's algorithm, converted to an FPTAS by standard techniques. More recently, [CJK] repeat this feat for certain bicriteria scheduling problems. (Apparently the simple general existence theorem above had not been observed in that field.)

One question arises: Under what conditions can we expect to have such an algorithm? Obviously, if the underlying single-objective problem is inapproximable, then there is no hope. But what if it is polynomially approximable — or solvable? What are general conditions for the existence of a polynomial algorithm for constructing a $P_\epsilon(x)$? In [CJK] certain straight-forward conditions were given, evaluating the complexity of certain exhaustive algorithms for

[1] Notice that we assume for simplicity a maximization problem. The dichotomy of maximization vs. minimization is a well-known complication that is usually technically inconsequential, but burdens the exposition and notation. As in this paper we deal with multi-objective problems, in which each objective can be independently either a maximization or a minimization problem, the situation is *exponentially* more complicated. In this abstract we shall only deal with maximization problems in our general theory, while we shall take the liberty of also using minimization problems as examples. Notice also that we restricted the objective values to be a positive, since we will discuss approximation; this is the usual restriction in the context of approximation algorithms and ratios. The Pareto curve is defined of course more generally for arbitrary objective functions; one may use the absolute value of the objective functions in this case to approximate the Pareto curve. We will not diverge here.

constructing $P_\epsilon(x)$ in terms of the complexity of the subroutines employed. However, there is no satisfactory, crisp, general condition in the literature.

The following useful condition characterizes exactly the situation, in terms of a multi-objective generalization of the *gap* problem used to characterize approximability in the single-objective domain.

Theorem 2 *There is an algorithm for constructing a $P_\epsilon(x)$ polynomial in $|x|$ and $\frac{1}{\epsilon}$ if and only if the following problem can be so solved: Given x and a k-vector $(b_1, \ldots, b_k)$, either return a solution s with $f_i(x,s) \geq b_i$ for all i, or answer that there is no solution s' with $f_i(x,s') \geq b_i(1+\epsilon)$.*

Sketch: *(If)* Suppose that we are given x and ϵ, and we wish a $P_\epsilon(x)$. Define $\epsilon' = \sqrt{1+\epsilon} - 1 \approx \epsilon/2$, and subdivide the k-space of objectives into hyperrectangles as in the proof of Theorem 1, using ϵ', and for each corner call the gap problem. Keep (an undominated subset of) all solutions returned. It is not hard to see that this is a $P_\epsilon(x)$.

(Only if) Conversely, if we have an ϵ-Pareto set, we can solve the gap problem for any given set of bounds, by looking only at solutions in the set. □

3. Linear Objectives

Let us restrict ourselves to the case in which all $f_i(x,s)$ are *linear*, that is, each s is a nonnegative n-dimensional vector and $f_i(x,s) = v_i \cdot s$, where the v_i's are k nonnegative n-vectors given in x. Armed with Theorem 2, we now wish to answer the following question: Is there a general technique that enables us to compute $P_\epsilon(x)$ whenever the underlying single-objective problem is tractable (or, even more ambitious, approximable)?

As we shall point out, the answer depends on the precise nature of the feasible set. If it is a continuous *convex* set (e.g., the multi-objective min-cost flow problem, in which we are given a flow value and are interested in the tradeoff of several linear cost functions on the edges, and in which any convex combination of solutions is itself a solution), then the answer is "yes." If it is discrete (e.g., minimum spanning tree, matching, shortest path, min-cut), then the answer is much more intriguing: It is usually "yes," by *ad hoc* algorithms. This is inherent, because we show, by a pseudopolynomial reduction, that the existence of an FPTAS depends on the tractability of an intriguing (and studied over the past 20 years [PY, MVV, BP]) "exact" version of A.

3.1. The Linear Convex Case

Let A be a tractable linear convex optimization problem for which we have been given multiple linear objectives $v_1, \ldots, v_k$. Let $M = \lceil 4k^2/\epsilon \rceil$. Consider the following algorithm for choosing a subset of $P(x)$:

given x,
for each vector $w \in \{0, \ldots, M\}^k$ do
  find the optimum of x under objective $v = \sum_i w_i v_i$
return all optima thus found

It is not hard to see that all optima returned are in $P(x)$. In fact, if it were possible to loop over all positive vectors, instead of these bounded integer vectors, then all of $P(x)$ would be recovered. By restricting ourselves to small integer vectors (growing as ϵ decreases), it is intuitive that we are going in the right direction. Indeed, this works in the case in which objectives do not differ by much. (In what follows, it is helpful to consider feasible solutions of A as k-vectors of objective values, not points in a convex set.) Let us call a feasible solution $s \in P(x)$ *balanced* if all objectives at s are within a ratio of 2 of each other. Let us call it M-enabled if there is a vector $w \in \{0, \ldots, M\}^k$ such that s is the optimum for the combined objective $v = \sum_i w_i v_i$.

Lemma 1 *Suppose that a feasible solution $s \in P(x)$ is balanced and not M-enabled. Then there is a convex combination of M-enabled solutions in $P(x)$ that is within a factor of ϵ from s in all objectives.*

Sketch: Let Y be the set of all M-enabled solutions, and consider the k-dimensional cone C defined by $\{w : w \cdot s \geq w \cdot y, y \in Y\}$, the set of all weights on the objectives for which s is better than all points in Y. We know that this cone is non-empty (since $s \in P(x)$), but it contains no point with coordinates $\leq M$. Consider now the maximum rotational cone R (all vectors whose angle with a particular direction —R's axis— is less than a fixed angle) that is inscribed in the cone C (this is the analog for cones of the maximum sphere inscribed in a polytope; ignore degeneracies for this sketch).

First, it is easy to see that the angle of R is less than $\sqrt{k}/M$ —otherwise R contains a point with coordinates between 0 and M, and so does C, a contradiction. Consider the hyperplanes of C that R touches. They correspond to points in Y, call them $y_1, \ldots, y_m$. Now consider s and the line segments L_j between s and the y_j's; they are normal to the hyperplanes that R touches. Consider also the hyperplane H normal to the axis of R through s. It is clear that the lines L_j have angles less than $\sqrt{k}/M$ with H. The claim is now that there is a point that is a convex combination of the y_j's and is very close to s.

To see why, notice first that the unit vectors e_j which are collinear with the $s - y_j$'s have a convex combination $\delta = \sum_j \alpha_j e_j$ that is a vector along the axis of R whose length is at most $\sqrt{k}/M$. Therefore, we can scale this to show that the $s - y_j$'s have a convex combination $\Delta = \sum_j \frac{\alpha_j (s-y_j)}{|s-y_j|C}$, where C is a normalizing constant $C = \sum_j \frac{\alpha_j}{|s-y_j|}$. It is clear that $|\Delta| = |\delta|/C$. Then it follows that $s' = s - \Delta$ is a convex combination of the y_j's, as follows: $s - \Delta = \sum_j \frac{\alpha_j y_j}{|s-y_j|C}$.

We claim that s' is the desired approximation of s. This follows from this chain of inequalities:

$$\frac{|\Delta|}{|s'|} \leq \frac{\sqrt{k}}{M} \frac{1}{|\sum_j \alpha_j y_j / |s-y_j||} \leq \frac{\sqrt{k}}{M} \frac{|s-y_J|}{|y_J|},$$

where $J = \arg\min_j \frac{|y_j|}{|s-y_j|}$. Now the last expression is at most $\frac{\sqrt{k}}{M}(1 + \frac{|s|}{|y_J|})$, and, since s is balanced and y_J is a Pareto point (and thus must dominate s in at least one dimension), this latter expression is at most $2k\sqrt{k}/M$.

Therefore, s is within a relative error of $2k\sqrt{k}/M$ from s', and, since it is balanced, all of its components are within a relative error of $4k^2/M$ of the corresponding component of s', which was to be proved. □

We can use the lemma to show that, when A is a polynomially solvable convex problem, then its multiobjective problem has a polynomial algorithm (in $|x|$ and $1/\epsilon$) for constructing a $P_\epsilon(x)$: Set $M = \lceil 4k^2/\epsilon \rceil$. For each i, loop through the following values of w_i: Starting with 0, increment w_i by 1 until M; after that continue going only through the even numbers until $2M$. Then go on with multiples of 4 until $4M$. Continue in this manner increasing the power of 2 until $2^{2p(|x|)}M$. For each combination of w_i's, find the optimum of x under a single objective $v = \sum_i w_i v_i$. There are $O((8p(|x|)k^2/\epsilon)^k)$ combinations. The optimal solutions computed for these combinations have the property that (the upper envelope of) their convex hull provides an ϵ approximation to the Pareto curve. This follows from the Lemma because, with these scaled values for the w_i, the objective values of any solution can be scaled so that they are all within a ratio of two.

In fact, the result can be improved by assuming that A is efficiently *approximable*, not quite solvable. However, this can also be shown using the ellipsoid algorithm:

Theorem 3 *If A is a linear convex optimization problem, then there is a polynomial algorithm, in $|x|$ and $1/\epsilon$ for constructing $P_\epsilon(x)$, if and only if the single-objective A can be approximated within ϵ, also in time polynomial in $|x|$ and $1/\epsilon$.*

3.2. The Linear Discrete Case

As we noted in the introduction, the discrete case is more intriguing. In the case of a single linear objective function, there is no difference between optimizing the function over a discrete feasible solution set or over its convex hull. In the multi-objective case there is a difference. There may well be points p of the Pareto curve that are dominated by the convex hull of two (or more) other points, i.e. Pareto solutions p that are strictly within (below) the upper envelope of the Pareto set. If we are not interested in such solutions, but rather wish to compute (approximately) the upper envelope, then this amounts to performing the optimization over the convex hull of the discrete feasible set; we can do this using the method of the previous subsection. However, if we are interested in such points p, i.e. we want to approximate the discrete Pareto curve itself, then this is a harder problem: as we shall see it can be intractable even in some cases where we can solve optimally the single objective function problem.

Our general result for the discrete case is a reduction: Let A be a linear discrete optimization problem, that is, a problem in which, for each x, there is a set of nonnegative integer n-vectors $F(x)$ called the *feasible solutions*. As is typical for combinatorial optimization problems, we will assume that the entries of the solutions are bounded by a polynomial in n (for most combinatorial problems the entries are usually 0-1). We say that such a problem has a FPTAS for constructing $P_\epsilon(x)$ if, for any fixed k, $P_\epsilon(x)$ for k objectives can be constructed in time polynomial in $|x|$ and $1/\epsilon$. The *exact version of* A is this problem: Given an instance x of A, and an integer B, is there a feasible solution with cost exactly B? Recall that a pseudopolynomial algorithm is an algorithm that runs in polynomial time in the magnitude (as opposed to the number of bits) of the numbers involved in the problem, in this case the coefficients of the objective function; equivalently, it runs in polynomial time if all numbers are written in unary notation.

Theorem 4 *There is an FPTAS for constructing an approximate Pareto curve for A, if there is a pseudopolynomial algorithm for the exact version of A.*

Sketch: By Theorem 2, an FPTAS exists iff there is an FPTAS for the "gap" problem: Given an instance and a k-tuple of bounds $(b_1, \ldots, b_k)$, either return a solution s with $v_i \cdot s \geq b_i$ for all i, or answer that there is no solution s' with $v_i \cdot s' \geq b_i(1+\epsilon)$. Let m be the largest entry of a feasible solution; m is polynomial in n. Let $r = \lceil nm/\epsilon \rceil$. For each i, define a new objective function g_i whose jth coefficient is $\min(\lfloor v_{ij} r / b_i \rfloor, r)$.

Consider a feasible solution s. Clearly, if $g_i(s) \geq r$ then $f_i(s) \geq b_i$, and on the other hand, if $f_i(s) \geq b_i(1+\epsilon)$ then $g_i(s) \geq r$. Thus, it suffices to determine if there is a solution s with $g_i(s) \geq r$ for all i. Note that the maximum value of $g_i(s)$ is rnm; let $M = rnm + 1$.

Each of the inequalities $g_i(s) \geq r$ can be reduced to the disjunction of polynomially many equalities. Furthermore, k equalities can be combined into one by multiplying the ith equality by M^{i-1} and adding the results, i.e. $g_i(s) = l_i$ for all i iff $\sum M^{i-1} g_i(s) = \sum M^{i-1} l_i$. The objective function $\sum M^{i-1} g_i$ is a linear function with polynomially bounded coefficients, so this problem can be solved in polynomial time. (There is some redundancy in the computation outlined above, but for simplicity we will not elaborate here.) □

Corollary 5 *There are FPTAS's for constructing an ϵ-approximate Pareto curve for the multi-objective versions of these problems (in the case of* MATCHING*, a fully polynomial RNC scheme):*

- SHORTEST PATH
- MINIMUM SPANNING TREE
- MATCHING

Sketch: The EXACT PATH problem is easily pseudopolynomial by dynamic programming. For EXACT SPANNING TREE, the problem was solved in [BP]. Finally, for MATCHING, the case of unary weights can be reduced to 0,1 weights [PY], which can be solved in RNC [MVV]. □

The converse to the theorem does not hold in general because the exact version does not differentiate between the minimization and maximization version of a problem: Consider for example the problem SHORTEST SIMPLE PATH. Clearly, since we consider nonnegative weight functions, the requirement of a simple path makes no difference for either the single or the multiple objective optimization problem. However, the exact version of the simple path problem is NP-hard, since for example finding the maximum value includes the Hamiltonian path problem as a special case.

A polynomial problem for which the exact version is strongly NP-complete (and thus the above Theorem does not apply) is MIN CUT. The reason is, simply, that it shares the same exact version with the NP-hard MAX CUT. As it turns out, the multiobjective problem in this case is intractable.

Theorem 6 *Unless P = NP, there is no FPTAS for constructing an ϵ-approximate Pareto curve for* BI-OBJECTIVE $s - t$ MIN CUT.

Sketch: Reduction from BISECTION WIDTH. Given a graph G with $2n$ nodes, we add two new nodes, a source s and a sink t, and add edges from these two to the other nodes. Define three objective functions f_1, f_2, f_3 (to be minimized) as follows: in f_1, edges from s have weight 1, and the rest of the edges have weight 0; in f_2, edges from t have weight 1 and the rest 0; in f_3, edges of G have weight 1 and the rest 0. We can also get away with two objective functions, by combining two of the them, for example letting $g_2 = n^2 f_2 + f_3$ and $g_1 = f_1$. It is easy to see that any FPTAS for constructing an ϵ-approximate Pareto curve for this instance, for small enough ϵ, would have to come up with the optimum bisection. □

4. An Algorithm for the Web Access Problem

Recall the web access problem, described in the introduction. There is a set of n sites. Each site has associated with it a cost c_i for accessing the records, a time delay t_i for obtaining them, and a quality q_i. We assume that all these numbers are given as rationals. A solution is a subset $S \subseteq \{1, 2, \ldots, n\}$ of the sites that is to be accessed, and there are three objective functions: The total cost $C(S) = \sum_{i \in S} c_i$, the total delay $T(S) = \max_{i \in S} t_i$ and the overall quality of the result $Q(S) = 1 - \prod_{i \in S}(1 - q_i)$. Obviously the first two functions are to be minimized and the third is to be maximized.

Theorem 7 *There is a polynomial algorithm in $|x|$ (the input size) and $1/\epsilon$ for constructing the approximate Pareto curve $P_\epsilon(x)$ for the web access problem.*

Sketch: First note that there are only n possible values for T (as is the case for objective functions of a bottleneck type, i.e. max or min). Order the sites in nondecreasing order by their delay. By Theorem 2 (adjusted for a mixture of minimization and maximization functions), it suffices to solve the following problem: given bounds b_1, b_2, b_3, find a solution S with $C(S) \leq b_1$, $T(S) \leq b_2$, $Q(S) \geq b_3$, or determine that there is no solution with $C(S) \leq b_1(1 - \epsilon)$, $T(S) \leq b_2(1 - \epsilon)$, $Q(S) \geq b_3(1 + \epsilon)$. Actually we only need to approximate the costs; the other quantities can be exact. Let $r = \lceil n/\epsilon \rceil$. Modify each cost c_i to $c'_i = \lceil c_i r / b_1 \rceil$. Clearly, if a subset S has modified cost $C'(S) \leq r$, then its original cost is $C(S) \leq b_1$; on the other hand, if $C(S) \leq b_1(1 - \epsilon)$, then $C'(S) \leq r$. We apply now dynamic programming. Process in order the sites with delay at most b_2. Compute in an array $A[1...r]$ the best quality of a solution using the processed sites for a given modified cost; i.e. after processing the first j sites, the entry $A[l]$ gives the maximum quality of a subset of $\{1, ..., j\}$ with modified cost l. Note that the entries are rationals with a bounded number of bits by the input. The array can be easily updated when processing a new site. Comparing the final entry $A[r]$ with b_3 solves the subproblem for the triple (b_1, b_2, b_3). □

References

[BP] F. Barahona, and W. R. Pulleyblank. Exact Arborescences, Matchings and Cycles. *Discrete Applied Mathematics*, 16, pp. 91–99, 1987.

[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.

[EHJ+] O. Etzioni, S. Hanks, T. Jiang, R. M. Karp, O. Madari, and O. Waarts. Efficient Information Gathering on the Internet. *Proc. 37th IEEE Symp. on Foundations of Computer Science*, pp. 234–243, 1996.

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

[Har] R. Hartley. Survey of Algorithms for Vector Optimization Problems. *Multiobjective Decision Making*, pp. 1–34, S. French, R.Hartley, L.C. Thomas, D. J. White Eds., Academic Press, 1983.

[MVV] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as Esay as Matrix Inversion. *Combinatorica*, 7(1), pp. 105–114, 1987.

[PY] C. H. Papadimitriou and M. Yannakakis. The Complexity of Restricted Spanning Tree Problems. *J. ACM*, 29(2), pp. 285–309, 1982.

[RM+] R. Ravi, M.V. Marathe, S.S. Ravi, D.J. Rosenkrantz, and H.B. Hunt. Many Birds with One Stone: Multi-objective Approximation Algorithms. *Proc. 25th STOC*, pp. 438-447, 1993.

[ST] D Shmoys and E. Tardos. An Approximation Algorithm for the Generalized Assignment Problem. *Mathematical Programming*, 62, pp. 461-474, 1993.