An FPTAS for Optimizing a Class of Low-Rank Functions Over a Polytope†

Shashi Mittal* Andreas S. Schulz**

Abstract

We present a fully polynomial time approximation scheme (FPTAS) for optimizing a very general class of non-linear functions of low rank over a polytope. Our approximation scheme relies on constructing an approximate Pareto-optimal front of the linear functions which constitute the given low-rank function. In contrast to existing results in the literature, our approximation scheme does not require the assumption of quasi-concavity on the objective function. For the special case of quasi-concave function minimization, we give an alternative FPTAS, which always returns a solution which is an extreme point of the polytope. Our technique can also be used to obtain an FPTAS for combinatorial optimization problems with non-linear objective functions, for example when the objective is a product of a fixed number of linear functions. We also show that it is not possible to approximate the minimum of a general concave function over the unit hypercube to within any factor, unless P = NP. We prove this by showing a similar hardness of approximation result for supermodular function minimization, a result that may be of independent interest.

†Last updated on September 7, 2011.

*Operations Research Scientist, Amazon.com, 331 Boren Avenue N, Seattle, WA 98107. This work was done when the author was a graduate student in the Operations Research Center at Massachusetts Institute of Technology. Email: mshashi@alum.mit.edu.

**Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, E62-569, Cambridge MA 02139, USA. Email: schulz@mit.edu.

1 Introduction

Non-convex optimization problems are an important class of optimization problems that arise in many practical situations (see e.g. Horst and Pardalos (1995) for a survey). However, unlike their convex counterpart for which efficient polynomial time algorithms are known (see e.g. Nesterov and Nemirovskii (1961)), non-convex optimization problems have proved to be much more intractable. A major impediment to efficiently solving non-convex optimization problems is the existence of multiple local optima in such problems; thus any algorithm which seeks to find a globally optimal solution (or a solution close to a global optimum) must avoid getting stuck in local optima.

In this paper, we focus on optimizing a special class of non-convex functions, called low-rank functions, over a polytope. Informally speaking, a function has low rank if it depends only on a few linear combinations of the input variables. We present a fully polynomial time approximation scheme (FPTAS) for optimizing a very general class of low-rank functions over a polytope. An FPTAS for a minimization (resp. maximization) problem is a family of algorithms such that for all $\epsilon > 0$ there is a $(1+\epsilon)$-approximation (resp. $(1-\epsilon)$-approximation) algorithm for the problem, and the running time of the algorithm is polynomial in the input size of the problem, as well as in $1/\epsilon$.

Throughout this paper, we use the following definition of a low-rank non-linear function, given by Kelner and Nikolova (2007).

Definition 1.1 *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be of* rank k*, if there exist k linearly independent vectors $a_1, \ldots, a_k \in \mathbb{R}^n$ and a function $g : \mathbb{R}^k \to \mathbb{R}$ such that $f(x) = g(a_1^T x, \ldots, a_k^T x)$.*

The optimization problem we are attempting to solve is

$$\begin{array}{ll} \min & f(x) = g(a_1^T x, \ldots, a_k^T x) \\ \text{s.t.} & x \in P. \end{array}$$

Here, P is a polytope, and g is a continuous function (this guarantees that a minimum exists). We assume that the optimal value of this program is strictly positive; this is necessary for the notion of approximation considered here to be valid. Recent work on optimization problems of this kind has focused on the special case when g is quasi-concave (see e.g. Porembski (2004), Kelner and Nikolova (2007), Goyal and Ravi (2009)); all of these works exploit the fact that the minimum of a quasi-concave function over a polytope is always attained at an extreme point of the polytope (see e.g. Bertsekas, Nedić, and Ozdaglar (2003)). In contrast, our approximation scheme does not require the assumption of quasi-concavity.

In general, non-linear programming problems of this form are known to be NP-hard. Pardalos and Vavasis (1991) proved that minimizing a quadratic function $f(x) = c^T x + \frac{1}{2} x^T Q x$, where the Hessian Q has just one non-zero eigenvalue which is negative (and hence $f(x)$ is a function of rank two), over a polytope is NP-hard. Subsequently, Matsui (1996) proved that minimizing the product of two strictly positive linear functions over a polytope is NP-hard. Both these hardness results imply that minimizing a rank two function over a polytope is NP-hard. In fact, as we show in Section 6, the optimum value of the problem stated above cannot be approximated to within any factor unless P =

NP. Therefore, we will need some extra assumptions on the properties of the function g to obtain an approximation scheme for the optimization problem (see Section 3.1).

We mention a few classes of non-convex optimization problems that we tackle in this paper.

1. **Multiplicative programming problems:** In this case, g has the form $g(y_1, \ldots, y_k) = \prod_{i=1}^k y_i$. It is known that such a function g is quasi-concave (Konno and Kuno 1992), and therefore its minimum is attained at an extreme point of the polytope. Multiplicative objective functions also arise in combinatorial optimization problems. For example, consider the shortest path problem on a graph $G = (V, E)$ with two edge weights $a : E \to \mathbb{Z}_+$ and $b : E \to \mathbb{Z}_+$. In the context of navigation systems, Kuno (1999) discusses the shortest path problem with the objective function $a(P) \cdot b(P)$ (where P is the chosen path), where a corresponds to the edge lengths, and b corresponds to the number of intersections at each edge in the graph. A similar problem is considered by Kern and Woeginger (2007) as well.

2. **Low rank bi-linear forms:** Bi-linear functions have the form $g(x_1, \ldots, x_k, y_1, \ldots, y_k) = \sum_{i=1}^k x_i \cdot y_i$. Such functions do not even possess the generalized convexity properties, such as quasi-concavity or quasi-convexity (Al-Khayyal and Falk 1983). Bi-linear programming problems are of two kinds: *separable*, in which x and y are disjunctively constrained, and *non-separable*, in which x and y appear together in a constraint. A separable bi-linear function has the neat property that its optimum over a polytope is attained at an extreme point of the polytope, and this fact has been exploited for solving such problems (see e.g. Konno (1976)). The non-separable case is harder, and it requires considerably more effort for solving the optimization problem (Sherali and Alameddine 1992). In this paper, we investigate the particular case when the number of bi-linear terms, k, is fixed.

3. **Sum-of-ratios optimization:** Sum-of-ratios functions have the form $g(x_1, \ldots, x_k, y_1, \ldots, y_k) = \sum_{i=1}^k x_i/y_i$. Even for the case of the sum of a linear term and a ratio of two linear terms, the function can have many local optima (Schaible 1977). Further, Matsui (1996) has shown that optimizing functions of this form over a polytope is an NP-hard problem. Problems of this form arise, for example, in multi-stage stochastic shipping problems where the objective is to maximize the profit earned per unit time (Falk and Palocsay 1992). For more applications, see the survey paper by Schaible and Shi (2003) and the references therein.

There are other functions which do not fall into any of the categories above, but for which our framework is applicable; an example is aggregate utility functions (Eisenberg 1961).

Before proceeding further, we state the computational model we are assuming for our algorithmic results to hold:

- The vectors $a_1, \ldots, a_k$ are known to us (i.e. they are part of the input).
- We are given a polynomial time oracle to compute the function g.
- For the polytope P, we have a polynomial time separation oracle.

Our results: The main contributions of this paper are as follows.

1. **FPTAS for minimizing low rank functions over a polytope:** We give an FPTAS for minimizing a low-rank function f over a polytope under very general conditions (Section 3.1). Even though we present our results only for the case of minimization, the method has a straightforward extension for maximization problems as well. The running time of our approximation scheme is exponential in k, but polynomial in $1/\epsilon$ and all other input parameters. Our algorithm relies on deciding feasibility of a polynomial number of linear programs. We emphasize here that this FPTAS does not require quasi-concavity of the function f. To the best of our knowledge, this is the first FPTAS for general non-quasi-concave minimization/non-quasi-convex maximization problems. We then derive approximation schemes for three categories of non-linear programming problems: multiplicative programming (Section 4.1), low-rank bi-linear programming (Section 4.2) and sum-of-ratios optimization (Section 4.3).

2. **Minimizing quasi-concave functions:** For the specific case of quasi-concave minimization, we give an alternative algorithm which returns an approximate solution which is also an extreme point of the polytope P (Section 5). Again, this algorithm relies on solving a polynomial number of linear programs, and it can be extended to the case of quasi-convex maximization over a polytope. As an application of our technique, we show that we can get an FPTAS for combinatorial optimization problems in which the objective is a product of a fixed number of linear functions, provided a complete description of the convex hull of the feasible points in terms of linear inequalities is known. For example, this technique can be used to get an FPTAS for the product version and the mean-risk minimization version of the spanning tree problem and the shortest path problem.

3. **Hardness of approximation result:** We show that unless P = NP, it is not possible to approximate the minimum of a positive valued concave function over a polytope to within any factor, even if the polytope is the unit hypercube (Section 6). This improves upon the $\Omega(\log n)$ inapproximability result given by Kelner and Nikolova (2007). We first show a similar result for unconstrained minimization of a supermodular set function. Then by using an approximation preserving reduction from supermodular function minimization to minimization of its continuous extension over a unit hypercube, we get the desired result. The hardness result for supermodular function minimization is in contrast with the related problem of submodular function maximization which admits a constant factor approximation algorithm (Feige, Mirrokni, and Vondrák 2007). We also give a stronger hardness of approximation result, namely that it is not possible to approximate the minimum of a concave quadratic function (even with just one negative eigenvalue in the Hessian) over a polytope to within any factor, unless P = NP.

The philosophy behind the approximation scheme is to view g as an objective function that combines several objectives ($a_1^T x, \dots, a_k^T x$ in this case) into one. Therefore the idea is to consider the original single-objective optimization problem as a multiple-objective optimization problem. We first construct an *approximate* Pareto-optimal front corresponding to the k linear functions $a_1^T x, \dots, a_k^T x$, and then choose the best solution from this approximate Pareto set corresponding to our objective function as the approximate solution. Constructing the exact Pareto-optimal front for linear functions, in general, is NP-hard, but an approximate Pareto-optimal front can be computed in polynomial time provided k is fixed (Section 2). Once we construct an approximate Pareto set, it is possible to compute an approximate solution for a large class of functions g (see Section 3 for more details).

Related work: An exhaustive reference on algorithms for non-linear programming can be found in Horst and Pardalos (1995). The case of optimizing low-rank non-linear functions is discussed extensively by Konno, Thach, and Tuy (1996). Konno, Gao, and Saitoh (1998) give cutting plane and tabu search algorithms for minimizing low-rank concave quadratic functions. A more recent work by Porembski (2004) deals with minimizing low-rank quasi-concave functions using cutting plane methods. The methods employed in both papers are heuristic, with no theoretical analysis of the running time of the algorithms, or performance guarantee of the solutions obtained. Vavasis (1992) gives an approximation scheme for low-rank quadratic optimization problems (i.e. the case where the Hessian has only a few non-zero eigenvalues.) However, Vavasis uses a different notion of approximation algorithm than the one we use in this paper.

A more theoretical investigation of low-rank quasi-concave minimization was done by Kelner and Nikolova (2007), who give an expected polynomial-time smoothed algorithm for this class of functions over integral polytopes with polynomially many facets. They also give a randomized fully-polynomial time approximation scheme for minimizing a low-rank quasi-concave function over a polynomially bounded polytope (i.e. one in which the l_1-norm of every point contained in the polytope is bounded by a polynomial in n, the dimension of the input space), provided a lower bound on the minimum of the quasi-concave function is known a-priori, and the objective function satisfies a Lipschitz condition. Further, they show that it is NP-hard to approximate the general quasi-concave minimization problem by a ratio better than $\Omega(\log n)$ unless P = NP. More recently, Goyal and Ravi (2009) give an FPTAS for minimizing a class of low-rank quasi-concave functions over convex sets. The particular class of low-rank quasi-concave functions which can be optimized using this technique is similar to the one which we deal with in our paper. Approximation algorithms for minimizing a non-linear function over a polytope without the quasi-concavity assumption have not been studied in the literature so far.

Konno and Kuno (1992) propose a parametric simplex algorithm for minimizing the product of two linear functions over a polytope. Benson and Boger (1997) give a heuristic algorithm for solving the more general linear multiplicative programming problem, in which the objective function can be a product of more than two linear functions. Survey articles for solving multiplicative programming problems can be found in the books by Horst and Pardalos (1995) and Konno, Thach, and Tuy (1996). For the case of combinatorial optimization problems with a product of two linear functions, Kern and Woeginger (2007) and Goyal, Genc-Kaya, and Ravi (2011) give an FPTAS when the description of the convex hull of the feasible solutions in terms of linear inequalities is known. However, the results in both the papers do not generalize to the case when the objective function is a product of more than two linear functions. In contrast, our results easily generalize to this case as well.

For separable bi-linear programming problems, Konno (1976) gives a cutting plane algorithm that returns an approximate locally optimal solution. Al-Khayyal and Falk (1983) handle the non-separable case using branch-and-bound, and they showed that their algorithm is guaranteed to converge to a globally optimal solution of the optimization problem. Another method for solving the non-separable case is the reformulation-linearization technique due to Sherali and Alameddine (1992). This technique is similar to the lift-and-project method for solving mixed integer programs: The algorithm first generates valid quadratic constraints by taking pairwise products of the constraints, then linearizes both the valid quadratic constraints and the bi-linear term to obtain a lower bounding linear program, and finally uses branch-and-bound to solve the resulting reformulation. Minimizing bi-linear functions of low-rank using a parametric simplex algorithm is discussed in the book by Konno, Thach, and Tuy (1996), however their algorithm works for the separable case only. From a theoretical point of view, an advantage of our technique, as compared to most of the existing algorithms in the literature, is that it works equally well for both separable as well as non-separable bi-linear programming problems.

A good reference for algorithms for solving the sum-of-ratios optimization problem is the survey paper by Schaible and Shi (2003). Almost all the existing algorithms for optimizing the sum of ratios of linear functions are heuristic, with no provable bounds on the running time of the algorithm, nor on the quality of the solution obtained. A common approach for solving these problems is to linearize the objective function by introducing a parameter for each ratio in the objective (see e.g. Falk and Palocsay (1992)). In contrast, our algorithm does not need to parametrize the objective function. We give the first FPTAS for this problem, when the number of ratios is fixed. Our algorithm is especially suited for the case where the number of ratios is small, but each ratio depends on several variables.

As mentioned before, the main idea behind our approximation schemes is exploiting the approximate Pareto-optimal front of the corresponding k linear functions. There is a substantial literature on multi-objective optimization and fully polynomial time algorithms for approximating the Pareto-optimal set (Safer and Orlin (1995a), Safer and Orlin (1995b), Papadimitriou and Yannakakis (2000), Safer, Orlin, and Dror (2004), Diakonikolas and Yannakakis (2008)). We use the procedure given by Papadimitriou and Yannakakis (2000) for constructing the approximate Pareto-optimal front in this paper. Multi-objective optimization techniques have been applied in discrete optimization problems (Mittal and Schulz 2008), however this technique has not yet been fully exploited for continuous optimization problems.

2 Preliminaries on Multi-objective Optimization

An instance π of a multi-objective optimization problem Π is given by a set of k functions $f_1, \ldots, f_k$. Each $f_i : X \to \mathbb{R}_+$ is defined over the same set of feasible solutions, X. Here, X is some subset of $\mathbb{R}^n$ (more specifically, we will consider the case when X is a polytope), and k is significantly smaller than n. Let $|\pi|$ denote the binary-encoding size of the instance π. Assume that each f_i takes values in the range $[m, M]$, where $m, M > 0$. We first define the Pareto-optimal front for multi-objective optimization problems.

Definition 2.1 *Let π be an instance of a multi-objective minimization problem. A* Pareto-optimal front*, denoted by $P(\pi)$, is a set of solutions $x \in X$, such that for each $x \in P(\pi)$, there is no $x' \in X$ such that $f_i(x') \leq f_i(x)$ for all i with strict inequality for at least one i.*

In other words, $P(\pi)$ consists of all undominated solutions. For example, if all f_i are linear functions and the feasible set X is a polytope, then the set of function values $(f_1(x), \ldots, f_k(x))$ for $x \in X$ is a polytope in $\mathbb{R}^k$.

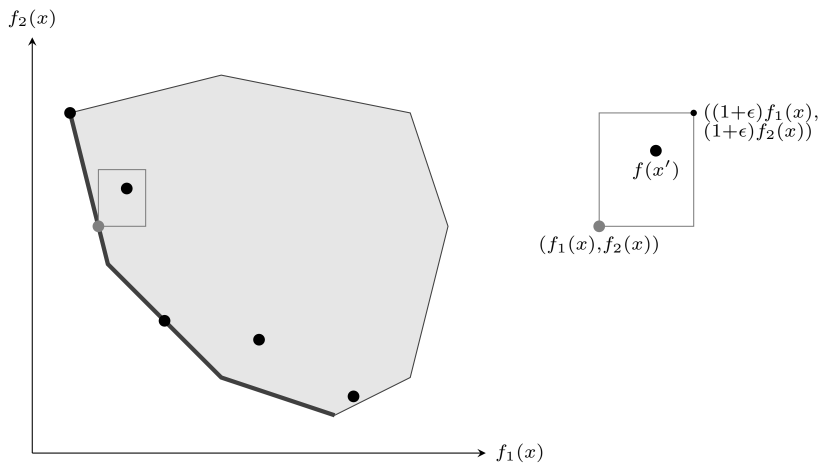

Figure 1: Figure illustrating the concept of Pareto-optimal front (shown by the thick boundary) and approximate Pareto-optimal front (shown by solid black points) for two objectives.

Then $P(\pi)$ in this case is the set of points on the "lower" boundary of this polytope. Still, $P(\pi)$ may have infinitely many points, and it may not be tractable to compute $P(\pi)$. This necessitates the idea of using an approximation of the Pareto-optimal front. One such notion of an approximate Pareto-optimal front is as follows. It is illustrated in Figure 1.

Definition 2.2 *Let π be an instance of a multi-objective minimization problem. For $\epsilon > 0$, an ϵ-*approximate Pareto-optimal front*, denoted by $P_\epsilon(\pi)$, is a set of solutions, such that for all $x \in X$, there is $x' \in P_\epsilon(\pi)$ such that $f_i(x') \leq (1+\epsilon)f_i(x)$, for all i.*

In the rest of the paper, whenever we refer to an (approximate) Pareto-optimal front, we mutually refer to both its set of solutions and their vectors of objective function values. Even though $P(\pi)$ may contain exponentially many (or even uncountably many) solution points, there is always an approximate Pareto-optimal front that has polynomially many elements, provided k is fixed. The following theorem gives one possible way to construct such an approximate Pareto-optimal front in polynomial time. We give a proof of this theorem here, as the details will be needed for designing the FPTAS.

Theorem 2.3 (Papadimitriou and Yannakakis (2000)) *Let k be fixed, and let $\epsilon, \epsilon' > 0$ be such that $(1-\epsilon')(1+\epsilon)^{1/2} = 1$. One can determine a $P_\epsilon(\pi)$ in time polynomial in $|\pi|$ and $1/\epsilon$ if the following 'gap problem' can be solved in polynomial-time: Given a k-vector of values $(v_1, \ldots, v_k)$, either*
(i) *return a solution $x \in X$ such that $f_i(x) \leq v_i$ for all $i = 1, \ldots, k$, or*
(ii) *assert that there is no $x \in X$ such that $f_i(x) \leq (1-\epsilon')v_i$ for all $i = 1, \ldots, k$.*

Proof. Suppose we can solve the gap problem in polynomial time. An approximate Pareto-optimal frontier can then be constructed as follows. Consider the hypercube in $\mathbb{R}^k$ of possible function values given by $\{(v_1, \ldots, v_k) : m \leq v_i \leq M \text{ for all } i\}$. We divide this hypercube into smaller hypercubes, such that in each dimension, the ratio of successive divisions is equal to $1+\epsilon''$, where $\epsilon'' = \sqrt{1+\epsilon} - 1$. For each corner point of all such smaller hypercubes, we solve the gap problem. Among all solutions returned by solving the gap problems, we keep only those solutions that are not Pareto-dominated by any other solution. This is the required $P_\epsilon(\pi)$. To see this, it suffices to prove that every point $x^* \in P(\pi)$ is approximately dominated by some point in $P_\epsilon(\pi)$. For such a solution point x^*, there is a corner point $v = (v_1, \ldots, v_k)$ of some hypercube such that $f_i(x^*) \leq v_i \leq (1+\epsilon'')f_i(x^*)$ for $i = 1, \ldots, k$. Consider the solution of the gap problem for $y = (1+\epsilon'')v$. For the point y, the algorithm for solving the gap problem cannot assert (ii) because the point x^* satisfies $f_i(x^*) \leq (1-\epsilon')y_i$ for all i. Therefore, the algorithm must return a solution x' satisfying $f_i(x') \leq y_i \leq (1+\epsilon)f_i(x^*)$ for all i. Thus, x^* is approximately dominated by x', and hence by some point in $P_\epsilon(\pi)$ as well. Since we need to solve the gap problem for $O((\log{(M/m)}/\epsilon)^k)$ points, this can be done in polynomial time. $\square$

3 The Approximation Scheme

Recall the optimization problem given in Section 1.

$$\begin{aligned} \min \qquad & f(x) = g(a_1^T x, \ldots, a_k^T x) \\ \text{s.t.} \qquad & x \in P. \end{aligned} \tag{1}$$

We further assume that the following conditions are satisfied:

1. $g(y) \leq g(y')$ for all $y, y' \in \mathbb{R}_+^k$ such that $y_i \leq y'_i$ for all $i = 1, \ldots, k$,
2. $g(\lambda y) \leq \lambda^c g(y)$ for all $y \in \mathbb{R}_+^k$, $\lambda > 1$ and some constant c, and
3. $a_i^T x > 0$ for $i = 1, \ldots, k$ over the given polytope.

There are a number of functions g which satisfy conditions 1 and 2, for example the l_p norms (with $c = 1$), bilinear functions (with $c = 2$) and the product of a constant number (say p) of linear functions (with $c = p$). Armed with Theorem 2.3, we now present an approximation scheme for the problem given by (1) under these assumptions. We denote the term $a_i^T x$ by $f_i(x)$, for $i = 1, \ldots, k$. We first establish a connection between optimal (resp. approximate) solutions of (1) and the (resp. approximate) Pareto-optimal front $P(\pi)$ (resp. $P_\epsilon(\pi)$) of the multi-objective optimization problem π with objectives $f_1, \ldots, f_k$ over the same polytope.

Before proceeding, we emphasize that the above conditions are not absolutely essential to derive an FPTAS for the general problem given by (1). Condition 1 may appear to be restrictive, but it can be relaxed, provided that there is at least one optimal solution of (1) which lies on the Pareto-optimal front of the functions $a_1^T x, \ldots, a_k^T x$. For example, the sum-of-ratios form does not satisfy this condition, but still we can get an FPTAS for problems of this form (see Section 4.3).

3.1 Formulation of the FPTAS

Lemma 3.1 *There is at least one optimal solution x^* to* (1) *such that $x^* \in P(\pi)$.*

Proof. Let $\hat{x}$ be an optimal solution of (1). Suppose $\hat{x} \notin P(\pi)$. Then there exists $x^* \in P(\pi)$ such that $f_i(x^*) \leq f_i(\hat{x})$ for $i = 1, \ldots, k$. By Property 1 of g, $g(f_1(x^*), \ldots, f_k(x^*)) \leq g(f_1(\hat{x}), \ldots, f_k(\hat{x}))$. Thus x^* minimizes the function g and is in $P(\pi)$. $\square$

Lemma 3.2 *Let $\hat{x}$ be a solution in $P_\epsilon(\pi)$ that minimizes $f(x)$ over all points $x \in P_\epsilon(\pi)$. Then $\hat{x}$ is a $(1+\epsilon)^c$-approximate solution of* (1)*; that is, $f(\hat{x})$ is at most $(1+\epsilon)^c$ times the value of an optimal solution to* (1).

Proof. Let x^* be an optimal solution of (1) that is in $P(\pi)$. By the definition of ϵ-approximate Pareto-optimal front, there exists $x' \in P_\epsilon(\pi)$ such that $f_i(x') \leq (1+\epsilon) f_i(x^*)$, for all $i = 1, \ldots, k$. Therefore,

$$\begin{aligned} f(x') = g(f_1(x'), \ldots, f_k(x')) &\leq g((1+\epsilon)f_1(x^*), \ldots, (1+\epsilon)f_k(x^*)) \\ &\leq (1+\epsilon)^c g(f_1(x^*), \ldots, f_k(x^*)) = (1+\epsilon)^c f(x^*), \end{aligned}$$

where the first inequality follows from Property 1 and the second inequality follows from Property 2 of g. Since $\hat{x}$ is a minimizer of $f(x)$ over all the solutions in $P_\epsilon(\pi)$, $f(\hat{x}) \leq f(x') \leq (1+\epsilon)^c f(x^*)$. $\square$

When the functions f_i are all linear, the gap problem corresponds to checking the feasibility of linear programs, which can be solved in polynomial time. Hence we get an approximation scheme for solving the problem given by (1). This is captured in the following theorem.

Theorem 3.3 *The gap problem corresponding to the multi-objective version of the problem given by* (1) *can be solved in polynomial time. Therefore, there exists an FPTAS for solving* (1)*, assuming Conditions 1-3 are satisfied.*

Proof. Solving the gap problem corresponds to checking the feasibility of the following linear program:

$$a_i^T x \leq (1-\epsilon')v_i, \quad \text{for} \quad i = 1, \ldots, k, \tag{2a}$$

$$x \in P. \tag{2b}$$

If this linear program has a feasible solution, then any feasible solution to this LP gives us the required answer to question (i). Otherwise, we can answer question (ii) in the affirmative. The feasibility of the linear program can be checked in polynomial time under the assumption that we have a polynomial time separation oracle for the polytope P (Grötschel, Lovász, and Schrijver 1988). The existence of the FPTAS follows from Lemma 3.1 and Lemma 3.2. $\square$

3.2 Outline of the FPTAS

The FPTAS given above can be summarized as follows.

1. Sub-divide the space of objective function values $[m, M]^k$ into hypercubes, such that in each dimension, the ratio of two successive divisions is $1 + \epsilon''$, where $\epsilon'' = (1+\epsilon)^{1/2c} - 1$.

2. For each corner of the hypercubes, solve the gap problem as follows, and keep only the set of non-dominated solutions obtained from solving each of the gap problems.

(a) Check the feasibility of the LP given by (2a)-(2b).

(b) If this LP is infeasible, do nothing. If feasible, then include the feasible point of the LP in the set of possible candidates for points in the approximate Pareto-optimal front.

3. Among the non-dominated points computed in Step 2, pick the point which gives the least value of the function f, and return it as an approximate solution to the given optimization problem.

The running time of the algorithm is $O\big((\frac{\log(M/m)}{\epsilon})^k \cdot LP(n, |\pi|)\big)$, where $LP(n, |\pi|)$ is the time taken to check the feasibility of a linear program in n variables and input size of $|\pi|$ bits. This is polynomial in the input size of the problem provided k is fixed. Therefore when the rank of the input function is a constant, we get an FPTAS for the problem given by (1).

4 Applications of the Approximation Scheme

Using the general formulation given in Section 3.1, we now give approximation schemes for three categories of optimization problems: multiplicative programming, low-rank bi-linear programming and sum-of-ratios optimization.

4.1 Multiplicative Programming Problems

Consider the following multiplicative programming problem for a fixed k:

$$\begin{aligned} \min \quad & f(x) = (a_1^T x) \cdot (a_2^T x) \cdot \ldots \cdot (a_k^T x) \\ \text{s.t.} \quad & x \in P. \end{aligned} \tag{3}$$

We assume that $a_i^T x > 0$, for $i = 1, \ldots, k$, over the given polytope P. In our general formulation, this corresponds to $g(y_1, \ldots, y_k) = \prod_{i=1}^k y_i$ with $c = k$. $f(x)$ has rank at most k in this case. Thus, we get the following corollary to Theorem 3.3.

Corollary 4.1 *Consider the optimization problem given by* (3)*, and suppose that k is fixed. Then the problem admits an FPTAS if $a_i^T x > 0$ for $i = 1, \ldots, k$ over the given polytope P.*

It should be noted that the function f given above is quasi-concave, and so it is possible to get an FPTAS for the optimization problem given by (3) which always returns an extreme point of the polytope P as an approximate solution (see Section 5).

4.2 Low Rank Bi-Linear Programming Problems

Consider a bi-linear programming problem of the following form for a fixed k.

$$\begin{aligned} \min \quad & f(x, y) = c^T x + d^T y + \sum_{i=1}^k (a_i^T x) \cdot (b_i^T y) \\ \text{s.t.} \quad & Ax + By \le h. \end{aligned} \tag{4}$$

where $c, a_i \in \mathbb{R}^m$, $d, b_i \in \mathbb{R}^n$, $A \in \mathbb{R}^{l \times m}$, $B \in \mathbb{R}^{l \times n}$ and $h \in \mathbb{R}^l$. $f(x, y)$ has rank at most $2k + 1$. We have the following corollary to Theorem 3.3.

Corollary 4.2 *Consider the optimization problem given by* (4)*, and suppose that k is fixed. Then the problem admits an FPTAS if $c^T x > 0$, $d^T y > 0$ and $a_i^T x > 0$, $b_i^T y > 0$ for $i = 1, \ldots, k$ over the given polytope $Ax + By \leq h$.*

It should be noted that our method works both in the separable case (i.e. when x and y do not have a joint constraint) as well as in the non-separable case (i.e. when x and y appear together in a linear constraint). For the case of separable bi-linear programming problems, the optimum value of the minimization problem is attained at an extreme point of the polytope, just as in the case of quasi-concave minimization problems. For such problems, it is possible to obtain an approximate solution which is also an extreme point of the polytope, using the algorithm given in Section 5.

4.3 Sum-of-Ratios Optimization

Consider the optimization of the following rational function over a polytope.

$$\begin{aligned} \min \quad & f(x) = \sum_{i=1}^{k} \frac{f_i(x)}{g_i(x)} \\ \text{s.t.} \quad & x \in P. \end{aligned} \tag{5}$$

Here, $f_1, \ldots, f_k$ and $g_1, \ldots, g_k$ are linear functions whose values are positive over the polytope P, and k is a fixed number. This problem does not fall into the framework given in Section 1 (the function combining $f_1, \ldots, f_k, g_1, \ldots, g_k$ does not necessarily satisfy Property 1). However, it is still possible to use our framework to find an approximate solution to this optimization problem. Let $h_i(x) = f_i(x)/g_i(x)$ for $i = 1, \ldots, k$. We first show that it is possible to construct an approximate Pareto-optimal front of the functions $h_i(x)$ in polynomial time.

Lemma 4.3 *It is possible to construct an approximate Pareto-optimal front $P_\epsilon(\pi)$ of the k functions $h_i(x) = f_i(x)/g_i(x)$ in time polynomial in $|\pi|$ and $1/\epsilon$, for all $\epsilon > 0$.*

Proof. From Theorem 2.3, it suffices to show that we can solve the gap problem corresponding to the k functions $h_i(x)$ in polynomial time. Solving the gap problem corresponds to checking the feasibility of the following system:

$$\begin{aligned} h_i(x) &\leq (1 - \epsilon')v_i, \quad \text{for } i = 1, \ldots, k, \\ x &\in P. \end{aligned}$$

Each constraint $h_i(x) \leq (1 - \epsilon')v_i$ is equivalent to $f_i(x) \leq (1 - \epsilon')v_i \cdot g_i(x)$, which is a linear constraint as $f_i(x)$ and $g_i(x)$ are linear functions. Hence solving the gap problem reduces to checking the feasibility of a linear program, which can be done in polynomial time under the assumption that we have a polynomial time separation oracle for the polytope P. $\square$

The corresponding versions of Lemma 3.1 and Lemma 3.2 for the sum-of-ratios minimization problem are given below.

Lemma 4.4 *There is at least one optimal solution x^* to* (5) *such that x^* is in $P(\pi)$, the Pareto-optimal front of the functions $h_1(x), \ldots, h_k(x)$.*

Proof. Suppose $\hat{x}$ is an optimal solution of the problem and $\hat{x} \notin P(\pi)$. Then there exists $x^* \in P(\pi)$ such that $h_i(x^*) \le h_i(\hat{x})$ for all $i = 1, \ldots, k$. Then $f(x^*) = \sum_{i=1}^k h_i(x^*) \le \sum_{i=1}^k h_i(\hat{x}) \le f(\hat{x})$. Thus x^* minimizes the function f and is in $P(\pi)$. $\square$

Lemma 4.5 *Let $\hat{x}$ be a solution in $P_\epsilon(\pi)$ that minimizes $f(x)$ over all points $x \in P_\epsilon(\pi)$. Then $\hat{x}$ is a $(1+\epsilon)$-approximate solution of the problem* (5).

Proof. Let x^* be an optimal solution of (5) that is in $P(\pi)$. By definition, there exists $x' \in P_\epsilon(\pi)$ such that $h_i(x') \le (1+\epsilon)h_i(x^*)$, for all $i = 1, \ldots, k$. Therefore,

$$f(x') = \sum_{i=1}^{k} h_i(x') \le \sum_{i=1}^{k} (1+\epsilon)h_i(x^*) \le (1+\epsilon)f(x^*).$$

Since $\hat{x}$ is a minimizer of $f(x)$ over all the solutions in $P_\epsilon(x)$, $f(\hat{x}) \le f(x') \le (1+\epsilon)f(x^*)$. $\square$

The existence of an FPTAS for problem (5) now follows from Lemma 4.4 and Lemma 4.5. We therefore have the following corollary.

Corollary 4.6 *Consider the problem given by* (5)*, and suppose that k is fixed. Then the problem admits an FPTAS if $f_i(x) > 0$, $g_i(x) > 0$ over the given polytope P.*

5 The Special Case of Minimizing Quasi-Concave Functions

The algorithm given in Section 3 may not necessarily return an extreme point of the polytope P as an approximate solution of the optimization problem given by (1). However, in certain cases it is desirable that the approximate solution we obtain is also an extreme point of the polytope. For example, suppose P describes the convex hull of all the feasible solutions of a combinatorial optimization problem, such as the spanning tree problem. Then an algorithm that returns an extreme point of P as an approximate solution can be used directly to get an approximate solution for the combinatorial optimization problem with a non-linear objective function as well. In this section, we demonstrate such an algorithm for the case when the objective function is a quasi-concave function, which we define below.

Definition 5.1 *A function $f : \mathbb{R}^n \to \mathbb{R}$ is quasi-concave if for all $\lambda \in \mathbb{R}$, the set $S_\lambda = \{x \in \mathbb{R}^n : f(x) \ge \lambda\}$ is convex.*

It is a well known result that the minimum of a quasi-concave function over a polytope is attained at an extreme point of the polytope (see e.g. Bertsekas, Nedić, and Ozdaglar (2003)). In fact, for this case, it is also possible to get an approximate solution of the problem which is an extreme point of the polytope, a result already given by Goyal and Ravi (2009). We can get a similar result using our framework, by employing a different algorithm that uses the

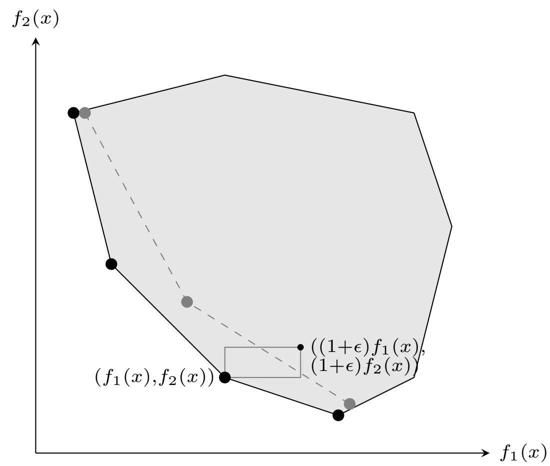

Figure 2: Figure illustrating the concept of convex Pareto-optimal front CP (shown by solid black points) and approximate convex Pareto-optimal front CP_ϵ (shown by solid gray points) for two objectives. The dashed lines represent the lower envelope of the convex hull of CP_ϵ

concept of approximate convex Pareto set, instead of approximate Pareto-optimal front. In the rest of this section, we only consider the case where the objective functions as well as the constraints are all linear. We first define a convex Pareto-optimal set below (Diakonikolas and Yannakakis 2008).

Definition 5.2 *Let π be an instance of a multi-objective minimization problem. The* convex Pareto-optimal set*, denoted by $CP(\pi)$, is the set of extreme points of $conv(P(\pi))$.*

Similar to the Pareto-optimal front, computing the convex Pareto-optimal front is an intractable problem in general. For example, determining whether a point belongs to the Pareto-optimal front for the two-objective shortest path problem is NP-hard (Hansen 1979). Also, the number of undominated solutions for the two-objective shortest path can be exponential in the input size of the problem. This means that $CP(\pi)$ can have exponentially many points, as the shortest path problem can be formulated as a min-cost flow problem, which has a linear programming formulation. Therefore, similar to the notion of an approximate Pareto-optimal front, we need to have a notion of an approximate convex Pareto-optimal front, defined below (Diakonikolas and Yannakakis 2008).

Definition 5.3 *Let π be an instance of a multi-objective minimization problem. For $\epsilon > 0$, an ϵ-*approximate convex Pareto-optimal set*, denoted by $CP_\epsilon(\pi)$, is a set of solutions, such that for all $x \in X$, there is $x' \in conv(CP_\epsilon(\pi))$ such that $f_i(x') \leq (1+\epsilon)f_i(x)$, for all i.*

The concept of convex Pareto-optimal set and approximate convex Pareto-optimal set is illustrated in Figure 2. In the rest of the paper, whenever we refer to an (approximate) convex Pareto-optimal front, we mutually refer to both its set of solutions and their vectors of objective function values.

Before giving an algorithm for computing a particular approximate convex Pareto-optimal set, we first give some intuition about the structure of the convex Pareto-optimal set. The Pareto-optimal front $P(\pi)$ corresponds to the solutions of the weighted linear program $\min \sum_{i=1}^{k} w_i f_i(x)$ over the polytope P, for all weight vectors $w \in \mathbb{R}^k_{\geq 0}$. The solution points in the convex Pareto-optimal set $CP(\pi)$ are the extreme point solutions of these linear programs. Thus one way to obtain a convex Pareto-optimal set would be to obtain the optimal extreme points of the weighted linear program for all non-negative weights w. The idea behind the algorithm for finding an approximate convex Pareto-optimal set $CP_\epsilon(\pi)$ is to choose a polynomial number of such weight vectors, and obtain the corresponding extreme point solutions for the weighted linear programs.

The algorithm for computing CP_ϵ is presented below. Without any loss of generality, for this section we assume that $m = 1/M$. For a positive integer N, let $[N]$ denote the set $\{1, \ldots, N\}$. In steps $2-3$, we compute the weight set $W(U)$, which is a union of k sets $W_j(U)$ for $j = 1, \ldots, k$. In each $W_j(U)$, the jth component is fixed at U, and the other components vary from 1 to U. In steps $4-7$ we compute the weight set $R(M)$, which again is a union of k sets $R_j(M)$ for $j = 1, \ldots, k$. In each $R_j(M)$, the jth component is fixed at 1, while the other components take values in the set $\{2^0, 2^1, \ldots, 2^{2\lceil \log_2 M \rceil}\}$. In steps $7-11$ of the algorithm, the k objective functions are combined together using the two weight sets, and CP_ϵ is then obtained by computing optimal extreme points for all such weighted objective functions over the polytope P.

1. $U \leftarrow \left\lceil \frac{2(k-1)}{\epsilon} \right\rceil$.
2. For $j = 1, \ldots, k$, $W_j(U) \leftarrow [U]^{j-1} \times \{U\} \times [U]^{k-j}$.
3. $W(U) \leftarrow \cup_{j=1}^{k} W_j(U)$.
4. $S(M) \leftarrow \{2^0, 2^1, \ldots, 2^{2\lceil log_2 M \rceil}\}$.
5. For $j = 1, \ldots, k$, $R_j(M) \leftarrow (S(M))^{j-1} \times \{1\} \times (S(M))^{k-j}$.
6. $R(M) \leftarrow \cup_{j=1}^{k} R_j(M)$.
7. $CP_\epsilon \leftarrow \emptyset$.
8. For each $r \in R(M)$ do
9. For each $w \in W(U)$ do
10. $q \leftarrow$ optimal basic feasible solution for $\{\min \sum_{i=1}^{k} r_i w_i (a_i^T x) : x \in P\}$.
11. $CP_\epsilon \leftarrow CP_\epsilon \cup \{q\}$.
12. Return CP_ϵ.

Theorem 5.4 (Diakonikolas and Yannakakis (2008)) *The above algorithm yields an approximate convex Pareto-optimal front CP_ϵ corresponding to the k linear functions $a_i^T x$, $i = 1, \ldots, k$, subject to the constraints $x \in P$.*

A sketch of the proof of this theorem is given in the appendix. For quasi-concave functions, it suffices to consider only the points in $CP_\epsilon(\pi)$ returned by this algorithm to solve the optimization problem given by (1). It should be noted that the following theorem holds specifically for $CP_\epsilon(\pi)$ computed using the above algorithm, and not for any arbitrary $CP_\epsilon(\pi)$.

Theorem 5.5 *Consider the optimization problem given by* (1). *If f is a quasi-concave function and satisfies Conditions 1-3 given in Section 3, then the set CP_ϵ obtained using the above algorithm contains a $(1+\epsilon)^c$-approximate solution to the optimization problem.*

Proof. The lower envelope of the convex hull of CP_ϵ is an approximate Pareto-optimal front. By Lemma 3.2, the approximate Pareto-optimal front contains a solution that is $(1+\epsilon)^c$-approximate. Therefore, to find an approximate solution of the optimization problem, it suffices to find a minimum of the function g over $conv(CP_\epsilon)$. Since f is a quasi-concave function, g is a quasi-concave function as well. Therefore, the minimum of g over $conv(CP_\epsilon)$ is attained at an extreme point of $conv(CP_\epsilon)$, which is in CP_ϵ. Since any point in CP_ϵ is an extreme point of the polytope P (as all the points in CP_ϵ are obtained by solving a linear program over the polytope P as given in the above algorithm), the theorem follows. $\square$

The overall running time of the algorithm is $O\big(k^2(\frac{(k-1)\log M}{\epsilon})^k \cdot LP(n, |\pi|)\big)$, where $LP(n, |\pi|)$ is the time taken to find an optimal extreme point of a linear program in n variables and $|\pi|$ bit-size input. We now discuss a couple of applications of this algorithm for combinatorial optimization problems.

5.1 Multiplicative Programming Problems in Combinatorial Optimization

Since the above algorithm always returns an extreme point as an approximate solution, we can use the algorithm to design approximation algorithms for combinatorial optimization problems where a complete description of the convex hull of the feasible set in terms of linear inequalities or a separation oracle is known. For example, consider the following optimization problem.

$$\begin{aligned} \min \quad & f(x) = f_1(x) \cdot f_2(x) \cdot \ldots \cdot f_k(x) \\ \text{s.t.} \quad & x \in X \subseteq \{0,1\}^n. \end{aligned} \tag{6}$$

Since the product of k linear functions is a quasi-concave function (Konno and Kuno 1992; Benson and Boger 1997), we can use the above algorithm to get an approximate solution of this problem by minimizing the product function over the polytope $P = conv(X)$. The FPTAS always returns an extreme point of P as an approximate solution, which is guaranteed to be integral. We therefore have the following theorem.

Theorem 5.6 *Consider the optimization problem given by* (6)*, and assume that a complete description of $P = conv(X)$ (or the dominant of P) is known in terms of linear inequalities or a polynomial time separation oracle. Then if k is fixed, the problem admits an FPTAS.*

Our FPTAS is both simple in description as well as easily generalizable to the case where we have more than two terms in the product, in contrast to the existing results in the literature (Kern and Woeginger 2007; Goyal, Genc-Kaya, and Ravi 2011; Goyal and Ravi 2009).

5.2 Mean-risk Minimization in Combinatorial Optimization

Another category of problems for which this framework is applicable is mean-risk minimization problems that arise in stochastic combinatorial optimization (Atamtürk and Narayanan 2008; Nikolova 2010). Let $f(x) = c^T x$, $c \in \mathbb{R}^n$ be the objective function of a combinatorial optimization problem, where as usual $x \in X \subseteq \{0,1\}^n$. Suppose that the coefficients c are mutually independent random variables. Let the vector $\mu \in \mathbb{R}^n_+$ denote the mean of the random variables, and $\tau \in \mathbb{R}^n_+$ the vector of variance of the random variables. For a given solution vector x, the average cost of the solution is $\mu^T x$ and the variance is $\tau^T x$. One way to achieve a trade-off between the mean and the variance of the solution is to consider the following optimization problem.

$$\begin{aligned} \min \quad & f(x) = \mu^T x + c\sqrt{\tau^T x} \\ \text{s.t.} \quad & x \in X \subseteq \{0,1\}^n. \end{aligned} \tag{7}$$

Here, $c \geq 0$ is a parameter that captures the trade-off between the mean and the variance of the solution. In this case, $f(x)$ is a concave function of rank two. If we have a concise description of $P = conv(X)$, then we can use the above algorithm to get an FPTAS for the problem. This is captured in the following theorem.

Theorem 5.7 *Consider the optimization problem given by* (7)*, and assume that a complete description of* $P = conv(X)$ *(or the dominant of* P*) is known in terms of linear inequalities or a polynomial time separation oracle. Then the problem admits an FPTAS.*

Again, although an FPTAS for this problem is known (Nikolova 2010), our FPTAS has the advantage of being conceptually simpler than the existing methods.

6 Inapproximability of Minimizing a Concave Function over a Polytope

In this section, we show that it is not possible to approximate the minimum of a concave function over a unit hypercube to within any factor, unless P = NP. First, we establish the inapproximability of supermodular function minimization.

Definition 6.1 *Given a finite set* S*, a function* $f : 2^S \to \mathbb{R}$ *is said to be* supermodular *if it satisfies the following condition:*

$$f(X \cup Y) + f(X \cap Y) \geq f(X) + f(Y), \quad \textit{for all } X, Y \subseteq S.$$

Definition 6.2 *A set function* $f : 2^S \to \mathbb{R}$ *is* submodular *if* $-f$ *is supermodular.*

In some sense, supermodularity is the discrete analog of concavity, which is illustrated by the continuous extension of a set function given by Lovász (1983). Suppose f is a set function defined on the subsets of S, where $|S| = n$. Then the continuous extension $\hat{f} : \mathbb{R}^n_+ \to \mathbb{R}$ of f is given as follows:

1. $\hat{f}(x) = f(X)$, where x is the $0/1$ incidence vector of $X \subseteq S$.

2. For any other x, there exists a unique representation of x of the form $x = \sum_{i=1}^{k} \lambda_i a_i$, where $\lambda_i > 0$, and a_i are $0/1$ vectors satisfying $a_1 \leq a_2 \leq \ldots \leq a_k$. Then $\hat{f}(x)$ is given by $\hat{f}(x) = \sum_{i=1}^{k} \lambda_i f(A_i)$, where a_i is the incidence vector of $A_i \subseteq S$.

The following theorem establishes a direct connection between f and $\hat{f}$.

Theorem 6.3 (Lovász (1983)) *f is a supermodular (resp. submodular) function if and only if its continuous extension $\hat{f}$ is concave (resp. convex).*

We first give a hardness result for supermodular function minimization.

Theorem 6.4 *Let $f : 2^S \to \mathbb{Z}_+$ be a supermodular function defined over the subsets of S. Then it is not possible to approximate the minimum of f to within any factor, unless P = NP.*

Proof. The proof is by reduction from the E4-Set splitting problem (Håstad 2001). The E4-Set splitting problem is this: given a ground set V, and a collection C of subsets $S_i \subset V$ of size exactly 4, find a partition V_1 and V_2 of V so as to maximize the number of subsets S_i such that both $S_i \cap V_1$ and $S_i \cap V_2$ are non-empty. Let $g : 2^V \to \mathbb{Z}$ be the function such that $g(V')$ is equal to the number of subsets S_i satisfying $V' \cap S_i \neq \emptyset$ and $(V \setminus V') \cap S_i \neq \emptyset$. Then g is a submodular function (g is just the extension of the cut function to hypergraphs), and therefore the function f defined by $f(V') = |C| - g(V') + \epsilon$ is supermodular, where $\epsilon > 0$. Clearly, f is a positive valued function.

Håstad (2001) has shown that it is NP-hard to distinguish between the following two instances of E4-Set splitting:

1. There is a set V' which splits all the subsets S_i, and

2. No subset of V splits more than a fraction $(7/8 + \eta)$ of the sets S_i, for any $\eta > 0$.

For the first case, the minimum value of f is ϵ, whereas for the second case, the minimum is at least $(\frac{1}{8} - \eta)|C|$. Therefore, if we had an α-approximation algorithm for supermodular function minimization, the algorithm would return a set for the first case with value at most $\epsilon\alpha$. Since ϵ is arbitrary, we can always choose ϵ so that $\epsilon\alpha < (\frac{1}{8} - \eta)|C|$, and hence it will be possible to distinguish between the two instances. We get a contradiction, therefore the hardness result follows. □

Using this result, we now establish the hardness of minimizing a concave function over a $0/1$ polytope.

Theorem 6.5 *It is not possible to approximate the minimum of a positive valued concave function f over a polytope to within any factor, even if the polytope is the unit hypercube, unless P = NP.*

Proof. Kelner and Nikolova (2007) have given an approximation preserving reduction from minimization of a supermodular function f to minimization of its continuous extension $\hat{f}$ over the $0/1$-hypercube. Thus any γ-approximation algorithm for the latter will imply a γ-approximation algorithm for the former as well. This implies that minimizing a positive valued concave function over a $0/1$-polytope cannot be approximated to within any factor, unless P = NP. $\square$

In fact, a similar hardness of approximation result can be obtained for minimizing a concave quadratic function of rank 2 over a polytope. Pardalos and Vavasis (1991) show the NP-hardness of minimizing a rank 2 concave quadratic function over a polytope by reducing the independent set problem to the concave quadratic minimization problem. In their reduction, if a graph has an independent set of a given size k, then the minimum value of the quadratic function is 0, otherwise the minimum value is a large positive number. This gives the same hardness of approximation result for minimizing a rank 2 quadratic concave function over a polytope.

The two inapproximability results show that in order to get an FPTAS for minimizing a non-convex function over a polytope, we need not only the low-rank property of the objective function, but also additional conditions, such as Property 1 of the function g given in Section 3.1.

Acknowledgements

The authors thank Vineet Goyal and R. Ravi for useful discussions, and for sharing an unpublished draft (Goyal and Ravi 2009) on a similar topic. The authors also thank Ilias Diakonikolas for sharing a detailed version of his paper (Diakonikolas and Yannakakis 2008).

References

Al-Khayyal, F. A. and J. E. Falk (1983). Jointly constrained biconvex programming. *Mathematics of Operations Research 8*, 273–286.

Atamtürk, A. and V. Narayanan (2008). Polymatroids and mean0risk minimization in discrete optimization. *Operations Research Letters 36*, 618–622.

Benson, H. P. and G. M. Boger (1997). Multiplicative programming problems: Analysis and efficient point search heuristic. *Journal of Optimization Theory and Applications 94*, 487–510.

Bertsekas, D., A. Nedić, and A. Ozdaglar (2003). *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific.

Diakonikolas, I. and M. Yannakakis (2008). Succinct approximate convex pareto curves. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, CA, pp. 74–83.

Eisenberg, E. (1961). Aggregation of utility functions. *Management Science*, 337–350.

Falk, J. E. and S. W. Palocsay (1992). Optimizing the sum of linear fractional functions. In C. A. Floudas and

P. M. Pardalos (Eds.), *Recent Advances in Global Optimization*, pp. 228–258. Dordrecht: Kluwer Academic Publishers.

Feige, U., V. S. Mirrokni, and J. Vondrák (2007). Maximizing non-monotone submodular functions. In *48th Annual IEEE Symposium on Foundations of Computer Science*, Providence, RI, pp. 461–471.

Goyal, V., L. Genc-Kaya, and R. Ravi (2011). An FPTAS for minimizing the product of two non-negative linear cost functions. *Mathematical Programming 126*, 401–405.

Goyal, V. and R. Ravi (2009). An FPTAS for minimizing a class of quasi-concave functions over a convex domain. Technical report, Tepper School of Business, Carnegie Mellon University.

Grötschel, M., L. Lovász, and A. Schrijver (1988). *Geometric Algorithms and Combinatorial Optimization*. Berlin: Springer.

Hansen, P. (1979). Bicriterion path problems. *Proceedings of the 3rd Conference on Multiple Criteria Decision Making Theory and Application*, 109–127.

Håstad, J. (2001). Some optimal inapproximability results. *Journal of the ACM 48*, 798–859.

Horst, R. and P. M. Pardalos (Eds.) (1995). *Handbook of Global Optimization*, Volume 1. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Kelner, J. A. and E. Nikolova (2007). On the hardness and smoothed complexity of quasi-concave minimization. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, Providence, RI, pp. 472–482.

Kern, W. and G. J. Woeginger (2007). Quadratic programming and combinatorial minimum weight product problems. *Mathematical Programming 110*, 641–649.

Konno, H. (1976). A cutting plane algorithm for solving bilinear programs. *Mathematical Programming 11*, 14–27.

Konno, H., C. Gao, and I. Saitoh (1998). Cutting plane/tabu search algorithms for low rank concave quadratic programming problems. *Journal of Global Optimization 13*, 225–240.

Konno, H. and T. Kuno (1992). Linear multiplicative programming. *Mathematical Programming 56*, 51–64.

Konno, H., T. Thach, and H. Tuy (1996). *Optimization on Low Rank Nonconvex Structures*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Kuno, T. (1999). Polynomial algorithms for a class of minimum rank-two cost path problems. *Journal of Global Optimization 15*, 405–417.

Lovász, L. (1983). Submodular functions and convexity. In A. Bachem, M. Grötschel, and B. Korte (Eds.), *Mathematical Programming - The State of the Art*, pp. 235–257. Springer-Verlag.

Matsui, T. (1996). NP-hardness of linear multiplicative programming and related problems. *Journal of Global Optimization 9*, 113–119.

Mittal, S. and A. S. Schulz (2008). A general framework for designing approximation schemes for combinatorial optimization problems with many objectives combined into one. In *APPROX-RANDOM*, Volume 5171 of *Lecture Notes in Computer Science*, Cambridge, MA, pp. 179–192.

Nesterov, Y. and A. Nemirovskii (1961). *Interior Point Polynomial Methods in Convex Programming*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Nikolova, E. (2010). Approximation algorithms for reliable stochastic combinatorial optimization. In *Proceedings of the APPROX-RANDOM*, Volume 6302 of *Lecture Notes in Computer Science*, Barcelona, Spain, pp. 338–351.

Papadimitriou, C. H. and M. Yannakakis (2000). On the approximability of trade-offs and optimal access of web sources. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, Redondo Beach, CA, pp. 86–92.

Pardalos, P. M. and S. A. Vavasis (1991). Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization 1*, 15–22.

Porembski, M. (2004). Cutting planes for low-rank like concave minimization problems. *Operations Research 52*, 942–953.

Safer, H. M. and J. B. Orlin (1995a). Fast approximation schemes for multi-criteria combinatorial optimization. Technical report, Operations Research Center, Massachusetts Institute of Technology.

Safer, H. M. and J. B. Orlin (1995b). Fast approximation schemes for multi-criteria flow, knapsack, and scheduling problems. Technical report, Operations Research Center, Massachusetts Institute of Technology.

Safer, H. M., J. B. Orlin, and M. Dror (2004). Fully polynomial approximation in multi-criteria combinatorial. Technical report, Operations Research Center, Massachusetts Institute of Technology.

Schaible, S. (1977). A note on the sum of a linear and linear-fractional function. *Naval Research Logistics 24*, 691–693.

Schaible, S. and J. Shi (2003). Fractional programming: The sum-of-ratios case. *Optimization Methods and Software 18*, 219–229.

Sherali, H. D. and A. Alameddine (1992). A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization 2*, 379–410.

Vavasis, S. A. (1992). Approximation algorithm for indefinite quadratic programming. *Mathematical Programming 57*, 279–311.

A Appendix

Proof of Theorem 5.4: Let us call a positive valued vector $(v_1, \ldots, v_k)$ α-balanced if for any $i, j \in \{1, \ldots, k\}$, $v_i/v_j \leq \alpha$. A solution x is U-enabled, if it is the optimal solution of the linear program for the objective $\min \sum_{i=1}^k w_i a_i^T x$ over the polytope P, where $w \in W(U)$ (Recall from Section 5 that $W(U) = \cup_{j=1}^k W_j(U)$, where $W_j(U) = [U]^{j-1} \times \{U\} \times [U]^{k-j}$). Let all the U-enabled solutions be $q^1, \ldots, q^l$, where l is the number of all such solutions.

Lemma A.1 (Diakonikolas and Yannakakis (2008)) *Let $\epsilon > 0$. Suppose that s is on the Pareto-optimal front of the k objectives $a_1^T x, \ldots, a_k^T x$ and is 2-balanced, but not U-enabled. Then there is a convex combination of U-enabled solutions, say s', such that $s'_i \leq (1+\epsilon)s_i$ for $i = 1, \ldots, k$.*

Proof. Suppose there is no convex combination of the U-enabled solutions that is within a factor of $1+\epsilon$ from s in all the components. This implies that the following linear program is infeasible.

$$\sum_{j=1}^{l} \lambda_j q^j \leq (1+\epsilon)s,$$
$$\sum_{j=1}^{l} \lambda_j = 1,$$
$$\lambda_1, \ldots, \lambda_l \geq 0.$$

By Farkas' lemma, there exist $w_1, \ldots, w_k$ and v which satisfy the following inequalities.

$$w \cdot q^j + v \geq 0, \quad j = 1, \ldots, l,$$
$$(1+\epsilon) w \cdot s + v < 0,$$
$$w \in \mathbb{R}_+^k.$$

This can be simplified to the following set of inequalities.

$$w \cdot q^j > (1+\epsilon) w \cdot s \quad \text{for all } j = 1, \ldots, l,$$
$$w \in \mathbb{R}_+^k.$$

Thus, in order to obtain a contradiction to our assumption that there is no convex combination of the U-enabled solutions that is within a factor $1+\epsilon$ from s in all the components, it will be sufficient to show that for *any* $w \in \mathbb{R}_+^k$, there is a j such that $w \cdot q^j \leq (1+\epsilon) w \cdot s$, which is what we will do in the rest of this proof.

Let $w \in \mathbb{R}_+^k$ be an arbitrary weight vector. Without loss of generality, we can assume that the maximum value of a component of vector w is U (this can be achieved by suitably scaling the components of w). Let w^* be the weight vector given by $w_i^* = \lceil w_i \rceil$ for $i = 1, \ldots, k$. Clearly, $w^* \in W(U)$. Let q^* be the optimal solution for the objective $\min \sum_{i=1}^k w_i^* a_i^T x$ over the polytope P, then q^* is U-enabled. We will show that $w \cdot q^* \leq (1+\epsilon) w \cdot s$, thus achieving the desired contradiction.

Let t be such that $w^*_t = U$. By our choice of w^*, each component of $w^* - w$ is at most 1. Therefore,

$$(w^* - w) \cdot s \leq \sum_{i \in [k] \setminus \{t\}} s_i \leq 2(k-1)s_t \leq \epsilon U s_t \leq \epsilon(w \cdot s),$$

where the second inequality follows from the fact that s is 2-balanced, the third inequality follows from our choice of $U = \lceil 2(k-1)/\epsilon \rceil$, and the last inequality follows from the fact that $s_t \leq \frac{1}{U}(w \cdot s)$ (as each component of w is at most U, by assumption). Therefore, from this chain of inequalities, we get

$$w^* \cdot s \leq (1+\epsilon) w \cdot s.$$

Also, q^* is the optimal solution for the objective $\min \sum_{i=1}^{k} w^*_i a_i^T x$, therefore

$$w^* \cdot q^* \leq w^* \cdot s.$$

Therefore, we get

$$w \cdot q^* \leq w^* \cdot q^* \leq w^* \cdot s \leq (1+\epsilon) w \cdot s.$$

This establishes the desired contradiction, and completes the proof of the lemma. □

Using the above lemma, we can now prove the theorem. Consider any Pareto-optimal solution $s = (s_1, \ldots, s_k)$. The maximum ratio between any two components of s is at most M^2. Therefore, for some $r \in R(M)$, all the components in the vector $(r_1 s_1, \ldots, r_k s_k)$ are within a factor of 2 of each other. Note that $(r_1 s_1, \ldots, r_k s_k)$ is on the Pareto-optimal front of the weighted k objectives $r_1 a_1^T x, \ldots, r_k a_k^T x$. The algorithm of Section 5 computes U-enabled solutions for these weighted k objectives for all $r \in R(M)$. The above lemma implies that there is a convex combination of the U-enabled solutions for the weighted objective functions, say s' such that $r_i s'_i \leq (1+\epsilon) r_i s_i$, for $i = 1, \ldots, k$. Equivalently, $s'_i \leq (1+\epsilon) s_i$, implying that the solution s is indeed approximately dominated by some convex combination of the solutions returned by the algorithm. □