Approximation Algorithms for Concave Quadratic Programming

Stephen Vavasis*

TR 90-1172
December 1990

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*Supported by the Applied Mathematical Sciences Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under grant DE-FG02-86ER25015.A000 and in part by the National Science Foundation, the Air Force office of Scientific Research, and the Office of Naval Research, through NSF grant DMS-8920550.

Approximation algorithms for concave quadratic programming

Stephen A. Vavasis*
Department of Computer Science
Cornell University
Ithaca, NY 14853

November 28, 1990

Abstract

We consider ϵ -approximation schemes for concave quadratic programming. Because the existing definition of ϵ -approximation for combinatorial optimization problems is inappropriate for nonlinear optimization, we propose a new definition for ϵ -approximation. We argue that such an approximation can be found in polynomial time for fixed ϵ and k , where k denotes the number of negative eigenvalues. Our algorithm is polynomial in $1/\epsilon$ for fixed k , and superexponential in k for fixed ϵ .

1 Concave quadratic programming

Quadratic programming is a nonlinear optimization problem of the following form:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{x}^T H \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{subject to} & W \mathbf{x} \geq \mathbf{b}. \end{array} \tag{1}$$

*Supported by the Applied Mathematical Sciences Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under grant DE-FG02-86ER25013.A000 and in part by the National Science Foundation, the Air Force office of Scientific Research, and the Office of Naval Research, through NSF grant DMS 8920550.

In this formulation, $\mathbf{x}$ is the n -vector of unknowns. The remaining variables stand for data in the problem instance: H is an $n \times n$ symmetric matrix, $\mathbf{c}$ is an n -vector, W is an $m \times n$ matrix, and $\mathbf{b}$ is an m -vector. The relation ' $\geq$ ' in the constraint $W\mathbf{x} \geq \mathbf{b}$ is the usual componentwise inequality.

Quadratic programming, a generalization of linear programming, has applications in economics, planning, and many kinds of engineering design. In addition, more complicated kinds of nonlinear programming problems are often simplified into quadratic programming problems.

No efficient algorithm is known to solve the general case of (1). The lack of an efficient algorithm is not surprising, since quadratic programming is known to be NP-hard, a result due to Sahni [1974]. An NP-hardness proof appears in this paper (it follows from the discussion in Section 3). More recently, Vavasis [1990] showed that the decision version of the problem lies in NP, and hence is NP-complete.

Many avenues for addressing (1) have been pursued in the literature. For example, efficient algorithms are known for the special case in which H is positive semidefinite, known as the *convex* case. See Kozlov, Tarasov and Haćijan [1979] for the first polynomial-time algorithm for the convex case. See Kapoor and Vaidya [1986] or Ye and Tse [1989] for efficient interior point algorithms for this problem. Active set methods (see Gill, Murray and Wright [1981]), a commonly-used class of methods for (1), are a combination of local search and heuristics.

A very successful way to address NP-hard combinatorial optimization problems has been approximation algorithms. In light of the importance of quadratic programming, it is surprising the ϵ -approximation algorithms have so far not been pursued. In this report, we investigate ϵ -approximation algorithms for the concave case, i.e., the case that H is negative semidefinite. Our techniques at present are not able to address the more general indefinite case (in which H has a mixture of positive and negative eigenvalues); see Section 4 for further discussion of the indefinite case. The concave case, like the general case, is NP-hard.

First it is necessary to give a definition of ϵ -approximation. To our knowledge, this concept has not been previously defined for nonlinear optimization in the literature (but see below), so it is necessary to come up with our own definition. We have the following proposal:

Definition 1 *Consider an instance of quadratic programming written in the*

form (1). Let $f(\mathbf{x})$ denote the objective function $\frac{1}{2}\mathbf{x}^T H \mathbf{x} + \mathbf{c}^T \mathbf{x}$. Let $\mathbf{x}^*$ be an optimum point of the problem. We say that $\mathbf{x}^\circ$ is an ϵ -approximate solution if there exists another feasible point $\mathbf{x}^\#$ such that

$$f(\mathbf{x}^\circ) - f(\mathbf{x}^*) \leq \epsilon[f(\mathbf{x}^\#) - f(\mathbf{x}^*)].$$

Notice that we may as well take $\mathbf{x}^\#$ in Definition 1 to be the point where the objective function is maximized. Thus, another way to interpret this definition is as follows. Let Π denote the feasible region, and let interval $[a, b]$ be $f(\Pi)$. Then $f(\mathbf{x}^\circ)$ should lie in the interval $[a, a + \epsilon(b - a)]$.

In the case that the objective function has no upper bound on the feasible region, the maximizer is of course undefined. Our definition loses its value in this situation because any feasible point is an ϵ -approximation for any $\epsilon > 0$.

The definition fails in the case that the objective function has no lower bound or in the case that no feasible points exist. In these cases, (1) is said to be *unbounded* or *infeasible* respectively. We should expect our approximation algorithm to return an indicator that the problem is unbounded or infeasible.

Finally, observe that any feasible point is a 1-approximation by this definition, and only the optimum is a 0-approximation. Thus, the definition makes sense only for ϵ in the interval $(0, 1)$.

This definition has some useful properties. First, it is insensitive to translations or dilations of the objective function. In other words, if the objective function $f(\mathbf{x})$ is replaced by a new function $g(\mathbf{x}) = af(\mathbf{x}) + b$ where $a > 0$, a vector $\mathbf{x}^\circ$ that was previously an ϵ -approximation will continue to have that property.

A second useful property is that ϵ -approximation is preserved under transformations of the feasible region. The most general kind of transformation that preserves the format of (1) is an affine linear transformation. Let $\theta : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be an affine linear transformation, i.e., a function of the form $\theta(\mathbf{x}) = V(\mathbf{x} - \mathbf{x}_0)$ where V is an invertible $n \times n$ matrix. Then the problem of minimizing $f(\theta(\mathbf{y}))$ subject to $W\theta(\mathbf{y}) \geq \mathbf{b}$ is still a quadratic program, and, moreover, it is not hard to see that if $\mathbf{x}^\circ$ is an ϵ -approximate solution of the original problem, then $\theta^{-1}(\mathbf{x}^\circ)$ is an ϵ -approximate solution to the transformed problem.

We now state the main theorem of this paper.

Theorem 2 *Consider the concave case of (1), i.e., the case that H is negative semidefinite. Let k denote the rank of H . There is an algorithm to find ϵ -approximate solution to (1) in time*

$$O \left(k! \left\lceil \sqrt{\frac{2k}{\epsilon}} \right\rceil^k \ell \right)$$

steps. In this formula, ℓ denotes the time to solve a linear programming problem of the same size as (1).

We remark that ℓ grows polynomially with the size of the input. This fact was first proved by Hačijan [1979]. The best known asymptotic bound for ℓ is due to Vaidya [1989].

The algorithm we propose is similar to algorithms that have appeared in the literature. In particular, it is very reminiscent of algorithms described in Pardalos and Rosen [1987]. Our contribution is to define a formal meaning for approximation algorithm, and then show that the algorithm achieves this bound.

The remainder of this paper is organized as follows. In Section 2 we provide the algorithm and prove the main theorem. In Section 3 we indicate why polynomial dependence on $1/\epsilon$ and exponential dependence on k is expected. In Section 4 we discuss open questions raised by this work.

The definition of approximation proposed for combinatorial optimization differs from our definition and is usually stated as follows. A feasible point $\mathbf{x}^\circ$ is an ϵ -approximation if

$$|f(\mathbf{x}^\circ) - f(\mathbf{x}^*)| \leq \epsilon \cdot f(\mathbf{x}^*).$$

See, for example, Papadimitriou and Steiglitz [1982] for an extensive discussion of approximation for combinatorial optimization. This definition does not work for nonlinear optimization because it is not preserved when a constant is added to the objective function. In particular, the definition becomes useless in the case that $f(\mathbf{x}^*) = 0$.

Work by Katoh and Ibaraki [1987] addresses the problem of approximation algorithms for certain kinds of quasiconcave optimization problem. Their work generalizes the kinds of allowable objective functions (i.e., they

do not restrict attention to quadratic programming), but they make a number of restrictions concerning the feasible region and range of the objective function. Their results do not seem to be directly comparable to ours.

2 Proof of the main theorem.

Our starting point is a problem of the form (1) in which H is negative semidefinite with rank k . We can perform change of basis to diagonalize H , either via an eigenvalue computation or an LDL^T factorization of H (see Golub and Van Loan [1989]). In either case we end up with a problem of the form

$$\begin{aligned} & \text{minimize} && \frac{1}{2}\mathbf{y}^T D \mathbf{y} + \mathbf{c}^T \mathbf{y} + \mathbf{f}^T \mathbf{z} \\ & \text{subject to} && A\mathbf{y} + B\mathbf{z} \geq \mathbf{b} \end{aligned} \tag{2}$$

where, in this new formulation, D is a $k \times k$ negative definite diagonal matrix, $\mathbf{y}$ is a k -vector of unknowns, and $\mathbf{z}$ is an $n - k$ -vector of unknowns. Thus, the negative-definite part of the problem is confined to k variables.

Now we define the function

$$\phi(\mathbf{y}) = \min\{\mathbf{f}^T \mathbf{z} : \mathbf{z} \in \mathbb{R}^{n-k}, A\mathbf{y} + B\mathbf{z} \geq \mathbf{b}\}.$$

For a fixed choice of $\mathbf{y}$, if the system $A\mathbf{y} + B\mathbf{z} \geq \mathbf{b}$ has no feasible choice for $\mathbf{z}$, then we adopt the convention that $\phi(\mathbf{y}) = \infty$. Similarly, in the case that $\mathbf{f}^T \mathbf{z}$ has no lower bound, we say that $\phi(\mathbf{y}) = -\infty$.

Thus, the original problem can now be expressed simply as minimizing

$$q(\mathbf{y}) + \phi(\mathbf{y}) \tag{3}$$

where

$$q(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T D \mathbf{y} + \mathbf{c}^T \mathbf{y}.$$

In this formulation there are no constraints on $\mathbf{y}$. This is equivalent to (2). In particular, for any $\mathbf{y}_1$ feasible for (2), $q(\mathbf{y}_1) + \phi(\mathbf{y}_1)$ will be the value of the minimum possible objective function value among all feasible vectors of the form $(\mathbf{y}_1, \mathbf{z})$.

Lemma 3 *The set $C = \{\mathbf{y} \in \mathbb{R}^k : \phi(\mathbf{y}) < \infty\}$ is convex, and on this domain, ϕ is a convex function.*

PROOF. First, notice that the set C in the lemma can be written:

$$C = \{\mathbf{y} \in \mathbb{R}^k : \text{There exists } \mathbf{z} \in \mathbb{R}^{n-k} \text{ such that } A\mathbf{y} + B\mathbf{z} \geq \mathbf{b}\}.$$

This domain is convex, as the following argument shows. If $\mathbf{y}', \mathbf{y}'' \in C$, then there are $\mathbf{z}', \mathbf{z}''$ such that $A\mathbf{y}' + B\mathbf{z}' \geq \mathbf{b}$ and $A\mathbf{y}'' + B\mathbf{z}'' \geq \mathbf{b}$. This means that for all $\lambda \in [0, 1]$,

$$\begin{aligned} A((1-\lambda)\mathbf{y}' + \lambda\mathbf{y}'') + B((1-\lambda)\mathbf{z}' + \lambda\mathbf{z}'') &= (1-\lambda)(A\mathbf{y}' + B\mathbf{z}') \\ &\quad + \lambda(A\mathbf{y}'' + B\mathbf{z}'') \\ &\geq (1-\lambda)\mathbf{b} + \lambda\mathbf{b} \\ &\geq \mathbf{b}. \end{aligned}$$

This shows that $(1-\lambda)\mathbf{y}' + \lambda\mathbf{y}'' \in C$.

To show that ϕ is convex, consider $\phi(\mathbf{y}')$ and $\phi(\mathbf{y}'')$ for $\mathbf{y}', \mathbf{y}'' \in C$. We know by assumption that $\phi(\mathbf{y}'), \phi(\mathbf{y}'') < \infty$. We also assume (see the lemma below) that $\phi(\mathbf{y}'), \phi(\mathbf{y}'') > -\infty$. Then the problem of computing $\phi(\mathbf{y}')$ is equivalent to solving a linear program, and we know from linear programming theory that the minimum is achieved, say at a vector $\mathbf{z}'$. In other words, there exists a $\mathbf{z}'$ such that $\phi(\mathbf{y}') = \mathbf{f}^T \mathbf{z}'$ and $A\mathbf{y}' + B\mathbf{z}' \geq \mathbf{b}$. The same holds for $\mathbf{y}''$, i.e., there is a $\mathbf{z}''$ such that $\phi(\mathbf{y}'') = \mathbf{f}^T \mathbf{z}''$ and $A\mathbf{y}'' + B\mathbf{z}'' \geq \mathbf{b}$. Then we conclude as above that

$$A((1-\lambda)\mathbf{y}' + \lambda\mathbf{y}'') + B((1-\lambda)\mathbf{z}' + \lambda\mathbf{z}'') \geq \mathbf{b}$$

so that

$$\phi((1-\lambda)\mathbf{y}' + \lambda\mathbf{y}'') \leq \mathbf{f}^T((1-\lambda)\mathbf{z}' + \lambda\mathbf{z}'')$$

because ϕ is defined to be the minimum of expressions such as the one on the right-hand side. This last inequality is the same as

$$\phi((1-\lambda)\mathbf{y}' + \lambda\mathbf{y}'') \leq (1-\lambda)\phi(\mathbf{y}') + \lambda\phi(\mathbf{y}'').$$

■

We now show how to determine whether the original problem is unbounded.

Lemma 4 *Problem (2) is unbounded if and only if*

1. For every $\mathbf{y} \in C$, $\phi(\mathbf{y}) = -\infty$, or
2. Region C is unbounded.

PROOF. First, we prove that either of the two conditions imply that (2) is unbounded. The first condition trivially implies that (2) is unbounded. Indeed, the existence of any $\mathbf{y}$ such that $\phi(\mathbf{y}) = -\infty$ means that (2) is unbounded, as we see from examining (3).

Suppose for the other case that C is unbounded. Then it must contain a ray (because it is convex). Let us write the ray in the form $\{\mathbf{y}_1 + t\mathbf{y}_2 : t \geq 0\}$ where $\mathbf{y}_2$ is a nonzero vector. For large enough t , the function $\phi(\mathbf{y}_1 + t\mathbf{y}_2)$ must be linear in t . This statement is proved by noting that the solution to the linear program

$$\begin{array}{ll} \text{minimize} & \mathbf{f}^T \mathbf{z} \\ \text{subject to} & B\mathbf{z} \leq \mathbf{b} - A\mathbf{y} \end{array}$$

implicit in the definition of ϕ may be taken to be a basic feasible solution, and for large enough t we can assume that the choice of basis columns is independent of t .

Examining (3), we see that ϕ behaves linearly along the ray for large enough t , but the term $\frac{1}{2}(\mathbf{y}_1 + t\mathbf{y}_2)^T D(\mathbf{y}_1 + t\mathbf{y}_2)$ is quadratic in t with a negative leading term. Thus, the quadratic term dominates for t large enough and the objective function tends to $-\infty$ along the ray.

Now we prove that if the original problem is unbounded, at least one of the two conditions holds. Let $\{(\mathbf{y}_1, \mathbf{z}_1) + t(\mathbf{y}_2, \mathbf{z}_2) : t \geq 0\}$ be a ray along which the objective function of (2) tends to $-\infty$. Either $\mathbf{y}_2$ or $\mathbf{z}_2$ must be nonzero. Suppose $\mathbf{y}_2$ is nonzero; this implies that C contains the ray $\mathbf{y}_1 + t\mathbf{y}_2$ and hence is unbounded.

The other case is that $\mathbf{y}_2 = \mathbf{0}$, in which case $\mathbf{z}_2 \neq \mathbf{0}$. The ray can be written $(\mathbf{y}_1, \mathbf{z}_1 + t\mathbf{z}_2)$. We can see that $B\mathbf{z}_2 \geq \mathbf{0}$ (otherwise the ray would become infeasible for t large enough). Also, $\mathbf{f}^T \mathbf{z}_2 < 0$ because the objective function decreases along the ray by assumption. We claim that the existence of this $\mathbf{z}_2$ implies that $\phi(\mathbf{y}) = -\infty$ for any $\mathbf{y} \in C$. Let $\mathbf{z}$ be the vector such that $A\mathbf{y} + B\mathbf{z} \geq \mathbf{b}$; then $A\mathbf{y} + B(\mathbf{z} + t\mathbf{z}_2) \geq \mathbf{b}$ for any $t > 0$, and $\mathbf{f}^T(\mathbf{z} + t\mathbf{z}_2)$ is arbitrarily negative. ■

Now we begin to describe our algorithm. The first task of our algorithm is to determine the smallest possible rectangle R such that $C \subset R$. The term *rectangle* refers to the k -fold product of compact closed intervals. Let us write

$$R = [l_1, u_1] \times \cdots \times [l_k, u_k].$$

We now explain how l_1 is determined (the other $2k - 1$ interval bounds are determined in a similar fashion). We define

$$l_1 = \min\{y_1 : \text{There exists } y_1, y_2, \dots, y_k, \mathbf{z} \text{ such that } A\mathbf{y} + B\mathbf{z} \geq \mathbf{b}\}.$$

Thus, we see that l_1 is found by solving a linear program. From the previous lemma, we see that if this linear program is unbounded, then the original problem is unbounded.

In this manner we can determine the remaining l_i 's and u_i 's. We assume that $l_i < u_i$ for all i , because otherwise y_i is uniquely determined and may be dropped from the problem.

We now assume that the coordinates of the vector $\mathbf{y}$ are translated and rescaled so that the rectangle R is the k -dimensional unit cube. This simplifies the notation in the upcoming paragraphs. Note that this rescaling does not change the format of (2). Let us state this formally:

Assumption. *Without loss of generality, we assume that $R = [0, 1]^k$.*

For the next step of our algorithm we choose an integer m . This integer is on the order of $\sqrt{k/\epsilon}$; the exact formula for m is given below. We place an $(m + 1) \times \cdots \times (m + 1)$ grid of points in the cube R . The coordinates of the points have the form:

$$(i_1/m, i_2/m, \dots, i_k/m)$$

where $i_1, \dots, i_k$ are integers satisfying $0 \leq i_1, \dots, i_k \leq m$.

The grid partitions R into m^k smaller cubes. The next step is to partition each of the smaller cubes into k -dimensional simplices. There are standard ways to triangulate cubes; Kuhn's triangulation, for example, divides each subcube into $k!$ simplices (see Todd [1976]). See Figure 1 dots for an example of R , C , the grid and the triangulation in the case $k = 2$.

The next step of the algorithm is to find a linear underestimating function $r_i(\mathbf{y})$ for the quadratic term of (3) on each simplex. The existence and properties of linear underestimating functions are explained in the following

Figure 1: An example of the mesh and the triangulation.

lemma. This lemma follows from various lemmas in Pardalos and Rosen [1987]. For the sake of completeness, we state and prove it here.

Lemma 5 *Let S be a simplex in $\mathbb{R}^k$ with nondegenerate vertices $\{\mathbf{s}_1, \dots, \mathbf{s}_{k+1}\}$. Let d denote the diameter of S . Let $q(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T H \mathbf{y} + \mathbf{c}^T \mathbf{y}$ be a concave quadratic function defined on $\mathbb{R}^k$ (in which H is an $n \times n$ symmetric matrix). Then there exists a unique affine linear function $r(\mathbf{y}) = \mathbf{p}^T \mathbf{y} + a$ on $\mathbb{R}^k$ such that $q(\mathbf{s}_i) = r(\mathbf{s}_i)$ for $i = 1, \dots, k+1$. Moreover, q, r satisfy the following bounds for all $\mathbf{y} \in S$:*

$$r(\mathbf{y}) \leq q(\mathbf{y}) \leq r(\mathbf{y}) + \gamma d^2/4,$$

where $-\gamma$ is the smallest eigenvalue of H .

PROOF. The proof of existence and uniqueness of the affine function is straightforward; the condition $q(\mathbf{s}_i) = \mathbf{p}^T \mathbf{s}_i + a$ for $i = k+1$ gives $k+1$ linear equations for $k+1$ unknowns (the unknowns are $a, p_1, \dots, p_k$). The nondegeneracy assumption implies that this system has a unique solution.

Now we turn to the bounds claimed in the theorem. Observe that if $\mathbf{y} \in S$, then $\mathbf{y} = \lambda_1 \mathbf{s}_1 + \dots + \lambda_{k+1} \mathbf{s}_{k+1}$, where each λ_i is nonnegative and $\lambda_1 + \dots + \lambda_{k+1} = 1$.

Since r is affine linear,

$$\begin{aligned} r(\mathbf{y}) &= r\left(\sum_{i=1}^{k+1} \lambda_i \mathbf{s}_i\right) \\ &= \sum_{i=1}^{k+1} \lambda_i r(\mathbf{s}_i) \end{aligned}$$

$$= \sum_{i=1}^{k+1} \lambda_i q(\mathbf{s}_i).$$

Using this fact, we have the following computation:

$$\begin{aligned}
q(\mathbf{y}) - r(\mathbf{y}) &= q\left(\sum_{i=1}^{k+1} \lambda_i \mathbf{s}_i\right) - \sum_{i=1}^{k+1} \lambda_i q(\mathbf{s}_i) \\
&= \frac{1}{2} \left(\sum_{i=1}^{k+1} \lambda_i \mathbf{s}_i\right)^T H \left(\sum_{i=1}^{k+1} \lambda_i \mathbf{s}_i\right) + \mathbf{c}^T \left(\sum_{i=1}^{k+1} \lambda_i \mathbf{s}_i\right) \\
&\quad - \frac{1}{2} \sum_{i=1}^{k+1} \lambda_i \mathbf{s}_i^T H \mathbf{s}_i - \sum_{i=1}^{k+1} \lambda_i \mathbf{c}^T \mathbf{s}_i \\
&= \frac{1}{2} \left(\sum_{i=1}^{k+1} \lambda_i \mathbf{s}_i\right)^T H \left(\sum_{i=1}^{k+1} \lambda_i \mathbf{s}_i\right) - \frac{1}{2} \sum_{i=1}^{k+1} \lambda_i \mathbf{s}_i^T H \mathbf{s}_i \\
&= \frac{1}{2} \sum_{i=1}^{k+1} (\lambda_i^2 - \lambda_i) \mathbf{s}_i^T H \mathbf{s}_i + \sum_{\substack{i,j \\ i < j}} \lambda_i \lambda_j \mathbf{s}_i^T H \mathbf{s}_j \\
&= -\frac{1}{2} \sum_{i=1}^{k+1} \lambda_i (1 - \lambda_i) \mathbf{s}_i^T H \mathbf{s}_i + \sum_{\substack{i,j \\ i < j}} \lambda_i \lambda_j \mathbf{s}_i^T H \mathbf{s}_j \\
&= -\frac{1}{2} \sum_{i=1}^{k+1} \lambda_i \left(\sum_{\substack{j \\ j \neq i}} \lambda_j \right) \mathbf{s}_i^T H \mathbf{s}_i + \sum_{\substack{i,j \\ i < j}} \lambda_i \lambda_j \mathbf{s}_i^T H \mathbf{s}_j \\
&= -\frac{1}{2} \sum_{i=1}^{k+1} \sum_{\substack{j \\ j \neq i}} \lambda_i \lambda_j \mathbf{s}_i^T H \mathbf{s}_i + \sum_{\substack{i,j \\ i < j}} \lambda_i \lambda_j \mathbf{s}_i^T H \mathbf{s}_j \\
&= -\frac{1}{2} \sum_{i=1}^{k+1} \sum_{\substack{j \\ j < i}} \lambda_i \lambda_j (\mathbf{s}_i^T H \mathbf{s}_i + \mathbf{s}_j^T H \mathbf{s}_j) + \sum_{\substack{i,j \\ i < j}} \lambda_i \lambda_j \mathbf{s}_i^T H \mathbf{s}_j \\
&= \frac{1}{2} \sum_{\substack{i,j \\ j < i}} \lambda_i \lambda_j (-\mathbf{s}_i^T H \mathbf{s}_i - \mathbf{s}_j^T H \mathbf{s}_j + 2\mathbf{s}_i^T H \mathbf{s}_j) \\
&= \frac{1}{2} \sum_{\substack{i,j \\ j < i}} \lambda_i \lambda_j (-(\mathbf{s}_i - \mathbf{s}_j)^T H (\mathbf{s}_i - \mathbf{s}_j)).
\end{aligned}$$

Now, since H is negative definite, we know that

$$0 \leq -(\mathbf{s}_i - \mathbf{s}_j)^T H(\mathbf{s}_i - \mathbf{s}_j) \leq \|\mathbf{s}_i - \mathbf{s}_j\|_2 \cdot \|H\|_2 \cdot \|\mathbf{s}_i - \mathbf{s}_j\|_2 \leq d^2 \gamma$$

where γ is the negative of the smallest eigenvalue of H .

This gives us the chain of inequalities:

$$\begin{aligned} 0 \leq q(\mathbf{y}) - r(\mathbf{y}) &\leq (d^2 \gamma) \cdot \frac{1}{2} \sum_{\substack{i,j \\ j < i}} \lambda_i \lambda_j \\ &\leq (d^2 \gamma) \cdot \frac{1}{4} \sum_{i=1}^{k+1} \lambda_i \sum_{j=1}^{k+1} \lambda_j \\ &\leq d^2 \gamma / 4. \end{aligned}$$

This concludes the proof of the lemma. ■

The next step of the algorithm is to compute the affine linear function described in the previous lemma for each of the t simplices contained in R constructed above. Here, we have defined $t = k!m^k$. The function $q(\mathbf{y})$ is taken to be the quadratic function $\frac{1}{2}\mathbf{y}^T D\mathbf{y} + \mathbf{c}^T \mathbf{y}$ in (3). Let the simplices be denoted $S_1, \dots, S_t$. Let $r_1, \dots, r_t$ be the affine functions constructed as in the lemma.

Now, for each i , we minimize $r_i(\mathbf{y}) + \phi(\mathbf{y})$ on simplex S_i . This is a linear programming problem. In particular, it is equivalent to the following optimization problem:

$$\begin{aligned} &\text{minimize} && r_i(\mathbf{y}) + \mathbf{f}^T \mathbf{z} \\ &\text{subject to} && A\mathbf{y} + B\mathbf{z} \geq \mathbf{b}, \\ &&& \mathbf{y} \in S_i. \end{aligned} \tag{4}$$

This problem can be solved in time ℓ .

Finally, return as the approximate solution to the original problem (2) the vector $(\mathbf{y}^\circ, \mathbf{z}^\circ)$ that achieves the minimum objective function value among all t linear programs enumerated in (4) above.

Now we prove bounds to show that this is an ϵ -approximate solution. The simpler bound is the gap between the objective function at $(\mathbf{y}^\circ, \mathbf{z}^\circ)$ and the optimal point $(\mathbf{y}^*, \mathbf{z}^*)$.

Lemma 6 *Let $(\mathbf{y}^\diamond, \mathbf{z}^\diamond)$ be computed as above. Let $(\mathbf{y}^*, \mathbf{z}^*)$ be an optimum for (2). Then*

$$[q(\mathbf{y}^\diamond) + \phi(\mathbf{y}^\diamond)] - [q(\mathbf{y}^*) + \phi(\mathbf{y}^*)] \leq \gamma k / (4m^2).$$

PROOF. Suppose $\mathbf{y}^*$ lies in simplex S_i for some i . Let $(\mathbf{y}^+, \mathbf{z}^+)$ denote the optimal value of (4) in S_i . Suppose $\mathbf{y}^\diamond$ lies in S_j . Since $(\mathbf{y}^\diamond, \mathbf{z}^\diamond)$ is the optimum to a problem of the form (4) with the lowest objective function value, we know

$$r_j(\mathbf{y}^\diamond) + \mathbf{f}^T \mathbf{z}^\diamond \leq r_i(\mathbf{y}^+) + \mathbf{f}^T \mathbf{z}^+ \quad (5)$$

Also, we know that

$$r_i(\mathbf{y}^+) + \mathbf{f}^T \mathbf{z}^+ \leq r_i(\mathbf{y}^*) + \mathbf{f}^T \mathbf{z}^* \quad (6)$$

by optimality of $(\mathbf{y}^+, \mathbf{z}^+)$.

Now we apply the bounds in the preceding lemma. Observe that the diameter of simplices S_i and S_j is $\sqrt{k}/m$, because this is the maximum distance between neighboring grid points in R . Notice also that γ by definition is the absolute value of the most negative diagonal entry of matrix D in (2). Then the lemma says that

$$r_i(\mathbf{y}^*) \leq q(\mathbf{y}^*)$$

and

$$r_j(\mathbf{y}^\diamond) \geq q(\mathbf{y}^\diamond) - \gamma k / (4m^2).$$

Furthermore, by the optimality of $\mathbf{z}^\diamond$ and $\mathbf{z}^*$, we know that

$$\mathbf{f}^T \mathbf{z}^\diamond = \phi(\mathbf{y}^\diamond)$$

and

$$\mathbf{f}^T \mathbf{z}^* = \phi(\mathbf{y}^*).$$

Combining these four relations with (5) and (6) yields:

$$q(\mathbf{y}^\diamond) + \phi(\mathbf{y}^\diamond) - \gamma k / (4m^2) \leq q(\mathbf{y}^*) + \phi(\mathbf{y}^*).$$

■

We have established an upper bound on the gap between the objective function at $(\mathbf{y}^\circ, \mathbf{z}^\circ)$ and optimum. Now we establish a lower bound on the gap between the maximum objective function value and the minimum objective value.

To find a point $(\mathbf{y}^\#, \mathbf{z}^\#)$ as in the definition of ϵ -approximation, let us recall that γ is the absolute value of the most negative entry of D . Say this entry is in the (i, i) position. Recall by construction of R that there exists a feasible point $(\mathbf{y}^0, \mathbf{z}^0)$ such that the i th entry of $\mathbf{y}^0$ is equal to 0. Similarly, there exists a point $(\mathbf{y}^1, \mathbf{z}^1)$ such that the i th entry of $\mathbf{y}^1$ is 1. We define

$$(\mathbf{y}^\#, \mathbf{z}^\#) = \frac{1}{2}(\mathbf{y}^0, \mathbf{z}^0) + \frac{1}{2}(\mathbf{y}^1, \mathbf{z}^1).$$

This point is feasible for (2) since it is the convex combination of two feasible points.

We now establish a bound on how far this point lies above optimal.

Lemma 7 *The following bound holds:*

$$\frac{1}{2}\mathbf{y}^{\#T}D\mathbf{y}^\# + \mathbf{c}^T\mathbf{y}^\# + \mathbf{f}^T\mathbf{z}^\# \geq \frac{1}{2} \left[q(\mathbf{y}^0) + \phi(\mathbf{y}^0) + q(\mathbf{y}^1) + \phi(\mathbf{y}^1) \right] + \gamma/8.$$

PROOF. First notice that

$$q(\mathbf{y}^0) + \mathbf{f}^T\mathbf{z}^0 \geq q(\mathbf{y}^0) + \phi(\mathbf{y}^0)$$

by definition of ϕ , and similarly,

$$q(\mathbf{y}^1) + \mathbf{f}^T\mathbf{z}^1 \geq q(\mathbf{y}^1) + \phi(\mathbf{y}^1).$$

Next, we have the following chain of equations.

$$\begin{aligned} \frac{1}{2}\mathbf{y}^{\#T}D\mathbf{y}^\# + \mathbf{c}^T\mathbf{y}^\# + \mathbf{f}^T\mathbf{z}^\# &= \frac{1}{8}(\mathbf{y}^0 + \mathbf{y}^1)^T D(\mathbf{y}^0 + \mathbf{y}^1) + \frac{1}{2}\mathbf{c}^T(\mathbf{y}^0 + \mathbf{y}^1) \\ &\quad + \frac{1}{2}\mathbf{f}^T(\mathbf{z}^0 + \mathbf{z}^1) \\ &= \left(\frac{1}{4}\mathbf{y}^{0T}D\mathbf{y}^0 + \frac{1}{2}\mathbf{c}^T\mathbf{y}^0 + \frac{1}{2}\mathbf{f}^T\mathbf{z}^0 \right) \\ &\quad + \left(\frac{1}{4}\mathbf{y}^{1T}D\mathbf{y}^1 + \frac{1}{2}\mathbf{c}^T\mathbf{y}^1 + \frac{1}{2}\mathbf{f}^T\mathbf{z}^1 \right) \end{aligned}$$

$$\begin{aligned}
& + \left(-\frac{1}{8}\mathbf{y}^{0T} D \mathbf{y}^0 + \frac{1}{4}\mathbf{y}^{0T} D \mathbf{y}^1 - \frac{1}{8}\mathbf{y}^{1T} D \mathbf{y}^1 \right) \\
= & \frac{1}{2}(q(\mathbf{y}^0) + \mathbf{f}^T \mathbf{z}^0) + \frac{1}{2}(q(\mathbf{y}^1) + \mathbf{f}^T \mathbf{z}^1) \\
& - \frac{1}{8}(\mathbf{y}^0 - \mathbf{y}^1)^T D (\mathbf{y}^0 - \mathbf{y}^1).
\end{aligned}$$

Comparing this chain of equations to the previous paragraph, we see that we could prove the lemma if we could get a positive lower bound on $-(\mathbf{y}^0 - \mathbf{y}^1)^T D (\mathbf{y}^0 - \mathbf{y}^1)/8$. Let us express this term in terms of coordinates:

$$-\frac{1}{8}(\mathbf{y}^0 - \mathbf{y}^1)^T D (\mathbf{y}^0 - \mathbf{y}^1) = \frac{1}{8} \sum_{j=1}^k (y_j^0 - y_j^1)^2 \cdot (-d_{jj})$$

where the diagonal entries of D are denoted $d_{11}, \dots, d_{kk}$. Notice that all the terms of this summation are nonnegative. Focusing on the i th term, we observe that $(y_i^0 - y_i^1)^2 = 1$ by choice of $\mathbf{y}^0$ and $\mathbf{y}^1$. Moreover, $-d_{ii} = \gamma$, the 2-norm of matrix D . Thus, we get a lower bound of $\gamma/8$ needed for the lemma. ■

Corollary 8 *The objective function value of the point $(\mathbf{y}^\#, \mathbf{z}^\#)$ is at least $\gamma/8$ above optimal.*

PROOF. This follows from the inequality established in the previous lemma after we notice that

$$q(\mathbf{y}^0) + \phi(\mathbf{y}^0) \geq q(\mathbf{y}^*) + \phi(\mathbf{y}^*)$$

and similarly for $\mathbf{y}^1$. ■

Thus, we have proved that we can find a point $(\mathbf{y}^\diamond, \mathbf{z}^\diamond)$ that is within $\gamma k/(4m^2)$ of optimum. On the other hand, there exists at least one point that is at least $\gamma/8$ above optimal. Thus, $(\mathbf{y}^\diamond, \mathbf{z}^\diamond)$ is an ϵ -approximation to optimum provided that

$$\frac{\gamma k}{4m^2} \leq \epsilon \cdot \frac{\gamma}{8},$$

i.e.,

$$m \geq \sqrt{\frac{2k}{\epsilon}}.$$

If we choose m to be the first integer larger than this number, we have established that $(\mathbf{y}^\circ, \mathbf{z}^\circ)$ is an ϵ -approximate solution. The running time is proportional to the time needed to solve $k!m^k$ linear programming problems, i.e.,

$$k! \left\lceil \sqrt{\frac{2k}{\epsilon}} \right\rceil^k \ell$$

steps.

This proves the main theorem. In practice, a more efficient version of the algorithm proposed in this section would construct a hierarchy of triangulations of R . For example, suppose that m is divisible by 2. Then we could partition R into a mesh with $m/2 + 1$ grid points in each dimension and carry out the algorithm of this section. The point returned by the algorithm from the coarse grid would not necessarily be an ϵ -approximation, but the results would give good upper and lower bounds on the value of the objective function in each cube. We could then use these bounds to determine which grid cells need to be further subdivided. Applying these ideas in a hierarchical fashion would lead to a branch-and-bound style algorithm. See Pardalos and Rosen [1987] for algorithms of this style; see Papadimitriou and Steiglitz [1982] for a general description of branch and bound.

3 Dependence on ϵ and k .

In this section we give some indication why the dependence on ϵ and k that we obtained in the main theorem might be expected. First, we address the dependence on ϵ . To simplify this discussion, let us restrict attention to the case $k = 1$, that is, a quadratic objective function of the form $cy^2 + \mathbf{f}^T \mathbf{z}$, where y is a scalar unknown and $c < 0$. Suppose that we were able to obtain an approximation algorithm with running time dependence on ϵ better than $1/\epsilon$. Suppose, for example, that we had an approximation algorithm whose running time were polynomial in $|\log \epsilon|$. It is known that if a point is sufficiently close to optimal for an instance of CQP—in particular, within $2^{-O(L)}$ of optimal—then an exact optimum may be found in polynomial time. Here, L denotes the number of bits needed to write the problem. If there were an approximation algorithm whose running time were poly-

mial in $|\log \epsilon|$, then we set $\epsilon = 2^{-O(L)}$ and in polynomial time come up with an exact solution to the concave problem with objective function $cy^2 + f^T z$.

This would seem to contradict recent results by Pardalos and Vavasis [1990], who proved that problems of this form (i.e., CQP problems with $k = 1$) are NP-hard. Thus, assuming $P \neq NP$, polynomial dependence on $|\log \epsilon|$ is not expected, and therefore polynomial dependence on $1/\epsilon$ seems like it might be the best possible.

Next, we investigate the dependence on k . In our bound the dependence on k is exponential. Suppose there were an approximation algorithm whose running time were polynomial in k and $1/\epsilon$. We claim that the existence of such an algorithm would imply that $P = NP$. In particular, consider the following transformation from the boolean 3-satisfiability problem to CQP. Let Q be an instance of 3SAT, i.e., a collection of m clauses each with three literals, and a total of n variables, $x_1, \dots, x_n$. We can make a CQP instance out of Q as follows. First, let there be one variable y_i for each boolean variable x_i . The objective function is:

$$\sum_{i=1}^n y_i(1 - y_i).$$

This is a concave quadratic objective function. There are $2n + m$ constraints. The first $2n$ constraints are:

$$\begin{aligned} 0 &\leq y_1 \leq 1, \\ &\vdots \\ 0 &\leq y_n \leq 1. \end{aligned}$$

These constraints imply the following facts: the objective function is non-negative on the feasible region, and, moreover, the objective function is zero iff every y_i is assigned to 0 or 1.

The remaining m constraints express the boolean clauses. For example, a clause of the form $x_5 \vee x_7 \vee \bar{x}_9$ would be expressed as a constraint:

$$y_5 + y_7 + (1 - y_9) \geq 1.$$

We claim that the boolean formula is satisfiable iff the minimum of the objective function on the feasible region is 0. This is proved using standard arguments.

Suppose now that we had a $3/(4n)$ -approximate solution to this problem. First, observe that under the assumption of an approximation algorithm running in polynomial time in $1/\epsilon$ and k (in this case, $k = n$), a $3/(4n)$ -approximate solution could be found in polynomial time. Observe that the maximum possible value of the objective function is $n/4$ (obtained by setting each y_i to $1/2$). In other words, the gap between minimum and maximum objective values is no more than $n/4$. Thus, a $3/(4n)$ approximate solution could be no more than $3/16$ above optimum. If the boolean formula were satisfiable, the approximation algorithm would have to return an assignment to $(y_1, \dots, y_n)$ such that the objective function is at most $3/16$. In particular, each term of the objective function is at most $3/16$, i.e., each y_i is either in the interval $[0, 1/4]$ or $[3/4, 1]$. But we claim that from such an assignment to the y_i 's, we can in fact read off a satisfying assignment to the original boolean formula. In particular, whenever y_i is between $[0, 1/4]$, set $x_i = 0$, else set $x_i = 1$.

Thus, from a $3/(4n)$ -approximate solution we can determine whether the original formula is satisfiable (by checking whether the $3/(4n)$ -approximate solution has an objective function value less than or equal to $3/16$).

Accordingly, we would expect that an algorithm polynomial in $1/\epsilon$ and k does not exist.

4 Open questions

Perhaps the most interesting open question is whether this work extends to indefinite quadratic programming (IQP) problems. As an example of the difficulties with the indefinite case, Murty and Kabadi [1987] show (among other things) that it is NP-hard to tell whether an indefinite quadratic program is unbounded or not. (The corresponding problem for CQP is solvable in polynomial time, and in fact our approximation algorithm detects unboundedness as indicated in Lemma 4.) It seems unreasonable that an approximation algorithm would be unable to detect unboundedness in a problem.

The difficulty mentioned in last paragraph is avoided by assuming that the feasible region of the problem is compact, in which case the problem cannot be unbounded. Even in this case, our techniques are only partially successful. In particular, our bounds concerning the point $(x^\#, y^\#)$ fail in

the indefinite case. These bounds were necessary to establish the main theorem.

Finding more efficient approximation algorithms is another interesting question. For example, could a branch-and-bound algorithm of the sort described in Section 2 give better worst case bounds? Could randomization help?

Another interesting question concerns the relationship between CQP and combinatorial problems in NP. Since concave quadratic programming is NP-hard, any combinatorial problem in NP can be expressed as an instance of CQP. However, CQP is general enough that many combinatorial problems can be expressed as a CQP instance without the need for complicated “gadgets.” This suggests that there could be a connection between approximation algorithms for CQP and approximation for a variety of combinatorial problems.

A final question concerns the relation between our definition of ϵ -approximate and the standard combinatorial definition mentioned in the introduction. Although the combinatorial definition does not extend to nonlinear problems, our definition *does* extend to combinatorial problems.

In particular, for any combinatorial optimization problem there is a feasible set S and a real-valued objective function f defined on S . Then clearly the inequality of Definition 1 can be applied to this case. The definition does not always match up with the standard combinatorial definition. For example, it is unclear whether the approximation algorithms for problem ΔTSP given in Papadimitriou and Steiglitz are approximation algorithms in our sense. Perhaps these algorithms return the worst possible feasible solution. It would be interesting to investigate these possibilities.

Acknowledgements

Thanks to Rajeev Motwani of Stanford for originally suggesting to me the problem of approximation algorithms for nonlinear programming. Thanks to Panos Pardalos of Penn State for valuable discussions on this subject.

References

- PHILIP E. GILL, WALTER MURRAY, AND MARGARET H. WRIGHT [1981],
Practical Optimization, Academic Press, London.

- GENE H. GOLUB AND CHARLES F. VAN LOAN [1989], *Matrix Computations, Second Edition*, Johns Hopkins University Press, Baltimore.
- L. G. HAČIJAN [1979], A polynomial algorithm in linear programming, *Dokl. Akad. Nauk SSSR* 244:1093–1096. Translated in *Soviet Math. Dokl.* 20:191–194.
- NAOKI KATOH AND TOSHIHIDE IBARAKI [1987], *Discrete Applied Mathematics* 17:39–66.
- SANJIV KAPOOR AND PRAVIN M. VAIDYA [1986], Fast algorithms for convex quadratic programming and multicommodity flows, *Proc. 18th Annual ACM Symposium on Theory of Computing*, 147–159.
- M. K. KOZLOV, S. P. TARASOV AND L. G. HAČIJAN [1979], Polynomial solvability of convex quadratic programming, *Dokl. Akad. Nauk SSSR* 248:1049–1051. Translated in *Soviet Math. Dokl.* 20:1108–1111.
- KATTA G. MURTY AND SANTOSH N. KABADI [1987], Some NP-complete problems in quadratic and nonlinear programming, *Math. Progr.* 39:117–129.
- CHRISTOS H. PAPADIMITRIOU AND KENNETH STEIGLITZ [1982], *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, New Jersey.
- P. M. PARDALOS AND J. B. ROSEN [1987], *Constrained Global Optimization: Algorithms and Applications*, Springer Verlag Lecture Notes in Computer Science vol. 268, Berlin.
- PANOS M. PARDALOS AND STEPHEN A. VAVASIS [1990], Quadratic programming with one negative eigenvalue is NP-hard, Technical Report 90-1160, Department of Computer Science, Cornell University, Ithaca, New York.
- SARTAJ SAHNI [1974], Computationally related problems, *SIAM J. Comp.* 3:262–279.
- MICHAEL J. TODD [1976], *The Computation of Fixed Points and Applications*, Springer, Berlin.

- PRAVIN M. VAIDYA [1989], Speeding-up linear programming using fast matrix multiplication (extended abstract), *Proc. 30th Symp. Foundations of Computer Science*, 332–337.
- STEPHEN A. VAVASIS [1990], Quadratic programming is in NP, *Inf. Proc. Lett.* 36:73-77.
- YINYU YE AND EDISON TSE [1989], An extension of Karmarkar's projective algorithm for convex quadratic programming, *Math. Progr.* 44:157–179.