Quadratic Programming with One Negative Eigenvalue Is NP-Hard

PANOS M. PARDALOS[1] and STEPHEN A. VAVASIS[2]*
[1]*Department of Computer Science, Pennsylvania State University, University Park, PA 16802, U.S.A.*; [2]*Department of Computer Science, Cornell University, Ithaca, NY 14853, U.S.A.*

(Received: 6 February 1991)

Abstract. We show that the problem of minimizing a concave quadratic function with one concave direction is NP-hard. This result can be interpreted as an attempt to understand exactly what makes nonconvex quadratic programming problems hard. Sahni in 1974 [8] showed that quadratic programming with a negative definite quadratic term (n negative eigenvalues) is NP-hard, whereas Kozlov, Tarasov and Hačijan [2] showed in 1979 that the ellipsoid algorithm solves the convex quadratic problem (no negative eigenvalues) in polynomial time. This report shows that even one negative eigenvalue makes the problem NP-hard.

Key words. Global optimization, quadratic programming, NP-hard.

1. Introduction

Nonlinear optimization includes many subclasses of problems. Up to date, the best methods seem naturally to exist for convex programming problems; see, for example [2]. Nonconvex minimization problems do not appear to admit efficient algorithms. The problem of designing algorithms that find global solutions is very difficult, since in general, there are no local criteria in deciding whether a local optimum is global [5], [6].

In this paper we consider the complexity of a class of nonconvex quadratic problems. The general concave quadratic problem has the following form:

$$\begin{aligned} &\min f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ &\text{s.t. } A\mathbf{x} \leq \mathbf{b}, \end{aligned} \tag{1}$$

where $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$, A is an $m \times n$ matrix, $\mathbf{b} \in \mathbb{R}^m$, and Q is an $n \times n$ symmetric negative semidefinite matrix. It is well known that this problem is NP-hard [8]. In fact, many well known combinatorial optimization problems (e.g. linear 0–1 programming) can be formulated as global concave minimization problems [6]. The simplest case of concave programming is when the corresponding symmetric

* This author's work 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.

Journal of Global Optimization **1**: 15–22, 1991

matrix Q is of rank one, with exactly one negative eigenvalue. We call this problem QP1NE.

In this paper we prove that QP1NE is also NP-hard (because of [9], this implies that this problem is NP-complete). Related complexity results on nonconvex quadratic programming can be found in [1], [3], and [7]. Although complexity results of this nature characterize worst case instances, they nevertheless are indicative of the computational difficulty of the problem. In addition, these results enhance our understanding of what makes nonconvex problems so difficult to solve.

2. Main Construction

In this section and the next section we construct the quadratic programming problem that is NP-hard. We will transform the clique problem to QP1NE. The actual instance of clique to be transformed will be introduced in next section; in this section we set up the framework for the quadratic program.

The objective function of the quadratic program will involve two variables, w and z. The actual objective function is:

$$\text{minimize } z - w^2 . \tag{2}$$

Notice that this objective function is a quadratic function with one negative eigenvalue.

In addition to w and z there will be two sequences of variables, $x_1, \dots, x_n$ and $y_{1,2}, \dots, y_{n-1,n}$. We have a variable y_{ij} for all pairs of indices (i, j) such that $1 \leq i < j \leq n$; this implies that there are $(n-1)n/2$ such variables. Thus, our instance of quadratic programming has a total of $n + 2 + (n-1)n/2$ variables.

The first sequence of constraints is

$$\begin{aligned} &0 \leq x_1 \leq 1 , \\ &\quad\vdots \\ &0 \leq x_n \leq 1 . \end{aligned} \tag{3}$$

These constraints specify an n-cube; if $(x_1, \dots, x_n)$ is a vertex of the cube, i.e., $x_i \in \{0, 1\}$ for each i, then we call $(x_1, \dots, x_n)$ a *binary* point.

Next, we let $b > 1$ be a constant to be specified below (it will turn out that $b = 4$ is acceptable). Variable w in the objective function is constrained by a linear equation:

$$w = bx_1 + b^2x_2 + \cdots + b^nx_n . \tag{4}$$

Next, we want to define a sequence of constraints so that $z = w^2$ at all binary feasible points. In this regard, notice that

$$w^2 = b^2x_1^2 + b^4x_2^2 + \cdots + b^{2n}x_n^2 + \sum_{1 \leq i < j \leq n} 2b^{i+j}x_ix_j . \tag{5}$$

We cannot include this constraint in our problem since it is nonlinear. Instead, we include a sequence of linear constraints to mimic this constraint at binary points. In particular, notice that at a binary point $x_i = x_i^2$. We introduce the variables y_{ij} for $1 \leqslant i < j \leqslant n$ constrained by

$$y_{ij} \geqslant 0 \tag{6}$$

and

$$y_{ij} \geqslant x_i + x_j - 1 \,. \tag{7}$$

Notice that the minimum value for y_{ij} at a binary point is exactly $x_i x_j$ under these two constraints. Moreover, these constraints are linear.

Now, finally we introduce the linear equation constraint on z to mimic the formula (5) for w^2:

$$z = b^2 x_1 + b^2 x_2 + \cdots + b^{2n} x_n + \sum_{1 \leqslant i < j \leqslant n} 2b^{i+j} y_{ij} \,. \tag{8}$$

For the remainder of this section we consider the quadratic program with objective function (2) and linear constraints (3), (4), (6), (7), and (8). In the next section we introduce additional constraints on the x_i's based on the instance of clique, but for now we establish properties of the problem as its stands. The properties are established via lemmas; the main property is Theorem 1, which states that a feasible point is optimal if and only if it is a binary feasible point.

We start with a general remark that the objective function $z - w^2$ depends linearly on each y_{ij}, and this dependence involves a positive coefficient. Therefore, if we are given some setting of the x_i's and we are trying to minimize $z - w^2$, then we ought to choose the minimum setting for each y_{ij} feasible with the given setting of the x_i's.

In other words, if $(x_1, \ldots, x_n)$ are fixed, there is a unique setting for the remaining variables in order to achieve the lowest objective function value. In particular, set $y_{ij} = \max(x_i + x_j - 1, 0)$ and set w, z according to (4) and (8). Thus, we will often specify a feasible point by providing only the values of variables $x_1, \ldots, x_n$.

LEMMA 1. *For a binary feasible point, the objective function is zero* (*provided the remaining variables are chosen optimally*).

Proof. This is true by construction. □

In the next lemma we characterize possible minima of the objective function. The main theorem afterwards sharpens this result.

LEMMA 2. *The minimum value of the objective function can be achieved only at a point such that $x_i \in \{0, 1/2, 1\}$ for each i.*

Proof. Let $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{w}, \hat{z})$ be a feasible point, possibly a candidate for optimum. Without loss of generality, we can assume that the $\hat{y}_{ij}$ values are at their minimum possible feasible values. Suppose for some i^* that $\hat{x}_{i^*} \notin \{0, 1/2, 1\}$. Implicit in the upcoming paragraphs is a proof that the point $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{w}, \hat{z})$ cannot be an extreme point of the polytope, and we will argue that this implies suboptimality.

Let

$$I = \{i : \hat{x}_i = \hat{x}_{i^*}\} .$$

Also, let

$$J = \{i : \hat{x}_i = 1 - \hat{x}_{i^*}\} .$$

Note that J might be empty. Note also that since $\hat{x}_{i^*} \neq 1/2$, $I \cap J = \emptyset$.

Now, consider the point $(\mathbf{x}(t), \mathbf{y}(t), w(t), z(t))$ parametrized by t as follows. For $i \in I$, $x_i(t) = \hat{x}_i + t$. For $i \in J$, $x_i(t) = \hat{x}_i - t$. For $i \notin I \cup J$, $x_i(t) = \hat{x}_i$. For all (i, j) such that $\hat{y}_{ij} = \hat{x}_i + \hat{x}_j - 1$ ($\hat{y}_{ij}$ is either equal to this quantity or to zero), we let $y_{ij}(t) = x_i(t) + x_j(t) - 1$. For the other (i, j), let $y_{ij}(t) = 0$. Finally, we let $w(t)$ and $z(t)$ vary linearly with $\mathbf{x}(t)$, $\mathbf{y}(t)$ so that (4) and (8) are satisfied, i.e.,

$$w(t) = bx_1(t) + \cdots + b^n x_n(t)$$

and similarly for $z(t)$.

Notice by construction that $(\mathbf{x}(0), \mathbf{y}(0), w(0), z(0)) = (\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{w}, \hat{z})$. We claim that there exists an $\epsilon > 0$ such that $(\mathbf{x}(t), \mathbf{y}(t), w(t), z(t))$ is feasible for all $t \in (-\epsilon, \epsilon)$. Notice that the constraints $0 \leqslant x_i \leqslant 1$ are satisfied for ϵ small enough since the x_i's that vary with t lie strictly between 0 and 1.

Next we check the constraints on the $y'_{ij}s$. There are three cases. The first case is that $\hat{x}_i + \hat{x}_j - 1 < 0$ in which case $\hat{y}_{ij} = 0$ and $y_{ij}(t) = 0$, so $y_{ij}(t)$ is greater than $x_i(t) + x_j(t) - 1$ for t close enough to zero.

The second case is that $\hat{x}_i + \hat{x}_j - 1 > 0$, in which case $\hat{y}_{ij} > 0$. Moreover, in this case, $y_{ij}(t) = x_i(t) + x_j(t) - 1$, and the constraint $y_{ij}(t) \geqslant 0$ is satisfied for t close enough to zero.

The last case is that $\hat{x}_i + \hat{x}_j - 1 = \emptyset$. In this case $y_{ij}(t) = x_i(t) + x_j(t) - 1$. Now, there are two subcases; in the first subcase $\{i, j\} \cap (I \cup J) = \emptyset$. In this case $y_{ij}(t)$ does not vary with t and hence remains feasible. In the second subcase, say $i \in I$, we know that $\hat{x}_j = 1 - \hat{x}_i$ (by the hypothesis for this case). This means that $j \in J$ by definition of J. Thus, $x_j(t) = \hat{x}_j - t$ and $x_i(t) = \hat{x}_i + t$, so the t's cancel in the expression for $y_{ij}(t)$.

Finally, the constraints for w and z are satisfied by definition of $w(t)$, $z(t)$.

Thus, we have constructed a feasible line segment containing $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{w}, \hat{z})$ in its interior. We claim that $w(t)$ depends in nontrivial way on t. Note that $w(t)$ takes the form

$$w(t) = \hat{w} + t \cdot \sum_{i=1}^{n} \sigma_i b^i$$

where $\sigma_i \in \{-1, 0, 1\}$ for each i, and at least one σ_i is nonzero (corresponding to i^*). Thus, w depends linearly on t with the coefficient of dependence being the summation above. If we assume that $b \geqslant 2$, then this summation cannot be zero because the largest nonzero term cannot be cancelled by subsequent terms.

The objective function on the segment takes the form $z(t) - w(t)^2$. This objective function actually corresponds to points in the feasible region for all $t \in (-\epsilon, \epsilon)$. This is a quadratic function $a_0 t^2 + a_1 t + a_2$ of one variable t whose leading coefficient is negative (because the linear term in $w(t)$ is nonzero). Such a function cannot have its minimum at $t = 0$.

This concludes the proof that $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{w}, \hat{z})$ cannot be a minimum. □

THEOREM 1. *A feasible point is a minimum for the objective function $z - w^2$ if and only if it is a binary point.*

Proof. Let $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{w}, \hat{z})$ be a minimum. As usual, the $\hat{y}_{ij}$'s are the minimum possible. By the previous lemma, we know that $\hat{x}_i \in \{0, 1/2, 1\}$. We also known that the objective function is equal to zero at a binary point. Accordingly, the only claim that we need to prove is that if some $\hat{x}_i = 1/2$, then the objective function is positive (so the point is not optimal).

To prove this, let I be the set of indices such that $\hat{x}_i = 1/2$, and let $\bar{I}$ denote the complementary set of indices. Define

$$s = \sum_{i \in \bar{I}} b^i \hat{x}_i$$

and

$$t = \sum_{i \in I} b^i .$$

Notice with these definitions that $\hat{w} = s + t/2$.

Next we try to determine the value of $\hat{z}$. Note that for $i, j \in \bar{I}$, $\hat{y}_{ij} = \hat{x}_i \hat{x}_j$. For $i, j \in I$, $\hat{y}_{ij} = 0$. Finally, for $i \in \bar{I}$, $j \in I$, we have that $\hat{y}_{ij} = \hat{x}_i/2$ (this follows because $\hat{y}_{ij} = \max(0, \hat{x}_i + 1/2 - 1)$). Similarly, for $i \in I$, $j \in \bar{I}$, $\hat{y}_{ij} = \hat{x}_j/2$.

Thus, we have

$$\begin{aligned} \hat{z} &= \sum_{i=1}^{n} b^{2i} \hat{x}_i + \sum_{1 \le i < j \le n} 2b^{i+j} \hat{y}_{ij} \\ &= T_1 + T_2 + T_3 \end{aligned}$$

where

$$\begin{aligned} T_1 &= \sum_{i \in \bar{I}} b^{2i} \hat{x}_i + \sum_{\substack{i<j \\ i,j \in \bar{I}}} 2b^{i+j} \hat{y}_{ij} \\ &= s^2 . \end{aligned}$$

Also

$$\begin{aligned} T_2 &= \sum_{\substack{i<j \\ i \in \bar{I}, j \in I}} 2b^{i+j} \hat{y}_{ij} + \sum_{\substack{i>j \\ i \in \bar{I}, j \in I}} 2b^{i+j} \hat{y}_{ji} \\ &= \sum_{i \in \bar{I}, j \in I} 2b^{i+j} \hat{x}_i/2 \\ &= \sum_{j \in I} b^j \sum_{i \in \bar{I}} b^i \hat{x}_i \\ &= st . \end{aligned}$$

Finally,

$$T_3 = \sum_{i \in I} b^{2i} \hat{x}_i$$
$$= \frac{1}{2} \sum_{i \in I} b^{2i} .$$

Thus, $\hat{z} = s^2 + st + T_3$. Accordingly,

$$\hat{z} - \hat{w}^2 = s^2 + st + T_3 - (s + t/2)^2$$
$$= T_3 - t^2/4 .$$

Thus, we must prove that $T_3 - t^2/4$ is positive (assuming $I \neq \emptyset$).

Let m be the highest numbered index in I. Then we see that $T_3 \geqslant b^{2m}/2$. On the other hand, assuming $b \geqslant 4$ (so that $b - 1 \geqslant 3b/4$), we have

$$t^2 = \left(\sum_{i \in I} b^i\right)^2$$
$$\leqslant \left(\sum_{i=1}^{m} b^i\right)^2$$
$$\leqslant \left(\frac{b^{m+1} - b}{g - 1}\right)^2$$
$$\leqslant \left(\frac{b^{m+1} - b}{3b/4}\right)^2$$
$$\leqslant \frac{16}{9} 4(b^m - 1)^2$$
$$\leqslant \frac{16}{9} b^{2m} .$$

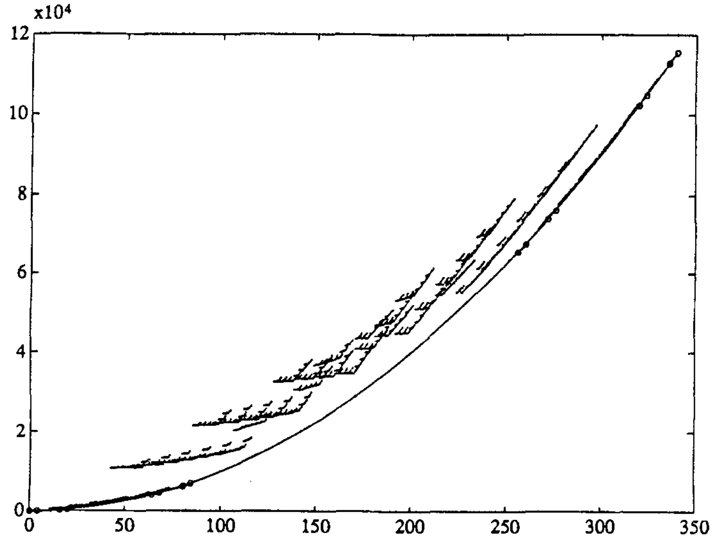

Fig. 1. A plot of z versus w for feasible points.

Thus, $T_3 \geq b^{2m}/2$ whereas $t^2/4 \leq (4/9)b^{2m}$. This proves that $T_3 > t^2/4$.

This concludes the proof that a feasible point is optimum iff it is binary. □

In Figure 1 we have illustrated the construction of this section. In particular, for the $n = 4$ case, we have generated 2401 testpoints $\mathbf{x} \in [0, 1]^4$. From each $\mathbf{x}$ we generate $(\mathbf{y}, w, z)$ and plotted (w, z) for each testpoint (each pair is represented by a dot). The 16 binary testpoints are plotted as small circles. The curve $z - w^2 = 0$ has also been plotted. This plot was generated by an implementation of the formulas in this section using MATLAB™, a software package developed by the MathWorks, Inc.

3. Transforming CLIQUE to QP1NE

We now explain how to transform CLIQUE to QP1NE. Suppose we are given an instance of clique, that is, an integer k and an undirected graph G with n vertices numbered 1 to n. The CLIQUE problem is to decide whether the graph has k vertices all connected to one another.

We explain how to transform this problem to QP1NE. First, write out the objective function and all the constraints specified in the previous section, i.e., (2), (3), (4), (6), (7) and (8). The number of x_i variables should be equal to n, the number of vertices of the graph.

Next introduce some additional constraints. For all pairs (i, j) such that edge (i, j) is not present in the graph, we add the constraint $x_i + x_j \leq 1$. We also add the constraint $x_1 + \cdots + x_n = k$.

Now, the claim is that the graph has a clique of size k if and only if the optimum of the QP1NE instance is zero. First, suppose the graph has a clique C of size k. Then if we set $x_i = 1$ for each $i \in C$, and $x_i = 0$ for $i \notin C$ we get a binary feasible point (feasible also with respect to the constraints introduced in the previous paragraph). This point has an optimum value of zero as proved by Theorem 1.

Conversely, suppose the optimum value of QP1NE is zero, say at point $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{w}, \hat{z})$. Then Theorem 1 tells us that $\hat{x}_i \in \{0, 1\}$ for all i. Then clearly the set of i such that $x_i = 1$ form a k-clique.

4. Conclusion

The question of how efficiently we can compute a global minimum of a nonlinear program appears to be very difficult. In this paper we tried to explain this difficulty from the point of view of complexity classes. It is proved that concave quadratic minimization with one negative eigenvalue is NP-hard. Since this is the simplest nonconvex quadratic problem, the result indicates that a major degree of difficulty is introduced even when one nonconvex variable appears in the objective function.

An interesting open question is the complexity of computing the global minimum to the class of quadratic functions that can be decomposed as the product of two nonparallel linear functions [4]. This is a special case of the more general problem of minimizing a rank-2 quadratic function with one negative and one positive eigenvalue. This latter problem is easily seen to be NP-hard from the results of this paper – add the square of a new unconstrained variable to the objective function we constructed in Section 2.

The present result suggests the conjecture that the problem of minimizing a product of linear functions is also NP-hard.

References

1. Garey, M. R. and Johnson, D. S. (1979), *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco.
2. Kozlov, M. K., Tarasov, S. P., and Hačijan, L. G. (1979), Polynomial Solvability of Convex Quadratic Programming, *Soviet Math. Doklady* **20**, 1108–1111.
3. Murty, K. G. and Kabadi, S. N. (1987), Some NP-Complete Problems in Quadratic and Non-linear Programming, *Mathematical Programming* **39**, 117–129.
4. Pardalos, P. M. (1990), Polynomial Time Algorithms for Some Classes of Nonconvex Quadratic Problems, To appear in *Optimization*.
5. Pardalos, P. M. and Rosen, J. B. (1986), Global Concave Minimization: A Bibliographic Survey, *SIAM Review* **28** (3), 367–379.
6. Pardalos, P. M. and Rosen, J. B. (1987), *Constrained Global Optimization: Algorithms and Applications*, Lecture Notes in Computer Science 268, Springer-Verlag, Berlin.
7. Pardalos, P. M. and Schnitger, G. (1988), Checking Local Optimality in Constrained Quadratic Programming is *NP*-hard, *Operations Research Letters* **7** (1), 33–35.
8. Sahni, S. (1974), Computationally Related Prolems, *SIAM J. Comput.* **3**, 262–279.
9. Vavasis, S. A. (1990), Quadratic Programming Is in NP, *Inf. Proc. Lett.* **36**, 73–77.