INFORMATICA, 1992, Vol.3, No.4, 474–496

ON THE SOLUTION OF INDEFINITE QUADRATIC PROBLEMS USING AN INTERIOR-POINT ALGORITHM

Chi-Geun HAN

Computer Science Department
The Pennsylvania State University
University Park, PA 16802

Panos M. PARDALOS

Department of Industrial and Systems Engineering
University of Florida, Gainesville, FL 32611

Yinyu YE[1]

Department of Management Sciences
The University of Iowa, Iowa City, Iowa 52242

Abstract. In this paper, we discuss computational aspects of an interior-point algorithm [1] for indefinite quadratic programming problems with box constraints. The algorithm finds a local minimizer by successively solving indefinite quadratic problems with an ellipsoid constraint. In addition, we present a sufficient condition for a local minimizer to be global, and we use this result to generate test problems with a known global solution. The proposed algorithm has been implemented on an IBM 3090 computer and tested on a variety of dense test problems, including problems with a known global optimizer.

Key words: nonconvex quadratic programming, interior point algorithms, computational testing.

1. Introduction. In this paper, we discuss computational aspects of an interior-point algorithm (IPA) proposed in [1] for indefinite quadratic programming (IQP) problems with box con-

[1] Research supported in part by NSF Grant DDM-8922636 and the Iowa Business School Summer Grant.

straints:

$$\min f(x) = \frac{1}{2}x^T Qx + c^T x, \tag{1.1}$$

$$\text{s.t.} \quad x \in P = \{x \in R^n \mid l \leqslant x \leqslant u\},$$

where $Q \in R^{n\times n}$ is an indefinite symmetric matrix and $x, c, l, u \in R^n$. Such problems arise quite naturally in a number of different applications. For example, every linear complementarity problem can be written in the above form [8].

Although polynomial time algorithms exist for the solution of convex quadratic programming (see [12]), the problem of minimizing a nonconvex quadratic function is computationally very difficult. It is well known that, from the complexity point of view, problem (1.1) belongs to the class of NP-complete problems ([17], [19], [21]). Even the problem of checking local optimality for a feasible point of a general quadratic problem is NP-hard ([15],[18]). There is a large literature for global optimization algorithms and heuristics for nonconvex quadratic problems [17]. Many of the proposed methods are based on local search techniques.

One of the first algorithms to compute local minima is the Frank and Wolfe algorithm ([6], [13]). However, this algorithm has a very slow rate of convergence. Forsgren et al. [5] used an inertia-controlling quadratic programming method for computing a local minimizer of nonconvex quadratic programs; see [7] for some major references. Coleman and Hulbert [1] proposed a direct active set method for solving definite and indefinite quadratic programs with box constraints. Other algorithms are described in [2] and [4]. Karmarkar [11] used an interior-point approach to solve approximately concave quadratic problems.

In this paper, we discuss computational aspects of an interior-point algorithm [1] for indefinite quadratic programming problems with box constraints. The algorithm also finds a local minimizer by successively solving indefinite quadratic problems with an ellipsoid constraint. In addition, we present a sufficient condition for a local minimizer to be global, and we use this result to generate test problems with a known global solution.

The paper is organized as follows. In Section 2, we describe the IPA algorithm that uses a procedure IQE to solve an indefinite quadratic problem with an ellipsoid constraint. The algorithm finds a point satisfying the first and second order optimality conditions under some nondegeneracy assumptions [1]. In Section 3 a sufficient condition for a local minimizer to be global is discussed. Using this sufficient condition, in Section 4, we propose a method to generate test problems with known global optimal solutions. In Section 5, we present preliminary computational results of our implementation of the IPA algorithm using these test problems and other random test problems.

2. An interior-point algorithm (IPA). In this section, we review the IPA algorithm that solves an indefinite quadratic problem subject to box constraints. For simplicity of analysis, let us transform (1.1) to

$$\min f(x) = \frac{1}{2}x^T Qx + c^T x, \tag{2.1}$$
$$\text{s.t.} \quad x \in S = \{x \in R^n \mid 0 \leqslant x \leqslant e\},$$

where e is a vector of all 1's. Note that this transformation is unnecessary in actual computation.

In the IPA algorithm, we solve a quadratic problem subject to an ellipsoid constraint at each step. It is known that the general quadratic problem with an ellipsoid constraint can be solved in polynomial time (e.g., [1]). First, we describe the main algorithm, IPA, and then present the procedure, IQE.

Initially, we have a starting point x^0 that is interior to the feasible region. We consider an ellipsoid E_1 with center x^0 that is inscribed in the feasible region. Then, we solve the quadratic problem with an ellipsoid E_1 constraint using the procedure IQE. Let x^1 be a solution of the problem. Again we consider an ellipsoid E_2 with center x^1 that is inscribed in the feasible region. By repeating this process, we compute a sequence of interior points x^0, x^1, x^2, ... After sufficiently many steps, we obtain an approximate local minimizer of problem (2.1). The following summarizes the algorithm IPA.

Algorithm IPA.

1. $k = 1$; $x^0 = 1/2e$; $D_1 = \mathrm{diag}(\frac{1}{2}, \ldots, \frac{1}{2})$.
2. Consider an ellipsoid $E_k \subseteq [0,1]^n$ with center x^{k-1} and radius $r < 1$ such that

$$E_k = \{x \mid \| D_k^{-1}(x - x^{k-1}) \| \leqslant r\},$$

where $\|.\|$ represents the l_2 norm throughout this paper.

3. Solve the following indefinite quadratic problems with an ellipsoid constraint using the procedure IQE (that will be discussed later):

$$\min f(x) = \frac{1}{2} x^T Q x + c^T x, \tag{2.2}$$

$$\text{s.t.} \quad x \in E_k.$$

Let x^k be a global minimizer of (2.2).

4. If x^k does not satisfy the stopping criterion (that will also be discussed later), then compute D_{k+1}, where $D_{k+1} = \mathrm{diag}(d_1, \ldots, d_n)$ and $d_i = \min\{x_i^k, 1 - x_i^k\}$, $i = 1, \ldots, n$, $k = k + 1$; goto Step 2;
Stop.

Now we describe the details of the procedure IQE for solving the problem (2.2), i.e., the indefinite quadratic problem with an ellipsoid constraint:

$$\min f(x) = \frac{1}{2} x^T Q x + c^T x, \tag{2.3}$$

$$\text{s.t.} \quad \| D^{-1}(x - x^{k-1}) \| \leqslant r,$$

where $D = D_k$, and Problem (2.3) has a global optimizer x^k **iff** (e.g., [14], [20])

$$(Q + \mu D^{-2})\Delta x = -(Q x^{k-1} + c), \tag{2.4}$$

$$\| D^{-1} \Delta x \| = r, \quad \mu > 0 \tag{2.5}$$

$$\text{and} \quad Q + \mu D^{-2} \quad \text{is positive semidefinite,} \tag{2.6}$$

where $\Delta x = x^k - x^{k-1}$ and μ is the multiplier of the ellipsoid constraint in (2.3). The equation (2.4) can be rewritten as

$$(DQD + \mu I)D^{-1}\Delta x = -D(Qx^{k-1} + c). \tag{2.7}$$

Let

$$\bar{Q} = DQD, \quad \Delta\bar{x} = D^{-1}\Delta x, \quad \text{and} \quad \bar{c} = D(Qx^{k-1} + c).$$

Then, equations (2.4) – (2.6) can be rewritten as

$$(\bar{Q} + \mu I)\Delta\bar{x} = -\bar{c}, \tag{2.8}$$

$$\| \Delta\bar{x} \| = r, \tag{2.9}$$

$$\mu \geqslant |\underline{\lambda}(\bar{Q})|, \tag{2.10}$$

where $\underline{\lambda}(\bar{Q})$ denotes the minimum eigenvalue of $\bar{Q}$. Since DQD is a congruent transformation and Q is an indefinite matrix, $\underline{\lambda}(\bar{Q})$ is negative. Note that for any symmetric matrix $\bar{Q}$, there exists an orthogonal and nonsingular matrix $U \in R^{n\times n}$ such that

$$U^T\bar{Q}U = \Lambda \quad \text{and} \quad UU^T = I,$$

where $\Lambda = \text{diag}(\lambda_i)$, $i = 1, \ldots, n$, and λ_is are the eigenvalues of $\bar{Q}$. Also note that the columns of U are eigenvectors of $\bar{Q}$. Since $\bar{Q} = U\Lambda U^T$, from (2.8)

$$U^T(U\Lambda U^T + \mu I)\Delta\bar{x} = -U^T\bar{c}. \tag{2.11}$$

Then, system (2.8) – (2.10) can be written as

$$(\Lambda + \mu I)U^T\Delta\bar{x} = -U^T\bar{c}, \tag{2.12}$$

$$\| U^T\Delta\bar{x} \| = r, \tag{2.13}$$

$$\mu \geqslant |\underline{\lambda}(\Lambda)| = |\underline{\lambda}(\bar{Q})|. \tag{2.14}$$

Let $\Delta\bar{\bar{x}} = U^T\Delta\bar{x}$ and $\bar{\bar{c}} = U^T\bar{c}$. Then we have

$$(\Lambda + \mu I)\Delta\bar{\bar{x}} = -\bar{\bar{c}}, \tag{2.15}$$

$$\| \Delta\bar{\bar{x}} \| = r, \tag{2.16}$$

$$\mu \geqslant |\underline{\lambda}(\Lambda)|. \tag{2.17}$$

If we solve the system (2.15) – (2.17) and get $\Delta\bar{\bar{x}}$, then from the relations $\Delta x = D\Delta\bar{x} = DU\Delta\bar{\bar{x}}$, we obtain $x^k = x^{k-1} + \Delta x$ that is a global solution of (2.2) or (2.3). It seems that the system (2.15) – (2.17) is simpler than the system (2.4) – (2.6) since we have only a diagonal matrix Λ in (2.15) – (2.17).

We now discuss how to solve the system (2.15) – (2.17). Assume the entries of Λ and $\bar{\bar{c}}$ are permuted so that λ_1 is the smallest eigenvalue of $\bar{Q}$ (denoted as $\underline{\lambda}(\bar{Q})$ above). We assume further that all other eigenvalues are strictly greater than λ_1 (this assumption can be removed at the expense of a slightly more complicated procedure). Note that with these assumptions, the matrix $\Lambda + |\lambda_1|I$ is singular and positive semidefinite, having as its range exactly the set of vectors whose first component is zero. Thus, consider the following two cases.

Case 1: $\bar{\bar{c}}_1 \neq 0$ ($\bar{\bar{c}}$ is not in the range of $\Lambda + |\lambda_1|I$). In this case we search for a μ that satisfies

$$\sum_{i=1}^{n} \left(\frac{-\bar{\bar{c}}_i}{\mu + \lambda_i} \right)^2 = r^2$$

by using a binary search for $|\lambda_1| + \frac{|\bar{\bar{c}}_1|}{r} < \mu \leqslant |\lambda_1| + \frac{\sqrt{n}\max_i |\bar{\bar{c}}_i|}{r}$.

Case 2: $\bar{\bar{c}}_1 = 0$ ($\bar{\bar{c}}$ is in the range of $\Lambda + |\lambda_1|I$). In this case we set $\mu = |\lambda_1|$ and compute $s = \sum_{i=2}^{n} (\Delta\bar{\bar{x}}_i)^2$, where $\Delta\bar{\bar{x}}$ is the minimum-norm solution of (2.15), i.e., $\Delta\bar{\bar{x}}_1 = 0$ and $\Delta\bar{\bar{x}}_i = \frac{\bar{\bar{c}}_i}{\mu + \lambda_i}$, for $i = 2, ..., m$. If $s \leqslant r^2$, reset $\Delta\bar{\bar{x}}_1 = \sqrt{r^2 - s}$; otherwise ($s > r^2$) find a new μ that satisfies

$$\sum_{i=2}^{n} \left(\frac{-\bar{\bar{c}}_i}{\mu + \lambda_i} \right)^2 = r^2$$

using a binary search for $|\lambda_1| < \mu \leqslant |\lambda_1| + \frac{\sqrt{n}\max_i |\bar{\bar{c}}_i|}{r}$, and then reset $\Delta\bar{\bar{x}}_i = \frac{\bar{\bar{c}}_i}{\mu + \lambda_i}$, for $i = 2, ..., m$.

Now from $\Delta\bar{\bar{x}}$, we obtain $\Delta x = D_k U \Delta\bar{\bar{x}}$. Thus $x^k = x^{k-1} + \Delta x$ is a global minimizer of (2.2) and

$$f(x^{k-1}) - f(x^k) = \frac{1}{2}\mu r^2 + \frac{1}{2}\Delta x^T (Q + \mu D_k^{-2}) \Delta x > 0.$$

One can verify that the above procedure is a polynomial-time algorithm. In practice, the radius r can be flexibly chosen as long as $x^k = x^{k-1} + \Delta x$ is an interior feasible point. This will result in a larger step to obtain x^k.

The algorithm IPA has been implemented and tested on a variety of test problems (Section 5). The purpose of these preliminary tests is to observe the convergence behavior and the iteration performance of the algorithm IPA as a whole. Our procedure IQE used in each iteration needs to compute all the eigenvalues and eigenvectors of a symmetric matrix – which may not be very practical for large scale problems. Some practically efficient procedures have been discussed in [14], which can be used to replace the procedure IQE in the algorithm IPA for solving large scale problems.

3. A sufficient condition of global minimum. When the matrix Q is indefinite, problem (1.1) may have many local minima which differ from the global solution. However, every global (local) minimizer of (1.1) occurs at a boundary point of P, not necessarily a vertex [17]. More generally [9], the following result characterizes the optimal solution x^* of (1.1): if Q has k negative eigenvalues counting multiplicities, then the dimension of the space spanned by the gradients to the active constraints is at least k. Hence, at least k of the constraints are active at x^*. Note that, if Q is negative definite, then every global (local) minimizer is a vertex of P.

From the complexity point of view, the problem of checking if a given feasible point is optimal is NP-complete ([15],[18]). However, complexity results of this nature characterize worst-case problem instances.

Next, we provide a sufficient condition (initially proposed by Neumaier [16] for general quadratic problems) for checking if a local minimizer is global. This sufficient condition is used (Section 4) to construct test problems with a known global solution.

Let $\bar{x}$ be a local minimizer of (1.1). The first order necessary conditions are given by

$$Q\bar{x} + c + \mu - \lambda = 0, \tag{3.1}$$

$$\mu_i(\bar{x}_i - u_i) = 0 \quad \text{and} \quad \lambda_i(l_i - \bar{x}_i) = 0, \quad i = 1, \ldots, n, \tag{3.2}$$

$$\bar{x} - u \leqslant 0 \quad \text{and} \quad l - \bar{x} \leqslant 0, \tag{3.3}$$

$$\mu \geqslant 0 \quad \text{and} \quad \lambda \geqslant 0. \tag{3.4}$$

From (3.2), (3.3), and (3.4), we have

$$(\lambda_i - \mu_i)(l_i - \bar{x}_i) = -\mu_i(l_i - \bar{x}_i) \geqslant 0, \tag{3.5}$$

$$(\lambda_i - \mu_i)(u_i - \bar{x}_i) = \lambda_i(u_i - \bar{x}_i) \geqslant 0, \tag{3.6}$$

for $i = 1, \ldots, n$. Let $\lambda_i - \mu_i = y_i$ for $i = 1, \ldots, n$. From (3.2), (3.5), and (3.6)

$$\inf\{y_i(l_i - \bar{x}_i), y_i(u_i - \bar{x}_i)\} = 0, \quad i = 1, \ldots, n.$$

The equation (3.1) can be rewritten as

$$Q\bar{x} + c = y. \tag{3.7}$$

We make the following nondegeneracy assumption:
If $\bar{x}_i = u_i$ or $\bar{x}_i = l_i$, then $y_i \neq 0$, $i = 1, \ldots, n$. Hence

$$l_i < \bar{x}_i < u_i \Leftrightarrow y_i = 0 \quad \text{for} \quad i = 1, \ldots, n.$$

Let m be the number of indices i for which $y_i \neq 0$. Without loss of generality, we may assume that

$$\begin{aligned} &\bar{x}_i = u_i \quad \text{or} \quad l_i, \quad i = 1, \ldots, m, \\ &l_i < \bar{x}_i < u_i, \quad i = m+1, \ldots, n, \end{aligned}$$

where $1 \leqslant m \leqslant n$.

We are going to prove that the matrix $M = Q + \begin{pmatrix} I_m \\ 0 \end{pmatrix} H(I_m\ 0)$ is positive semidefinite for some matrix $H \in R^{m \times m}$. First, if $m = n$

(i.e., the objective is concave), then by choosing an H such that $Q + H$ is diagonally dominant, we can easily prove that M is a positive semidefinite matrix. For the second case (if $m < n$), we need the following derivation.

Let

$$Q = \begin{pmatrix} Q_A & Q_B \\ Q_B^T & Q_C \end{pmatrix},$$

where $Q_A \in R^{m\times m}$, $Q_B \in R^{m\times(n-m)}$, and $Q_C \in R^{(n-m)\times(n-m)}$. We make the assumption that the submatrix Q_C is positive definite. Define $Z \in R^{n\times(n-m)}$ and $V \in R^{n\times m}$ such that

$$Z = \begin{pmatrix} 0 \\ I_{n-m} \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} I_m \\ 0 \end{pmatrix},$$

where I is an identity matrix. Note that the columns of Z form a basis of the null space of V^T. From the second order optimality conditions,

$$Z^T Q Z = Q_C,$$

which is a positive semidefinite matrix. Hence, we can find $L \in R^{(n-m)\times(n-m)}$ such that

$$Q_C = LL^T.$$

In addition, define matrices $E \in R^{m\times(n-m)}$ and $F \in R^{m\times m}$ such that

$$E = (V^T Q Z)L^{-T} = Q_B L^{-T} \text{ and}$$

$$F = V^T Q V - EE^T = Q_A - Q_B {Q_C}^{-1} {Q_B}^T.$$

Note that F is the Schur complement of Q_C in the matrix

$$\begin{pmatrix} Q_C & Q_B^T \\ Q_B & Q_A \end{pmatrix}.$$

Next, construct a matrix $H = \operatorname{diag}(h_i)$, $h_i \geqslant 0$, $i = 1, \ldots, m$ so that $F + H$ is positive semidefinite. Let $F + H = NN^T$. Using the above information, we are going to show that the following matrix

$$\begin{pmatrix} Q_A + H & Q_B \\ Q_B^T & Q_C \end{pmatrix}$$

is positive semidefinite. Note that

$$
\begin{aligned}
(Z\ V)^T(Q + \begin{pmatrix} I_m \\ 0 \end{pmatrix} H(I_m\ 0))(Z\ V) \\
&= \begin{pmatrix} Z^TQZ & Z^TQV \\ V^TQZ & V^TQV \end{pmatrix} + \begin{pmatrix} 0 \\ I_m \end{pmatrix} H(0\ I_m) \\
&= \begin{pmatrix} Z^TQZ & Z^TQV \\ V^TQZ & V^TQV + H \end{pmatrix} \\
&= \begin{pmatrix} LL^T & LE^T \\ EL^T & EE^T + NN^T \end{pmatrix} \\
&= \begin{pmatrix} L & 0 \\ E & N \end{pmatrix} \begin{pmatrix} L^T & E^T \\ 0 & N^T \end{pmatrix} \\
&\equiv BB^T.
\end{aligned}
$$

Hence the matrix

$$(Z\ V)^T(Q + \begin{pmatrix} I_m \\ 0 \end{pmatrix} H(I_m\ 0))(Z\ V)$$

is positive semidefinite and since $(Z\ V)$ is nonsingular, the matrix

$$Q + \begin{pmatrix} I_m \\ 0 \end{pmatrix} H(I_m\ 0) = \begin{pmatrix} Q_A + H & Q_B \\ Q_B^T & Q_C \end{pmatrix}$$

is positive semidefinite.

Using the following theorem, we can check if a nondegenerate local minimizer of (1.1) is a global minimizer.

Theorem 3.1. *Let $\bar{x}$ be a nondegenerate local minimizer of (1.1) and*

$$a_i = \frac{H_{ii}}{2y_i}, \quad i = 1, \ldots, m, \tag{3.8}$$

where H and y are defined as above. Then for all $x \in P$,

$$(Q\bar{x} + c)^T(x - \bar{x}) \geqslant 0 \tag{3.9}$$

and

$$f(x) - f(\bar{x}) \geqslant (Q\bar{x} + c)^T(x - \bar{x})(1 - \sum_{i=1}^{m} a_i(x_i - \bar{x}_i)). \tag{3.10}$$

If

$$\alpha = \sum_{i=1}^{m} | a_i(u_i - l_i) | \leqslant 1,$$

then $\bar{x}$ is a global minimizer of (1.1).
If

$$\alpha = \sum_{i=1}^{m} | a_i(u_i - l_i) | < 1,$$

then $\bar{x}$ is the unique global minimizer of (1.1).

Proof. Note that for any $x \in P$,

$$f(x) = f(\bar{x}) + (Q\bar{x} + c)^T(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T Q(x - \bar{x}) \tag{3.11}$$

and from (3.7),

$$(Q\bar{x} + c)^T(x - \bar{x}) = y^T(x - \bar{x}). \tag{3.12}$$

If

$$Y = \operatorname{diag}(y_i),\ i = 1, \ldots, m,\ e = (1, \ldots, 1)^T \in R^m, \text{ and}$$

$$s = Y(I_m\ 0)(x - \bar{x}) \in R^m, \tag{3.13}$$

then

$$(Q\bar{x} + c)^T(x - \bar{x}) = e^T s. \tag{3.14}$$

Note that

$$s_i = y_i(x_i - \bar{x}_i) \geqslant \inf\{y_i(l_i - \bar{x}_i), y_i(u_i - \bar{x}_i)\} = 0, \quad i = 1, \ldots, m$$

and

$$s_i = 0 \quad \text{iff} \quad x_i - \bar{x}_i = 0,\ i = 1, \ldots, m.$$

Hence, from (3.14),

$$(Q\bar{x} + c)^T(x - \bar{x}) \geqslant 0 \quad \text{with equality} \quad \text{iff } x_i = \bar{x}_i,\ i = 1, \ldots, m.$$

Since H and

$$Q + \begin{pmatrix} I_m \\ 0 \end{pmatrix} H(I_m\ 0)$$

are positive semidefinite, we have

$$\begin{aligned} 0 &\leqslant \frac{1}{2}(x-\bar{x})^T\left(Q+\begin{pmatrix} I_m \\ 0 \end{pmatrix} H(I_m\ 0)\right)(x-\bar{x}) \\ &= \frac{1}{2}x^TQx+\frac{1}{2}\bar{x}^TQx+\frac{1}{2}(x-\bar{x})^T\begin{pmatrix} I_m \\ 0 \end{pmatrix} H(I_m\ 0)(x-\bar{x}) \\ &= f(x)-f(\bar{x})-(Q\bar{x}+c)^T(x-\bar{x})+\frac{1}{2}s^TY^{-1}HY^{-1}s \\ &\leqslant f(x)-f(\bar{x})-(Q\bar{x}+c)^T(x-\bar{x})+\frac{1}{2}(e^Ts)(v^Ts), \end{aligned}$$

where

$$v_i=(Y^{-1}HY^{-1})_{ii}=2\frac{a_i}{y_i}, \quad i=1,\ldots,m.$$

Since

$$v_is_i=2a_i(x_i-\bar{x}_i),\ i=1,\ldots,m \quad \text{and} \quad v^Ts=2\sum_{i=1}^m a_i(x_i-\bar{x}_i),$$

$$\begin{aligned} &f(x)-f(\bar{x})-(Q\bar{x}+c)^T(x-\bar{x})+\frac{1}{2}(e^Ts)(v^Ts) \\ &\quad = f(x)-f(\bar{x})-(Q\bar{x}+c)^T(x-\bar{x})(1-\sum_{i=1}^m a_i(x_i-\bar{x}_i)). \end{aligned}$$

Therefore,

$$f(x)-f(\bar{x})\geqslant(Q\bar{x}+c)^T(x-\bar{x})(1-\sum_{i=1}^m a_i(x_i-\bar{x}_i)). \tag{3.15}$$

Note that

$$\max\Big\{\sum_{i=1}^m a_i(x_i-\bar{x}_i) \mid l_i\leqslant x_i\leqslant u_i, \quad i=1,\ldots,m\Big\}\leqslant\sum_{i=1}^m \mid a_i(u_i-l_i)\mid.$$

It follows that if

$$\alpha=\sum_{i=1}^m \mid a_i(u_i-l_i)\mid\leqslant 1, \quad \text{then} \quad f(x)-f(\bar{x})\geqslant 0.$$

Therefore $\bar{x}$ is a global minimizer of (1.1).

Assume that $\alpha < 1$ and $x' \neq \bar{x}$ is another global minimizer of (1.1). From (3.10), $(Q\bar{x} + c)^T(x' - \bar{x}) = 0$. Since (3.9) holds with equality only when $x'_i = \bar{x}_i$, $i = 1, \dots, m$, if $m = n$, then $x' = \bar{x}$. Otherwise we can set $x' - \bar{x} = Zb$ for some nonzero vector $b \in R^n$. We know that $b^T Z^T Q Z b$ is positive since $Z^T Q Z$ is positive definite. But this is a contradiction to $b^T Z^T Q Z b = 0$ in (3.11). Therefore, if $\alpha < 1$, then $\bar{x}$ is unique. Q.E.D.

4. Test problem generation. In this section, we describe a method to generate indefinite quadratic problems with box constraints (1.1) and a known global solution, using the sufficient condition discussed in Section 3. Initially, we choose a point x^* and a matrix Q. By making c satisfy the sufficient condition, we can generate an instance of problem (2.1) that has the optimum solution x^*.

We can construct the test problems according to the following steps:

1. Choose an $x^* = (x_1^*, \dots, x_m^*, x_{m+1}^*, \dots, x_n^*)$ such that

$$x_i^* = \begin{cases} l_i \text{ or } u_i, & i = 1, \dots, m, \\ l_i < x_i^* < u_i, & i = m+1, \dots, n, \end{cases}$$

   where $1 \leqslant m < n$.

2. Generate a matrix $Q \in R^{n \times n}$ such that

$$Q = W \Lambda W^T,$$

   where $W = \begin{pmatrix} U_1 & A \\ 0 & U_2 \end{pmatrix} \in R^{n \times n}$,
   $\Lambda = \text{diag}(-\lambda_1, \dots, -\lambda_m, \lambda_{m+1}, \dots, \lambda_n)$,
   $U_1 \in R^{m \times m}$, $U_2 \in R^{(n-m) \times (n-m)}$ are orthogonal matrices,
   $A \in R^{m \times (n-m)}$ can be any matrix, and $\lambda_i > 0$ for$1 \leqslant i \leqslant n$.
   Partition Q as

$$Q = \begin{pmatrix} Q_A & Q_B \\ Q_B{}^T & Q_C \end{pmatrix},$$

   where $Q_A \in R^{m \times m}$, $Q_B \in R^{m \times (n-m)}$, and $Q_C \in R^{(n-m) \times (n-m)}$.

3. Let

$$F = Q_A - Q_B Q_C{}^{-1} Q_B{}^T$$
$$= (F + H) - H,$$

where $H = \text{diag}(h_i)$, $h_i > 0$, $i = 1, \ldots, m$ and $F + H$ is a positive semidefinite matrix.

4. The sufficient condition for x^* to be a global minimizer of (1.1) is given by

$$\max \{(a, 0)^T (x - x^*) | x \in P\} \leqslant 1 \qquad (4.1)$$

with $a \in R^m$. By considering the box constraints, we choose a such that

$$a_i = \begin{cases} t_i, & \text{if } x_i^* = l_i, \\ -t_i & \text{o.w. } (x_i^* = u_i), \end{cases}$$

where t_i, $i = 1, \ldots, m$, is randomly chosen from $(0, \frac{1}{m(u_i - l_i)})$.

5. Compute $y \in R^n$ as follows:

$$y_i = \begin{cases} \frac{h_i}{2a_i}, & i = 1, \ldots, m, \\ 0, & i = m + 1, \ldots, n. \end{cases}$$

6. From $Qx^* + c = y$, compute $c \in R^n$. Now x^* is a global minimizer of (1.1).

The following simple example illustrates the steps to generate the test problems.

0. n=3, m=2, l_i=0, u_i=1, for $i = 1, 2, 3$.
1. $x^* = (1, 0, 1/4)^T$.
2.

$$Q = \begin{pmatrix} -8 & 4 & 0 \\ 4 & 2 & -8 \\ 0 & -8 & 16 \end{pmatrix}, \quad Q_A = \begin{pmatrix} -8 & 4 \\ 4 & 2 \end{pmatrix}, \quad Q_B = \begin{pmatrix} 0 \\ -8 \end{pmatrix}, \quad Q_C = (16).$$

Note that Q has eigenvalues $\lambda = (-9.73, 0, 19.73)^T$.

3.

$$F = Q_A - Q_B Q_C^{-1} Q_B^T$$
$$= \begin{pmatrix} -8 & 4 \\ 4 & -2 \end{pmatrix}.$$

Let $h = (12,6)^T$ so that $F + H$ is PSD.

4. $a = (-1/4, 1/4)^T$.
5. $y = (-24, 12, 0)^T$.
6. $c = y - Qx^* = (-16, 10, -4)^T$. Then x^* is a global minimizer of

$$\min \frac{1}{2}x^T Q x + c^T x,$$

$$\text{s.t.} \quad 0 \leqslant x \leqslant 1.$$

5. Computational results. We implemented the algorithm IPA on an IBM 3090-600S computer with vector facilities using the VS Fortran compiler and double precision accuracy for real variables. The ESSL subroutines were used for matrix and vector operations, random number generation, and the eigensystem solving. The condition $\| x^k - x^{k-1} \| < 10^{-4}$ was used for stopping criterion of the main algorithm IPA. Since a fractional number of components of a local minimizer hit the upper bound 1, $\|x^k\| > 1$ in all our numerical examples. Thus, this (absolute) convergence condition is actually harder to meet than the relative convergence criterion. We used binary search for the parameter μ in IQE and the binary search was terminated when "upper_bound - lower_bound" $< 10^{-8}$ was satisfied. From the preliminary computational results, we found that when the parameter r is close to 1, the number of iterations needed to find a stationary point becomes small. Therefore, we set $r = 0.99$ in our implementation. All averages were obtained from 5 problems and CPU times are given in seconds.

We have two parameters for generating the test problems: the number of active constraints (*nac*) at the optimum solution and the number of negative eigenvalues (*neg*) of matrix Q. Note that $neg \leqslant nac$. Two types of dense test problems are solved: test problems with a known solution (type I) and random problems (type II). For generating type I test problems we have the following:

1. The optimum solution x^* is chosen randomly such that $x_i^* \in (0,1)$, for $i = 1, \ldots, n$. Half of the m active constraints are active at the upper bound.

2. The orthogonal matrices, U_1 and U_2, are Householder matrices of random vectors $v \in R^{m \times m}$ and $w \in R^{(n-m)\times(n-m)}$, respectively. Each element of v and w are chosen randomly from (0,1).
3. The eigenvalues $\lambda_i, i = 1, \ldots, n$, of Λ are chosen randomly so that $1 < |\lambda_i| < 2,\ i = 1, \ldots, n$.
4. The submatrix $A \in R^{m\times(n-m)}$ of W has elements $a_{ij},\ i = 1, \ldots, m$, $j = 1, \ldots, n-m$, chosen randomly from (0,1).
5. Choose the diagonal matrix H such that $F + H$ is diagonally dominant.

Table 1 shows the computational results obtained by solving the test problems generated by the method discussed in Section 4. From Table 1, it is observed that IPA takes more steps to converge to a stationary point as the number of active constraints increases.

Table 1. IPA on varying *nac* ($n = 100,\ neq = nac$)

nac	Avg. Itr.	Avg. CPU time
10	40.8	10.4
30	68.8	15.2
50	96.4	19.1
70	100.4	20.7
90	118.8	28.9

The relationship between the number of negative eigenvalues of matrix Q and the complexity of the test problems with a known solution is shown in Table 2. This shows that the number of iterations is not dependent on the parameter *neg*. It is interesting to note that the algorithm always finds a global minimizer for these type of test problems (although many local minima exist).

Next, we solve randomly generated indefinite quadratic problems (type II). The matrix Q is generated by $Q = U^T \Lambda U$, where U is an orthogonal matrix (Householder matrix of a random vector v, where $0 < v_i < 1,\ i = 1, \ldots, n$) and Λ is a diagonal matrix whose diagonal elements are chosen randomly from

$$-2 < \lambda_i < -1, \quad i = 1, \ldots, neg,$$

$$1 < \lambda_i < 2, \quad i = neg + 1, \ldots, n.$$

The components of c are randomly chosen from $(0, 1)$.

Table 2. IPA on varying neg ($n = 100$, $nac = 60$)

neg	Avg. Itr.	Avg. CPU time
10	94.6	19.1
20	95.0	19.1
30	105.8	21.1
40	97.4	19.4
50	94.4	19.1
60	104.4	20.9

Table 3 shows the computational results for type II test problems. Note that all test problems we solve are dense.

Table 3. IPA for randomly generated problems ($n = 100$, $neg = nac$)

nac	Avg. Itr.	CPU time
10	88.6	19.3
30	94.4	20.7
50	97.2	22.1
70	91.4	22.4
90	86.8	22.0

In addition, we implement the Frank-Wolfe method for solving (2.1). At step k of the Frank-Wolfe method, we solve a linear programming problem of the form

$$\begin{aligned} &\min \nabla f(x^k)^T x, \\ &\text{s.t.} \quad 0 \leqslant x \leqslant e. \end{aligned}$$

Note that the linear programming problem can be easily solved. Let x_{dir} be the solution of the linear programming problem. Using a line search method in $(x^k, x_{dir}]$, we find x^{k+1} that minimizes $f(x)$. From our computational experience, we also find that the Frank-Wolfe method does not generate a sequence of solutions that satisfies the stopping criterion $\| x^k - x^{k-1} \| < 10^{-4}$. Hence, we compare the objective function values of the same problem at each optimum solution obtained by the Frank-Wolfe method and Algorithm IPA. Let x^*_{IPA} and x^*_{FR} be the final solutions of Algorithm IPA and the Frank-Wolfe method, respectively. The Frank-Wolfe method terminates when $f(x^*_{IPA}) > f(x^*_{FR})$ or the number of iterations is greater than 5000.

Tables 4 and 5 show the computational results for the type I and type II test problems, respectively. For each instance, we solved one type I test problem and one type II test problem ($n = 100$). From Table 4, we can observe that when the number of active constraints at the optimum solution is small, algorithm IPA converges faster than Frank-Wolfe method. Note that the Frank-Wolfe method is very efficient when the solution is a vertex (or close to a vertex) of the feasible domain. Table 5 shows that in 6 out of 10 cases, the Frank-Wolfe method fails to find a better solution than Algorithm IPA does.

Table 4. IPA and Frank-Wolfe method (type I, $neq = nac$)

nac	IPA (CPU/Itr.)	Frank-Wolfe (CPU/Itr.)
10	10.3/40	45.1/5000
20	13.9/58	42.0/4757
30	15.8/71	36.3/4223
40	18.3/89	30.6/3661
50	19.0/96	20.5/2570
60	21.6/105	6.9/853
70	23.9/114	2.2/291
80	22.7/123	0.7/101
90	21.0/84	0.03/6

Table 6 shows the computational results for the randomly generated problems with various problem sizes. For this table, we used two stopping criteria: $\| x^k - x^{k-1} \| < 10^{-4}$ or $|f(x^k) - f(x^{k-1})|/|f(x^k)| < 10^{-6}$ since the former (absolute) convergence criterion becomes difficult to satisfy as n increases due to the reason mentioned at the begining of this section.

Table 5. IPA and Frank-Wolfe method (type II, $neq = nac$)

nac	IPA (CPU/Itr.)	Frank-Wolfe (CPU/Itr.)
10	20.9/94	20.3/2472
20	22.6/99	41.0/5000
30	19.3/84	7.3/932
40	19.5/82	40.8/5000
50	23.9/102	9.0/1179
60	23.6/102	40.5/5000
70	22.6/93	38.4/5000
80	21.4/85	38.1/5000
90	21.7/84	36.8/5000

Table 6. IPA on randomly generated problems ($neq = nac = n/2$)

n	Avg. Itr.	Avg. CPU time
100	94.6	21.9
200	136.8	196.8
300	166.2	745.0

Table 7 shows the improvement from the function value at the starting point to the final local minimizer for the set of 5 problems of dimension $n = 300$.

Table 7. Initial and function values for randomly generated problems ($n = 300$)

Problem	Initial Fun. Value	Obtained Fun. Value
1	−27.2	−149.7
2	−43.5	−189.1
3	−45.4	−196.9
4	−31.8	−156.8
5	−53.7	−233.1

6. Concluding remarks. In this paper, we discussed computational aspects of an interior-point algorithm that finds a local minimizer of indefinite quadratic problems with box constraints, by successively solving indefinite quadratic problems over an ellipsoid. The preliminary computational results (with dense problems) show that the difficulty of the problem is not dependent on the number of negative eigenvalues of Q. However, the algorithm needs more steps to converge as the number of active constraints increases. Although the algorithm is not guaranteed to find a global solution of the problem, it was observed that the algorithm found the global minimizer in many cases. Further research is needed to apply the algorithm to solve large sparse problems.

Using a local search technique, it is an interesting problem to check the "quality" of the local minimizer computed. Suppose that with different starting points, we compute $v_1, \ldots, v_N$ local minimizers (or stationary points) and take $f(v) = \min\{f(v_i) | 1 \leqslant i \leqslant N\}$ as an approximation of the global optimizer of $f(x)$ over P. Space covering techniques can be used to calculate the "quality" of $f(v)$. Let L be the Lipschitz constant of $f(x)$. Consider the spheres S_i with center v_i and radius $r_i = (f(v_i) - f(v))/L$, $i = 1, \ldots, N$. If $\cup_{i=1}^{N} S_i \supseteq P$, then $f(v)$ is the global minimizer. If not let $r_i^{\epsilon} = (f(v_i) - f(v) + \epsilon)/L, \epsilon > 0$, and let S_i^{ϵ} be the corresponding spheres. If $\cup_{i=1}^{N} S_i^{\epsilon} \supseteq P$, then $f(v) - f(v^*) \leqslant \epsilon$, where $f(v^*)$ is the global minimum. Such techniques are discussed in detail in [3].

REFERENCES

[1] Coleman, T.F., and L.A. Hulbert (1989). A direct active set algorithm for large sparse quadratic programs with simple bounds. *Mathematical Programming,* 45, 373–406.

[2] Conn, A.R., N.I.M. Gould, and P.L. Toint. (1988). Testing a class of methods for solving minimization problems with simple bounds on the variables. *Mathematics of Computation,* 50(182), 399–430 .

[3] Evtushenko, G.Y. (1985). *Numerical Optimization Techniques.* Optimization

Software Inc., Publications Division N.Y.

[4] Fletcher, R., and M.P. Jackson (1974). Minimization of a quadratic function of many variables subject only to lower and upper bounds. *J. Inst. Maths Applics*, **14**, 159–174.

[5] Forsgren, A.L., P.E. Gill, and W. Murray. (1990). On the identification of local minimizers in inertia-controlling methods for quadratic programming. To appear in *SIAM J. on Matrix Analysis and applications,*

[6] Frank, M., and P. Wolfe (1956). An algorithm for quadratic programming. *Naval Res. Logist. Quart*, **3**, 95–110.

[7] Gill, P.E., W. Murray, M.A. Saunders, and M.H. Wright (1991). Inertia-controlling methods for general quadratic programming. *SIAM Review*, **33**(1), 1–36.

[8] Gupta, S., and P.M. Pardalos (1988). A note on a quadratic formulation for linear complementarity problems. *JOTA*, **57**(1), 197–202.

[9] Hager, W.W., P.M. Pardalos, I.M. Roussos, and H.D. Sahinoglou (1991). Active constraints, indefinite quadratic test problems, and complexity. *JOTA*, **68**(3), 499–511.

[10] Han, C.G., P.M. Pardalos, and Y. Ye (1990). Computational aspects of an interior point algorithm for quadratic programming problems with box constraints. In T.F. Coleman and Y. Li (Eds.), *Large-Scale Numerical Optimization, SIAM*, 92–112.

[11] Karmarkar, N. (1990). An interior-point approach to NP-complete problems - Part I. *Contemporary Mathematics*, **114**, 297–308.

[12] Kozlov, M.K., S.P. Tarasov, and L.G. Khachian (1979). Polynomial solvability of convex quadratic programming. *Soviet Math. Dokl.*, **20**, 1108–1111.

[13] Minoux, M. (1986). *Mathematical Programming, Theory and Algorithms.* New York, Wiley.

[14] Moré, J.J., and D.C. Sorensen (1983). Computing a trust region step. *SIAM J. Sci. Stat. Comput.*, **4**, 553–572.

[15] Murty, K.G., and S.N. Kabadi (1987). Some NP-complete problems in quadratic. *Mathematical Programming*, **39**, 117–129.

[16] Neumaier, A. (1989). *An optimality Criterion for global Quadratic Optimization.* Technical Report, Inst. f. Angewandte Mathematik, Universität Freiburg, D-7800 Freiburg, West Germany.

[17] Pardalos, P.M., and J.B. Rosen (1987). *Constrained Global Optimization:*

, *Algorithms and Applications*. Springer-Verlag, Lecture Notes in Computer Science 268.

[18] Pardalos, P.M., and G. Schnitger (1988). Checking local optimality in constrained quadratic programming is NP-hard. *Operations Research Letters*, 7(1), 33–35.

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

[20] Sorensen, D.C. (1982). Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, **19**, 409–426.

[21] Vavasis, S.A. (1991). *Nonlinear Optimization: Complexity Issues*. Oxford University Press.

[22] Ye, Y., and E. Tse (1989). An extension of Karmarkar's projective algorithm for convex quadratic programming. *Mathematical Programming*, **44**, 157–179.

[23] Ye, Y. (1988). On the affine scaling algorithm for nonconvex quadratic programming. To appear in *Mathematical Programming*.

Received September 1992

C-G. Han received his Ph.D. degree in Computer Science from the Pennsylvania State University in 1991. He is currently an assistant professor at the Kyung Hee University, Korea. His research interests include computational mathematical programming.

P. M. Pardalos was born in Greece in 1954. He received his B. S. degree in Mathematics from Athens University and his Ph. D. degree in Computer Science from the University of Minnesota. Currently he has a visiting associate professor at the University of Florida and a professor at the Technical University of Crete, Greece. His research interests include global and combinatorial optimization, parallel algorithms and software development. He is the author and editor of several books in global optimization and serves as a managing editor of the Journal of Global Optimization.

Y. Ye was born in China in 1948. He received his B. S. degree from Huazhong University, and his M. S. and Ph. D. degrees from Stanford University. He is an associate professor at the University of Iowa since 1988. He has published extensively in interior point methods and mathematical programming.