New Infeasible Interior Point Algorithm Based on Monomial Method

Yi-Chih Hsieh and Dennis L. Bricker
Department of Industrial Engineering
The University of Iowa, Iowa City, IA 52242
USA
(January, 1995)

Scope and Purpose

The Karush-Kuhn-Tucker (KKT) equations provide both necessary and sufficient conditions for optimality of convex linearly-constrained quadratic programming problems. These equations consist of both linear equations (the primal and dual feasibility constraints) and nonlinear equations (the complementary slackness conditions, specifying that the product of each complementary pair of variables is zero). While for many years complementary pivoting algorithms to solve the KKT equations were the state of the art for quadratic programming problems, more recently there has been much progress in the development of interior-point methods. Path-following interior-point algorithms proceed by relaxing the complementarity constraints of the KKT equations, specifying that each complementary product equals a positive parameter $\mu$, which is successively reduced at each iteration. The solutions of the relaxed KKT equations constitute the "central path", parameterized by $\mu$, converging to the optimum as $\mu$ approaches zero. For each $\mu$, these equations are typically "solved" by one iteration of Newton's method, in which the nonlinear (complementarity) equations are approximated by linear equations and the
resulting system of linear equations are then solved. The resulting sequence of points satisfy the primal and dual feasibility constraints exactly, but the (relaxed) complementarity conditions only approximately. An alternate approach, the so-called "monomial method", treats the complementarity constraints directly, and approximates the linear equations by monomial equations in order to obtain a system of equations which, after a logarithmic transformation, yields a linear system of equations. The sequence of points which results from solving these log-linear equations therefore satisfy the (relaxed) complementarity equations exactly, but the primal and dual feasibility constraints only approximately. This paper discusses this new path-following algorithm for quadratic programming, and evaluates its performance by presenting the results of some numerical experiments.

Abstract

We propose a new infeasible path-following algorithm for convex linearly-constrained quadratic programming problem. This algorithm utilizes the monomial method rather than Newton's method for solving the KKT equations at each iteration. As a result, the sequence of iterates generated by this new algorithm is infeasible in the primal and dual linear constraints, but, unlike the sequence of iterates generated by other path-following algorithms, does satisfy the complementarity equations. Performance of this new algorithm is demonstrated by the results of solving QP problems (both separable and nonseparable) which are constructed so as to have known optimal solutions. Additionally, results of solving continuous quadratic knapsack problems indicate that for problems of a given size, the computational time of this new algorithm is less variable than other algorithms.
Keywords: Infeasible Interior Point Methods, Monomial Method, Complementarity Surface, Quadratic Programming, Continuous Knapsack Problems.
BIOGRAPHICAL INFORMATION

Yi-Chih Hsieh

Yi-Chih Hsieh is now a Ph.D. candidate in the Department of Industrial Engineering, The University of Iowa, Iowa City. He holds a B.B.A. in Industrial Management Science from National Cheng Kung University ('88), Taiwan, and an M.S. in Applied Mathematics in National Chung Hsing University ('92), Taiwan. His research interests include mathematical programming, queueing theory, system reliability, and scheduling. He is a member of ORSA, Mathematical Programming Society, and Phi Tau Phi Scholar Honor Society.

Dennis L Bricker

Dennis Bricker is Associate Professor of Industrial Engineering of The University of Iowa. He received B.S. and M.S. degrees in Mathematics from the University of Illinois ('65 and '66), and M.S. and Ph.D. degrees in Industrial Engineering and Management Sciences from Northwestern University ('72 and '75). He is a member of ORSA, TIMS, IIE, and the Mathematical Programming Society.
1 INTRODUCTION

We consider a standard form of linearly-constrained convex quadratic program:

\begin{align*}
\text{(QP)} & \quad \text{Min} \quad \frac{1}{2} x'Qx + c'x \\
& \quad \text{s.t.} \quad Ax - y = b \\
& \quad \quad \quad \quad x, y \geq 0
\end{align*}

Its Lagrangian dual problem is:

\begin{align*}
\text{(QPD)} & \quad \text{Max} \quad -\frac{1}{2} x'Qx + b'w \\
& \quad \text{s.t.} \quad -Qx + A'w + s = c \\
& \quad \quad \quad \quad s, w \geq 0
\end{align*}

where \( x, s, c \in \mathbb{R}^{n \times 1}, y, w, b \in \mathbb{R}^{m \times 1}, Q \in \mathbb{R}^{n \times n}, \) and \( A \in \mathbb{R}^{m \times n} \).

We make the following assumptions:

(A1) The matrix \( Q \) is positive semi-definite.

(A2) The feasible region is nonempty and bounded.

For \( x, y > 0 \) in (QP) and \( s, w > 0 \) in (QPD), we may apply the logarithmic barrier function technique, and obtain the nonlinear programming problems, \((\text{QP}_\mu)\) and \((\text{QPD}_\mu)\) as:

\begin{align*}
\text{(QP}_\mu) & \quad \text{Min} \quad \frac{1}{2} x'Qx + c'x - \mu \sum_{j=1}^{n} \log x_j - \mu \sum_{j=1}^{m} \log y_j \\
& \quad \text{s.t.} \quad Ax - y = b \\
& \quad \quad \quad \quad x, y > 0
\end{align*}

and

\begin{align*}
\text{(QPD}_\mu) & \quad \text{Max} \quad -\frac{1}{2} x'Qx + b'w + \mu \sum_{j=1}^{m} \log w_j + \mu \sum_{j=1}^{n} \log s_j \\
& \quad \text{s.t.} \quad -Qx + A'w + s = c \\
& \quad \quad \quad \quad w, s > 0
\end{align*}

where \( \mu > 0 \) is a barrier parameter.

It is well-known that the optimal solution of problem \((\text{QP}_\mu)\) would converge to the optimal solution of the original problem (QP) as \( \mu \to 0 \). Since \( Q \) is positive semi-definite, the convex programming theory further implies that the global solution, if one exists, is completely characterized by the KKT equations as [8]:

\begin{align*}
& \text{KKT for MIN: } \\
& \quad \text{First-order conditions: } F_x(x, y, z) = 0, \quad F_y(x, y, z) = 0, \quad F_z(x, y, z) = 0 \\
& \quad \text{Second-order conditions: } D_{xx}F(x, y, z) \succ 0, \quad D_{yy}F(x, y, z) \succ 0, \quad D_{xy}F(x, y, z) = 0
\end{align*}
\[
\begin{pmatrix}
A & -I \\
-Q & 0
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix}
+
\begin{pmatrix}
0 & 0 \\
I & A'
\end{pmatrix}
\begin{pmatrix}
s \\
w
\end{pmatrix}
- \begin{pmatrix}
b \\
c
\end{pmatrix}
=
\begin{pmatrix}
F_1(x,y,s,w) \\
F_2(x,y,s,w)
\end{pmatrix}
= \begin{pmatrix}
0 \\
ue
\end{pmatrix}
\]

(1.1)

where \(X, S, W,\) and \(Y\) are diagonal matrices with diagonal entries equal to the components of \(x, s, w,\) and \(y,\) respectively, and \(e\) is a vector of dimension \(m+n\) with each component 1. \(F_1\) refers to primal-dual feasibility and \(F_2\) refers to complementarity.

With the advent of the interior point algorithm (based on a projective method) by Karmarkar [7] for linear programming (LP), some attention has been devoted to the wider class of problems that can be solved by interior point algorithms in polynomial time, e.g. QP problems, linear complementarity problems, etc. For LP and QP problems, most (actually, almost all) of the researchers, e.g., Renegar [13], Monteiro and Adler [9][10], Monteiro, Adler, and Resende [11], etc., directly used Newton's method to solve the system of KKT equations (1.1) and to find the new search directions for primal-dual problem. Their algorithms generate a sequence of iterates following the central trajectory to obtain the optimal solution. Recently, Kojima, Megiddo, and Mizuno [7] proposed a so-called "infeasible interior-point algorithm" for LP problems. Their algorithm generates a sequence of iterates, not necessarily primal-dual feasible, within a neighborhood of the central trajectory for the system of equations (1.1). The main advantage of the infeasible interior-point algorithms is that there is no need to use Big-M or an extra phase to obtain the feasible initial interior point.

Recently a so-called "monomial method" has been used to solve system of nonlinear equations (Burns [1][2] and Burns and Locascio [3]). In this paper we propose a new infeasible path-following algorithm based on the monomial method. This algorithm, as we shall show, generates a sequence of iterates following a path on the complementarity surface, i.e., for each iteration \(k\) we have

\[
(x^k, y^k, s^k, w^k) \in J = \{(x, y, s, w) > 0 : F_2(x, y, s, w) = \mu e, \mu > 0\}.
\]
However, the primal-dual feasibility equation \( F_1(x^k, y^k, s^k, w^k) = 0 \) is not necessarily satisfied.

The main purpose of this paper is to study a new path-following algorithm via testing some randomly generated QP problems. Based on Calamai’s QP test problems [4], we propose a simple linear transformation to construct nonseparable QP problems with specific optimal solutions. In addition, we test a set of continuous quadratic knapsack test problems, and the variability in the computation times is reported for the algorithm. It should be mentioned that the convergence for this algorithm is still under investigation.

2. MONOMIAL ALGORITHM

2.1 Basic Concept of Monomial Method

The monomial method was proposed to solve a system of nonlinear equations. In contrast to Newton’s method, which is based on linearization of the nonlinear equations, the monomial method is based on a system of approximating equations that are monomial in form. Additionally, as shown by Burns and Locascio [3], the monomial method possesses several properties which are not commonly shared by other methods such as transformation invariance, matrix invariance, and system invariance etc. We briefly review the monomial method in this section and refer the readers to Burns [1][2] and Burns and Locascio [3] for more details.

Consider the following general system of \( N \) equations with \( N \) unknowns of the form:

\[
\sum_{i=1}^{T} \tilde{s}_{iq} \tilde{c}_{iq} \prod_{j=1}^{N} \bar{x}^{a_{ij}} = 0, \quad q = 1, 2, \ldots, N. \tag{2.1}
\]

where

\( \tilde{s}_{iq} \in \{-1, +1\} \) refer to signs of terms,

\( \tilde{c}_{iq} > 0 \) are referred to as coefficients,
$a_{ijq}$ real numbers without restriction in sign, are the exponents,

$\bar{x}_{ij} > 0$ are the variables,

$T_{q}$ is the total number of terms in equation $q$.

Next we define $u_{iq} = \bar{c}_{iq} \prod_{j=1}^{N} \bar{x}_{ij}^{a_{ijq}}$, so that (2.1) can be rewritten as

$$\sum_{i=1}^{T_{q}} s_{iq} u_{iq} = 0, \quad q = 1, 2, \ldots, N. \quad (2.2)$$

Let $T_{q}^{+} = \{ i \mid s_{iq} = 1 \}$ and $T_{q}^{-} = \{ i \mid s_{iq} = -1 \}$.

Hence, (2.2) can be expressed as:

$$\sum_{i \in T_{q}^{+}} u_{iq} - \sum_{i \in T_{q}^{-}} u_{iq} = 0, \quad q = 1, 2, \ldots, N. \quad (2.3)$$

We further define the "weight" of each term

$$\delta_{iq} = \frac{\bar{u}_{iq}}{P_{q}} \forall i \in T_{q}^{+} \text{ and } \delta_{iq} = \frac{\bar{u}_{iq}}{Q_{q}} \forall i \in T_{q}^{-}. \quad (2.4)$$

where

$$P_{q} = \sum_{i \in T_{q}^{+}} u_{iq}, \quad Q_{q} = \sum_{i \in T_{q}^{-}} u_{iq}, \quad \bar{u}_{iq} = u_{iq}{\mid_{x = x'}} , \quad \overline{P}_{q} = P_{q}{\mid_{x = x'}} , \text{ and } \overline{Q}_{q} = Q_{q}{\mid_{x = x'}}.$$

**Property 2.1:** $\sum_{i \in T_{q}^{+}} u_{iq} \geq \prod_{i \in T_{q}^{+}} (u_{iq} / \delta_{iq})^{\delta_{iq}}$ and $\sum_{i \in T_{q}^{-}} u_{iq} \geq \prod_{i \in T_{q}^{-}} (u_{iq} / \delta_{iq})^{\delta_{iq}}$, $q = 1, 2, \ldots, N$, with equalities when $\bar{x} = x'$.

**Proof.** The two inequalities follow from the weighted arithmetic-geometric mean inequality. Furthermore, for $\bar{x} = x'$, we have

$$\prod_{i \in T_{q}^{+}} (u_{iq} / \delta_{iq})^{\delta_{iq}}{\mid_{\bar{x} = x'}} = \prod_{i \in T_{q}^{+}} \left[ \frac{u_{iq}{\mid_{\bar{x} = x'}}}{\bar{P}_{q}} \right]^{\delta_{iq}} = \prod_{i \in T_{q}^{+}} \left( \bar{P}_{q} \right)^{\delta_{iq}} = \overline{P}_{q} = \sum_{i \in T_{q}^{+}} u_{iq}{\mid_{\bar{x} = x'}}. \quad (2.5)$$

The proof is similar for $i \in T_{q}^{-}$.

Using this property, we can approximate (2.3) as
\[
\prod_{i \in T_q} \left( u_{iq} / \delta_{iq} \right)^{\delta_{iq}} \prod_{i \in T_q} \left( u_{iq} / \delta_{iq} \right)^{\delta_{iq}} = 1
\]  

(2.5)

or equivalently (Burns and Locascio [3])

\[
H_q \prod_{j=1}^N \bar{x}_j^{D_{jq}} = 1
\]

(2.6)

where

\[
H_q = \prod_{i \in T_q} \left( \bar{c}_{iq} / \delta_{iq} \right)^{\delta_{iq}} \prod_{i \in T_q} \left( \bar{c}_{iq} / \delta_{iq} \right)^{\delta_{iq}} \quad \text{and} \quad D_{jq} = \sum_{i \in T_q} a_{iq} \delta_{iq} - \sum_{j \in T_q} a_{iq} \delta_{iq}
\]

(2.7)

Transforming the variables according to \( x_j = e^{z_j} \), we have

\[
\sum_{j=1}^N D_{jq} \bar{x}_j = -\log H_q, \; q = 1, 2, \ldots, N.
\]

(2.8)

Thus the linear equations (2.8) provide an alternate approximation to equation (2.1), and might be used to generate a different sequence of iterates (using the inverse transformation \( \bar{x}_j = -\log x_j \)) than that generated by Newton’s method. It should be noted that, since the exponential transformation is applied by the monomial method, the positivity of iterates is always maintained by the monomial method.

2.2 Derivation for Monomial Algorithm

Assume that \( (x^k, y^k, s^k, w^k) > 0 \) is an approximate solution for the QP problem.

After applying the monomial method to approximate equations (1.1), we have the following system of equations:

\[
\begin{bmatrix}
A_{1x}^k & A_{1y}^k & 0 & 0 & \|z_x^k\| & \log C_x^k & \xi_x^k \\
A_{2x}^k & 0 & A_{2y}^k & A_{2z}^k & \|z_y^k\| & \log C_y^k & \xi_y^k \\
I & 0 & I & 0 & \|z_s^k\| & -\log C_s^k & \xi_s^k \\
0 & I & 0 & I & \|z_w^k\| & \log C_w^k & \xi_w^k
\end{bmatrix} = 0
\]

(2.9)

where \( A_{1x}^k \in R^{m \times n}, \; A_{1y}^k \in R^{m \times m}, \; A_{2x}^k, A_{2y}^k, A_{2z}^k \in R^{n \times m}, \; A_{2w}^k \in R^{n \times m}, \; \xi_x^k, \xi_y^k, \xi_s^k, \xi_w^k \in R^{m \times 1} \) and \( \xi_x^k, \xi_y^k, \xi_s^k, \xi_w^k \in R^{n \times 1} \).

After solving (2.9), we may apply the inverse transformation

\[
(x^{k+1}, y^{k+1}, s^{k+1}, w^{k+1}) = (e^{z_x^k}, e^{z_y^k}, e^{z_s^k}, e^{z_w^k})
\]

(2.10)

and for each iteration we adjust \( \mu^k \) as
$$\mu^k = \sigma^k \frac{(x^k)'s^k + (y^k)'w^k}{n + m}, \quad 0 < \sigma^k < 1.$$  \hspace{1cm} (2.11)

2.3 Monomial Algorithm

Monomial Algorithm:

Step 0: (Initializing)

Set \( k = 0 \). Start with any initial solution \((x^0, y^0, s^0, w^0) > 0\), and choose three small values for \( \varepsilon_1, \varepsilon_2, \) and \( \varepsilon_3 \).

Step 1: (Checking optimality)

Compute \( m^k \) by (2.11), \( t_1^k = b + y^k - Ax^k \), and \( t_2^k = Qx^k + c - A^t w^k - s^k \).

If \( m^k < \varepsilon_1, \frac{f^k}{b} + 1 < \varepsilon_2, \) and \( \frac{\|x^k\|_2 + 1}{\|Qx^k + c\|_2 + 1} < \varepsilon_3 \), then stop, and the current solution is accepted as the optimal solution. Else go to the next step.

Step 2: (Evaluating weights)

Compute weights for positive and negative terms for each equation by (2.4).

Step 3: (Intermediate computation)

Compute \( A_{x_1}^k, A_{y_2}^k, A_{s_2}^k, A_{w_2}^k, \bar{x}^k, \bar{z}^k, \xi^k, \xi^k, \), and \( \bar{z}_w^k \) by (2.7) and (2.8).

Step 4: (Solving nonlinear equations)

Solve \( \bar{z}_x^k, \bar{z}_y^k, \bar{z}_s^k, \) and \( \bar{z}_w^k \) by (2.9).

Step 5: (Finding the new solution)

\((x^{k+1}, y^{k+1}, s^{k+1}, w^{k+1}) = (e^x, e^y, e^s, e^w)\)

Set \( k = k + 1 \), and go to Step 1.

2.4 Modified Monomial Algorithm

As the Step 5 shown in 2.3, the sequence generated by the Monomial algorithm is without any restrictions in the new directions \( \bar{z} = (\bar{z}_x, \bar{z}_y, \bar{z}_s, \bar{z}_w) \) (i.e., there is no upper bound for the norm of new direction vectors). For computational purpose, however, we
use a simple transformation and introduce a parameter \(0 < \eta^k < 1\) to keep the direction vectors \(\tilde{z} = (\tilde{z}_x, \tilde{z}_y, \tilde{z}_w)\) bounded. In practice, the modified Monomial algorithm can eliminate some numerical errors when implementing the algorithm. Consider

\[
e^{\tilde{z}_x^k} = x^k e^{dz_x^k}, \quad e^{\tilde{z}_y^k} = y^k e^{dz_y^k}, \quad e^{\tilde{z}_s^k} = s^k e^{dz_s^k}, \quad \text{and} \quad e^{\tilde{z}_w^k} = w^k e^{dz_w^k}.
\]

Taking the logarithm of the above terms, we have

\[
z_x^k = \ln x^k + dz_x^k, \quad n_y^k = \ln y^k + dz_y^k, \quad n_s^k = \ln s^k + dz_s^k, \quad \text{and} \quad n_w^k = \ln w^k + dz_w^k.
\]

Substituting these equations to (2.9) and adding the parameter \(\eta^k\), we get

\[
\begin{bmatrix}
A_{1x}^k & A_{1y}^k & 0 & 0 \\
A_{2x}^k & A_{2y}^k & dz_x^k & dz_y^k \\
I & 0 & I & 0 \\
0 & I & 0 & I
\end{bmatrix}
\begin{bmatrix}
dz_x^k \\
dz_y^k \\
dz_s^k \\
dz_w^k
\end{bmatrix}
= \begin{bmatrix}
\eta^k (\xi_x^k - (A_{1x}^k \ln x^k + A_{1y}^k \ln y^k)) \\
\eta^k (\xi_y^k - (A_{2x}^k \ln x^k + A_{2y}^k \ln s^k + A_{2w}^k \ln w^k)) \\
\xi_s^k - (\ln x^k s^k) \\
\xi_w^k - (\ln y^k w^k)
\end{bmatrix}
\begin{bmatrix}
\eta^k (\xi_x^k - (A_{1x}^k \ln x^k + A_{1y}^k \ln y^k)) \\
\eta^k (\xi_y^k - (A_{2x}^k \ln x^k + A_{2y}^k \ln s^k + A_{2w}^k \ln w^k)) \\
\eta^k (\xi_s^k - (\ln x^k s^k)) \\
\eta^k (\xi_w^k - (\ln y^k w^k))
\end{bmatrix}
\begin{bmatrix}
\eta^k (\xi_x^k - (A_{1x}^k \ln x^k + A_{1y}^k \ln y^k)) \\
\eta^k (\xi_y^k - (A_{2x}^k \ln x^k + A_{2y}^k \ln s^k + A_{2w}^k \ln w^k)) \\
\eta^k (\xi_s^k - (\ln x^k s^k)) \\
\eta^k (\xi_w^k - (\ln y^k w^k))
\end{bmatrix}
\begin{bmatrix}
\ln \sigma_x^k \\
\ln \sigma_y^k \\
\ln \sigma_s^k \\
\ln \sigma_w^k
\end{bmatrix}
\]

where \(0 < \eta^k < 1\) is used to control the norm of direction vectors.

Hence the modified Monomial algorithm may be stated as below.

**Modified Monomial Algorithm:**

The modified algorithm is the Monomial algorithm except that Step 0, Step 4, and Step 5 are replaced with Step 0', Step 4', and Step 5', respectively.

Step 0': (Initializing)

Set \(k = 0\). Start with any initial solution \((x^0, y^0, s^0, w^0) > 0\), choose three small values for \(\varepsilon_1, \varepsilon_2, \text{and} \varepsilon_3\). Define \(D_{\text{max}} > 0\).

Step 4': (Solving nonlinear equations)

Solve (2.12). If \(\max_{i,j} |dz_{x_i}^k, dz_{y_j}^k, dz_{s_i}^k, dz_{w_j}^k| > D_{\text{max}}\), then reduce \(\eta^k\) and resolve (2.12) until \(\max_{i,j} |dz_{x_i}^k, dz_{y_j}^k, dz_{s_i}^k, dz_{w_j}^k| < D_{\text{max}}\).

Step 5': (Finding the new solution)
\[(x^{k+1}, y^{k+1}, s^{k+1}, w^{k+1}) = (x^k e^{d^k_x}, y^k e^{d^k_y}, s^k e^{d^k_s}, w^k e^{d^k_w})\].

Set \(k = k + 1\), and go to Step 1.

3. SOME PROPERTIES OF MONOMIAL ALGORITHM

In this section, we show that the sequence generated by the Monomial algorithm follows a path on the complementaritity surface, that is \((x^k, y^k, s^k, w^k) \in J\) for all \(k\). In addition, we prove that the complementarity is reduced by a constant for each iteration \(k\) when \(\sigma^k = \sigma\) for all \(k\).

Property 3.1: The elements of matrices \(\xi^k_s\) and \(\xi^k_w\) in (2.9) are all \(-\log(1/m^k)\).

Proof. The property follows immediately by (2.7).

This property implies that the sequence generated by the Monomial algorithm is on a path of the complementarity surface. That is, \((x^k, y^k, s^k, w^k) \in J\) for all \(k\).

Property 3.2: Let \(\sigma^k = \sigma\) for all \(k\). Then we have \(\mu^{k+1}/\mu^k = \sigma\).

Proof. Since \((x^k, y^k, s^k, w^k) \in J\) for all \(k\) by Property 3.1, we get
\[
\frac{\mu^{k+1}}{\mu^k} = \frac{\sigma x^{k+1}_s}{\mu^k} = \frac{\sigma e^{\xi^k_x + \xi^k_s}}{\mu^k} = \frac{\sigma e^{-\ln(1/m^k)}}{\mu^k} = \sigma.
\]

4. NUMERICAL RESULTS

In this section the results of two numerical experiments are reported. All the results were obtained using a HP715/75 workstation and the algorithm (modified Monomial algorithm) was coded in APL.

In the first experiment, according to Calamai et al. [4], we construct the separable QP test problems. A simple linear transformation is proposed to construct the nonseparable QP test problems with specific optimal solutions. Therefore computational
results of both separable and nonseparable problems are reported. The aim of the second experiment is to evaluate the variability of this algorithm. The test procedure and test problems follow Dussault et al.\[5\] and Pardalos et al. \[12\], and the results are compared with those of both an iterative algorithm and a potential reduction algorithm.

4.1. The Construction of Test Problems

4.1.1 Separable QP Problems

Following Calamai et al. \[4\] we briefly present the construction of the test problems for the separable linearly-constrained QP problems. The optimal solutions for the constructed test problems can be specified, and may be either extreme points, interior points, or boundary points.

Consider the subproblem \(l\) of the separable convex quadratic problem in the form:

\[
\begin{align*}
\text{(SQP}_l\text{)} & \quad \text{Min} \quad f_l(x_{1l}, x_{2l}) = (1/2)\left\{\rho_{1l}(x_{1l} - t_{1l}^0)^2 + \rho_{2l}(x_{2l} - t_{2l}^0)^2\right\} \\
& \quad \text{s.t.} \quad 3x_{1l} + 2x_{2l} \geq \alpha_l \\
& \quad \quad \quad 2x_{1l} + 3x_{2l} \geq \alpha_l \\
& \quad \quad \quad -x_{1l} - x_{2l} \geq -3 \\
& \quad \quad \quad x_{1l}, x_{2l} \geq 0
\end{align*}
\]

where \(\alpha_l \in [5,15/2)\), \(\rho_{1l}, \rho_{2l}, \theta_{1l}, \theta_{2l} \in \{0,1\}\), \(t_{1l}, t_{2l} \in \mathbb{R}^{n_l}\).

As the discussions in Calamai et al. \[4\], we have the following cases for the optimal solutions:

Case 1: If \(t_{1l}^0 = t_{2l}^0 = 1\) and \(\rho_{1l} = \rho_{2l} = 1\), the optimal solution is extreme point \((\alpha_l/5, \alpha_l/5)\) and if \(t_{1l}^0 = 3\) and \(\rho_{1l} = 1\), \(\rho_{2l} = 0\), the optimal solution is the extreme point \((9 - \alpha_l, \alpha_l - 6)\).

Case 2: \(t_{1l}^0 = t_{2l}^0 = 3\) and \(\rho_{1l} = \rho_{2l} = 1\), the optimal solution is a boundary point \((3/2,3/2)\),

Case 3: \(t_{1l}^0 = \alpha_l/5, \alpha_l/5 < t_{2l}^0 < 3 - (\alpha_l/5)\) and \(\rho_{1l} = \rho_{2l} = 1\), the optimal solution is the interior point \((t_{1l}^0, t_{2l}^0)\).

Thus, we may define the separable convex quadratic problems as:
\( (\text{SQP}) \)

Min \( F(x) = \sum_{i=1}^{M} f_i(x_{1i}, x_{2i}) \)

s.t. \( 3x_{1i} + 2x_{2i} \geq \alpha_i \)

\( 2x_{1i} + 3x_{2i} \geq \alpha_i \)

\( -x_{1i} - x_{2i} \geq -3 \)

\( x_{1i}, x_{2i} \geq 0, \; l \in \{1,2,\ldots,M\} \) \tag{4.2}

where \( M \) is the number of subproblems in (SQP).

It should be noted that the coefficients of the linear constraints in (4.2) may be different; the only requirement is that they be linearly independent and that the feasible region be nonempty.

**Example (separable QP problems):**

Min \( \frac{1}{2} \{(x_1 - 30)^2 + (x_2 - 5)^2 + (x_3 - 35)^2 + (x_4 - 25)^2 \} - 1387.5 \)

\[
\begin{bmatrix}
3 & 0 & 1 & 0 & 0 \\
0 & 5 & 2 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 \\
0 & 0 & 7 & 3 & 0 \\
0 & 0 & 2 & 6 & 0 \\
0 & 0 & -1 & -1 & 0 \\
\end{bmatrix} \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
\end{bmatrix} \geq \begin{bmatrix}
50 \\
50 \\
-30 \\
63 \\
63 \\
-30 \\
\end{bmatrix}
\]

\( x \geq 0 \)

with optimal solution \( x^* = (27.5, 2.5, 20, 10)' \).

**4.1.2 Nonseparable QP Problems**

Using the separable QP problems generated in Section 4.1.1, we illustrate an easy linear transformation procedure to construct the nonseparable QP problems. Consider the following three problems.

\( (P) \)

Min \( f(x) = \frac{1}{2} x'Qx + c'x \)

s.t. \( Ax \geq b \)

\( x \geq 0 \)
\((P')\)

\[
\begin{align*}
\text{Min } f(x') &= \frac{1}{2} x'(T'Q) x' + (T'c)' x' \\
\text{s.t. } ATx' &\geq b \\
T'x' &\geq 0
\end{align*}
\]

and

\((P)\)

\[
\begin{align*}
\text{Min } f(x) &= \frac{1}{2} (x - \omega)' (T'Q) (x - \omega) + (T'c)' (x - \omega) \\
\text{s.t. } AT(x - \omega) &\geq b \\
T(x - \omega) &\geq 0 \\
\bar{x} &\geq 0
\end{align*}
\]

If \(T \in \mathbb{R}^{n \times n}\) is a nonsingular and lower triangular matrix, and we let \(x' = T^{-1}x\) and \(\bar{x} = T^{-1}x + \omega\), then we have the following lemma and theorem.

**Lemma 4.1:** \(T'Q\) is positive semi-definite.

**Proof.** Since \(Q\) is positive semi-definite, we have \(Q = L'DL\), where \(L\) is a lower triangular matrix and \(D\) is a diagonal matrix with nonnegative elements.

For \(u \neq 0\), we have

\[
u' (T'Q) u = u' (L'DL) Tu
= u' (LT)' DLT u
= u' \bar{L}' D \bar{L} u
= (\bar{L} u)' D \bar{L} u
= (L' \bar{L} u)' DL'
= \sum_y d_y (l_y')^2 \geq 0
\]

This completes the proof.

**Theorem 4.2:**

If \(x^*\) is the optimal solution for problem \((P)\) and \(\omega = \max(0, -\min_i T^{-1} x_i^*)\), then

1. \(T^{-1} x^*\) is the optimal solution for problem \((P')\), and
2. \(T^{-1} x^* + \omega\) is the optimal solution for problem \((P)\).
Proof.

(1) Since $x' = T^{-1} x^*$ satisfies all the constraints of problem $(P')$, it is a feasible solution for problem $(P')$. Suppose that there exists a solution $x_{p'} = T^{-1} x_p$ for problem $(P')$, where $x_p$ is feasible for problem $(P)$, such that $f(x_{p'}) < f(x')$. Then we have $f(x_p) < f(x^*)$ (since the objectives for $(P)$ and $(P')$ are the same). This is a contradiction, since $x^*$ is the optimal solution for problem $(P)$.

(2) Similar proof.

By Lemma 4.1 one observes that problem $(P')$ is a convex quadratic problem whose variables are without restriction in sign and problem $(\overline{P})$ is a nonseparable convex quadratic problem. However the optimal solutions for these two problems can be obtained by separable problem $(P)$ through some linear transformation. Therefore, we may construct nonseparable QP test problems with known solutions.

Example (nonseparable):

Consider again the separable QP example with optimal solution $x^* = (27.5, 2.5, 20, 10)^t$ in above section. Let the linear transformation mapping matrix $T$ be the lower triangular matrix with ones for all the nonzero elements, i.e.,

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Since $T^{-1} x^* = (27.5, -25, 17.5, -10)^t$ and $\omega = \max_i (0, -\min_i T^{-1} x^*) = 25$, using the transformation of $\overline{x} = T^{-1} x + 25$, we get the nonseparable problem $(\overline{P})$ as
\[
\begin{align*}
\text{Min} & \quad \frac{1}{2} \begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} x + \begin{pmatrix} -345 \\ -290 \\ -235 \\ -125 \end{pmatrix} x \\
\text{s.t.} & \quad \begin{pmatrix}
9 & 6 & 0 & 0 \\
7 & 2 & 0 & 0 \\
-2 & -1 & 0 & 0 \\
10 & 10 & 10 & 3 \\
8 & 8 & 8 & 6 \\
-2 & -2 & -2 & -1 \\
\end{pmatrix} x \geq \begin{pmatrix}
425 \\
275 \\
-105 \\
888 \\
813 \\
-205 \\
\end{pmatrix}, \\
\begin{pmatrix}
1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \end{pmatrix} x \geq \begin{pmatrix}
25 \\
50 \\
75 \\
100 \end{pmatrix}, \\
\bar{x} \geq 0
\end{align*}
\]

which is a nonseparable QP problem and the optimal solution is given by
\[\bar{x}^* = T^{-1} x^* + 25 = (52.5, 0, 42.5, 15)'.\] Note that, for this example, the density of the Hessian matrix is 100%, i.e., all the elements in \(Q\) are nonzero.

It should be noted that the elements in \(T\) may be arbitrarily chosen. The only requirements are that \(T\) be nonsingular and lower triangular.

### 4.2 Numerical Results for Separable and Nonseparable QP Test Problems

Using the separable and nonseparable test problems in Section 4.1, we test QP problems with various problem sizes, namely, \(M = 2, M = 4, M = 8\), and \(M = 12\). The number of variables and constraints are \(2M\) and \(3M\) for separable problems, and \(2M\) and \(5M\) for nonseparable problems. The other characteristics are illustrated in Table 4.1.

For these test problems we impose the following conditions.

(C4.2.1) For each size of both separable and nonseparable problem, 10 random test problems are tested, and for each test problem the initial solution is randomly generated for this algorithm.

(C4.2.2) \(\epsilon_1 = \epsilon_2 = \epsilon_3 = 0.00001\).

(C4.2.3) For the modified Monomial algorithm, we let \(\eta^0 = 0.95\), and adjust the \(\eta\) by
Table 4.1. Test problems’ densities (% of nonzero elements).

<table>
<thead>
<tr>
<th>Number of Subproblems</th>
<th>Separable Problems</th>
<th>Nonseparable Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Matrix A</td>
<td>Matrix Q</td>
</tr>
<tr>
<td>M=2</td>
<td>50.00%</td>
<td>25.00%</td>
</tr>
<tr>
<td>M=4</td>
<td>25.00%</td>
<td>12.50%</td>
</tr>
<tr>
<td>M=8</td>
<td>12.50%</td>
<td>6.25%</td>
</tr>
<tr>
<td>M=12</td>
<td>8.33%</td>
<td>4.17%</td>
</tr>
</tbody>
</table>

\[
\eta^{k+1} = \begin{cases} 
\max(0.2, 0.5 \eta^k) & \text{if } \max_{i,j} \{d_{z_{i,k}}^+, d_{z_{i,k}}^-, d_{z_{j,k}}^+, d_{z_{j,k}}^-\} > 10 \\
\min(0.95, 1.2 \eta^k) & \text{otherwise}
\end{cases}
\]

(C4.2.4) Different values for \( \sigma \) are tested, namely, 0.3, 0.5, 0.7, 0.9, and 0.95.

The number of iterations and CPU time (in seconds) for both separable and nonseparable QP problems with different problem sizes are reported in Table 4.2. From Table 4.2, the results may be summarized as followings:

1. On the average, the algorithm needs more iterations for nonseparable problems than for separable problems.
2. The average of the number of iterations and CPU time are reduced with the decrease of \( \sigma \). (Note: By Property 3.2 we observe that the complementarity is reduced by a constant \( \sigma \) at each iteration. Thus the complementarity will improve faster if \( \sigma \) is smaller. But too small a value of \( \sigma \), while it rapidly reduces complementarity, will result in some computational errors in practice. The ideal algorithm is one which would satisfy the primal-dual feasibility before complementarity.)
3. The number of iterations for the Monomial algorithm is very stable, but the total CPU time increases drastically with the increase of problem size.

4.3 Continuous Quadratic Knapsack Problems
To measure the predictability of this algorithm, we follow Dussault et al. [5] and Pardalos et al. [12] to solve different scales of continuous quadratic knapsack problems, in Table 4.2 The number of iterations and CPU for the separable and nonseparable test problems.

<table>
<thead>
<tr>
<th>$\sigma^0$</th>
<th>number of subproblems</th>
<th>$M = 2$</th>
<th>$M = 4$</th>
<th>$M = 8$</th>
<th>$M = 12$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.95 separable</td>
<td>22.7 (0.809)</td>
<td>28.0 (1.721)</td>
<td>27.6 (5.862)</td>
<td>20.0 (9.495)</td>
<td></td>
</tr>
<tr>
<td>nonseparable</td>
<td>21.0 (1.045)</td>
<td>21.5 (2.48)</td>
<td>27.1 (12.407)</td>
<td>24.1 (26.473)</td>
<td></td>
</tr>
<tr>
<td>0.90 separable</td>
<td>18.0 (0.609)</td>
<td>19.4 (1.178)</td>
<td>19.5 (4.12)</td>
<td>19.9 (9.441)</td>
<td></td>
</tr>
<tr>
<td>nonseparable</td>
<td>18.1 (0.762)</td>
<td>19.8 (2.303)</td>
<td>22.4 (10.266)</td>
<td>23.2 (25.575)</td>
<td></td>
</tr>
<tr>
<td>0.70 separable</td>
<td>15.4 (0.541)</td>
<td>18.6 (1.119)</td>
<td>19.0 (4.021)</td>
<td>19.1 (9.084)</td>
<td></td>
</tr>
<tr>
<td>nonseparable</td>
<td>18.0 (0.941)</td>
<td>19.3 (2.337)</td>
<td>21.6 (10.101)</td>
<td>23.7 (26.132)</td>
<td></td>
</tr>
<tr>
<td>0.50 separable</td>
<td>12.9 (0.323)</td>
<td>14.5 (0.908)</td>
<td>15.5 (3.264)</td>
<td>16.0 (7.574)</td>
<td></td>
</tr>
<tr>
<td>nonseparable</td>
<td>16.2 (0.793)</td>
<td>16.8 (1.958)</td>
<td>19.4 (8.895)</td>
<td>19.2 (21.169)</td>
<td></td>
</tr>
<tr>
<td>0.30 separable</td>
<td>12.0 (0.283)</td>
<td>12.7 (0.764)</td>
<td>13.5 (2.877)</td>
<td>14.0 (6.619)</td>
<td></td>
</tr>
<tr>
<td>nonseparable</td>
<td>15.1 (0.664)</td>
<td>13.3 (1.3)</td>
<td>15.8 (7.243)</td>
<td>17.6 (19.455)</td>
<td></td>
</tr>
</tbody>
</table>

which the Hessian matrix can be generated such that the matrix is diagonally dominant with various degrees.

We consider the continuous quadratic knapsack problems of the form:

$$\text{Min } \frac{1}{2} x^t Q x$$

$$s.t. \sum_{i=1}^{n} x_i = 1, \ x \geq 0$$
convex hull of points, and the maximum clique problem, etc., (Pardalos et al. [12]). In this paper the test problems were generated according to the method used by Dussault et al.

\[ Q = p\Delta + GG^T, \]  
where \( \Delta \) is a diagonal matrix with

\[ \Delta = \text{diag}(d_1, d_2, \ldots, d_n) \]  
where \( d_i = (n-1)m, \ 1 \leq i \leq n \),

\( G \) is an \( n \times m \) matrix with elements randomly generated from \([0,1]\), and \( p \) is an input parameter. We observe that \( GG^T \) is positive definite with rank \( n \) and all elements are less than \( n \). It is obvious that \( Q \) is positive definite and diagonally dominant when \( p \) is 1 and the diagonal dominance becomes small as \( p \) gets close to 0. The results are reported in Section 4.4.

4.4 Numerical Results for Continuous Quadratic Knapsack Problems

For the above test problems, we use similar conditions as those test problems in Section 4.2, except that:

(C4.4.1) Different numbers of problem sizes, \( n = 10, 30, 50, \) and 100, i.e., the number of primal variables, are tested for \( m = 5, 10, 15, 20, 25, \) and \( p = 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0. \)

(C4.4.2) Instead of testing only one problem for each combination of parameters \( n, m, \) and \( p \) as in Pardalos et al. [12], we test 10 random QP problems and compute the mean CPU time.

(C4.4.3) \( \sigma = 0.3. \)

(C4.4.4) To conform to our QP formulation and to avoid increasing the problem size, we replace the constraint \( \sum_{i=1}^{n} x_i = 1 \) with \( \sum_{i=1}^{n} x_i \geq 1. \) Note that for this inequality constraint, the optimal solutions will be the same as those for the equality
constraint, since for this continuous quadratic knapsack test problems we have $Qx \geq 0$, i.e., the objective function is convex and increasing.

To measure the predictability for this algorithm, for each combination of $m$ and $n$, the minimum and maximum of the CPU times of the 190 problems (since 19 different values for $p$ ) are determined, and the ratio
\[
\frac{\text{Max CPU time}}{\text{Min CPU time}}
\]
is computed as a measure of the predictability of computational effort of the modified Monomial algorithm. It is expected that an algorithm is more predictable and stable if the ratio is closer to 1. The main advantage of this measure is that this value for the measure totally depends on the ratio of CPU time, hence different computer languages, computers, and implementation skills do not so significantly affect the value of this measure.

The Max CPU/Min CPU, for each combination of $n$ and $m$, is shown in Table 4.3. Note that each ratio value is based on $19 \times 10 = 190$ random problems, since 10 random problems are tested for 19 different values of $p$ when $n$ and $m$ are fixed. Thus, this table reports the results of testing 3,800 random problems.

From Table 4.3, we can conclude that the ratios of (Max CPU/ Min CPU) for the new path-following algorithm are in the range (1.1930,1.4201), and the range and values of the intervals are all smaller than those presented by Pardalos et al. [12] for the potential reduction algorithm, namely (1.90,2.60), and by Dussault [5], namely (9.40,15.00), for an iterative algorithm. This implies that this path-following algorithm seems more predictable than both iterative algorithm and potential reduction algorithm.

Table 4.3  Max CPU/Min CPU for Monomial algorithm.

<table>
<thead>
<tr>
<th>Number of variables</th>
<th>Monomial Algorithm</th>
<th>Ratio of CPU (min,max)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$m=5$</td>
<td>$m=10$</td>
</tr>
<tr>
<td>$n =10$</td>
<td>1.2069</td>
<td>1.3571</td>
</tr>
<tr>
<td>$n =30$</td>
<td>1.2900</td>
<td>1.2061</td>
</tr>
<tr>
<td>$n =50$</td>
<td>1.3020</td>
<td>1.2635</td>
</tr>
<tr>
<td>$n =100$</td>
<td>1.3453</td>
<td>1.3097</td>
</tr>
</tbody>
</table>
5 CONCLUSIONS

In this paper, based on the monomial method, we have proposed a new infeasible interior point algorithm which generates a sequence of iterates following a path on the complementarity surface rather than the central path for the linearly-constrained QP problems. Both separable and nonseparable QP problems are constructed to test this new algorithm. In addition, the variability in the computation times is reported for the continuous knapsack problems. Limited numerical results indicate that this algorithm performs better than both iterative algorithm and potential reduction algorithm. However, the convergence of this algorithm and the choice for the values of $\sigma$ and $\eta$ still need further investigation.
REFERENCE


