Fixed-Point Iteration Method for Solving the Convex Quadratic Programming with Mixed Constraints

Ruopeng Wang, Hong Shi, Kai Ruan, Xiangyu Gao
Department of Mathematics & Physics, Beijing Institute of Petrochemical Technology, Beijing, China
Email: wangruopeng@bipt.edu.cn
Received October 24, 2013; revised November 24, 2013; accepted December 3, 2013

ABSTRACT
The present paper is devoted to a novel smoothing function method for convex quadratic programming problem with mixed constraints, which has important application in mechanics and engineering science. The problem is reformulated as a system of non-smooth equations, and then a smoothing function for the system of non-smooth equations is proposed. The condition of convergences of this iteration algorithm is given. Theory analysis and primary numerical results illustrate that this method is feasible and effective.

KEYWORDS
Fixed-Point Iteration; Convex Quadratic Programming Problem; Convergence; Smoothing Function

1. Introduction
Consider the following convex quadratic programming problem
\[
\min -\frac{1}{2}x^TQx + c^Tx
\]
\[
\text{s.t. } \begin{cases} A_{11}x_1 + A_{12}x_2 \geq b_1 \\ A_{21}x_1 + A_{22}x_2 = b_2 \end{cases}
\]
where \( Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \) is an \( n \times n \) symmetrical positive semi-definite partitioned matrix, \( A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \) is an \( m \times n \) partitioned matrix, \( b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \) and \( c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \) are vectors in \( R^n \) and \( R^m \), respectively. And \( Q_1, Q_2, A_1, A_2 \) are the partitioned matrices of \( Q \) and \( A \) respectively.

As is well known, this problem models a large spectrum of applications in computer science. Operation research and engineering as the nonlinear programming is solved eventually based on the convex quadratic programming problem. Therefore, the study on its effective solvers becomes a prolonged research subject.

Common effective algorithm is proposed related to certain schemes of convex quadratic programming problem, see [1-5]. Moreover, these methods are not polynomial algorithm in [6]. And the theory and algorithm of quadratic programming are presented completely. According to the optimization condition (KKT) and the duality theory of quadratic programming, the fixed-point iteration method is obtained for equality constrained (see [7]) and inequality constrained (see [8]) convex quadratic programming problem respectively. In fact, equality and inequality can be converted with inputting artificial variables and slack variables. The main default is to increase the scale and numerical difficulty while introduces these variables. Motivated by smoothing methods of
[9-11] and fixed-point method of [7,8], the paper mainly concerns about the mixed constraint quadratic programming and the fixed-point iteration method is given. And also the rank of the coefficient matrix is not full. The method considering is fulfilled efficiently. Thus the proposing question in [8] is solved.

2. Scheme of Algorithm

According to the dual theory, the dual problem of primal problem (1) is

\[
\begin{align*}
\max & -\frac{1}{2} x^T Q x + b^T y \\
\text{s.t.} & \begin{cases}
Q_{11} x_i + Q_{12} x_H - A_{11}^T y_i - A_{21}^T y_H + c_i \geq 0 \\
Q_{21} x_i + Q_{22} x_H - A_{12}^T y_i - A_{22}^T y_H + c_H = 0
\end{cases}
\end{align*}
\]

(2)

where \( y = \begin{bmatrix} y_i \\ y_H \end{bmatrix} \in R^n \).

Let \( u = \begin{bmatrix} u_i \\ u_H \end{bmatrix}, u_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, u_H = \begin{bmatrix} x_H \\ y_H \end{bmatrix} \), we know if \( x^* = \begin{bmatrix} x_i^* \\ x_H^* \end{bmatrix} \in R^n \) is the optimal solution of (1), there exists a vector \( y^* = \begin{bmatrix} y_i^* \\ y_H^* \end{bmatrix} \in R^n \), an \( n \times n \) symmetrical positive semi-definite partitioned matrix, and \( (x^*, y^*) \) is the optimal solution of (2).

Let us define the following notation

\[
\begin{align*}
F &= \{ u = (x_i, y_i, x_H, y_H) \in R^{*\times n} | u_i \geq 0 \} \\
F^* &= \{ u^* = (x_i^*, y_i^*, x_H^*, y_H^*) \} (x_i^*, x_H^*) \text{ is the optimal solution of (1) and } (x_i^*, y_i^*, x_H^*, y_H^*) \text{ is the optimal solution of (2)} \}
\end{align*}
\]

It is easy to verify that \( F \) is a closed set.

Suppose that the feasible region \( F^* \) is not empty. A necessary and sufficient condition for an optimal solution is that:

1) Primal feasibility

\[
\begin{align*}
A_{11} x_i + A_{12} x_H - b_i & \geq 0 \\
A_{21} x_i + A_{22} x_H - b_H & = 0
\end{align*}
\]

2) Dual feasibility

\[
\begin{align*}
Q_{11} x_i + Q_{12} x_H - A_{11}^T y_i - A_{21}^T y_H + c_i & \geq 0 \\
Q_{21} x_i + Q_{22} x_H - A_{12}^T y_i - A_{22}^T y_H + c_H & = 0
\end{align*}
\]

3) Complementary condition

\[
\begin{align*}
x_i^T [Q_{11} x_i + Q_{12} x_H - A_{11}^T y_i - A_{21}^T y_H + c_i] & = 0 \\
y_i^T [A_{11} x_i + A_{12} x_H - b_i] & = 0
\end{align*}
\]

The scheme are rewritten below

\[
u_i \geq 0, (Mu + q)_i \geq 0, u^T (Mu + q) = 0 \text{ (Mu + q)}_H = 0
\]

(3)

Where

\[
M = \begin{bmatrix}
Q_{11} & -A_{11}^T & Q_{12} & -A_{21}^T \\
A_{11} & 0 & A_{12} & 0 \\
Q_{21} & -A_{12}^T & Q_{22} & -A_{22}^T \\
A_{21} & 0 & A_{22} & 0
\end{bmatrix}
\]
$q = \begin{bmatrix} q_l \\ q_u \end{bmatrix}, q_i = \begin{bmatrix} c_l \\ -b_j \end{bmatrix}, q_{ii} = \begin{bmatrix} c_{ii} \\ -b_{ii} \end{bmatrix}$

More formally, the following link fixed-point iteration problem is performed

$$u = \max\left(0, u - r (Mu + q)\right)$$

(4)

The component form is

$$u_j = \max\left(0, u - r (M_j u_j + q_j)\right)$$

(5)

Where $r > 0$.

The question (5) is a non-smooth optimization problem, and the Newton-type methods cannot be applied directly. In the paper, we converted the problem into a smooth optimization.

At the beginning, we introduce a function mapping $R$ into $R^+$, and have the following properties.

**Property 1** Let $\theta: R \rightarrow R^*$ be a real function, and such that:

1) $\theta(t)$ is strictly convex and differentiable;
2) $\lim_{t \to +\infty} \theta(t) = 0$;
3) $\lim_{t \to -\infty} \theta(t) = 1$.

We assumed that $\theta(t)$ is a convex and differentiable function, which indicates one can apply the classical Newton-type algorithm directly. And the assumption $\theta'(0) > 0$ means that 0 is not a sub-gradient of $\theta$ at 0.

The hypothesis (3) and (4) indicate the approximate function $\theta(t)$ has the same numerical properties as the max-function.

We consider the following function, and $\theta_p(t)$ describe as above

$$\theta_p(t) = p \theta\left(\frac{t}{p}\right)$$

(6)

For any $p \in (0,1]$, $\theta_p(t)$ satisfies the following properties:

**Theorem 1** Let be a function given as (6), then for any arbitrary $p \in (0,1]$, $\theta_p(t)$ is strictly convex and differentiable, and we have $\frac{d\theta_p(t)}{dt} \bigg|_{t=0} > 0$.

**Proof** According to property 1 and (6), the convexity and differentiability are straightforward.

Nothing that

$$\frac{d\theta_p(t)}{dt} \bigg|_{t=0} = \theta'\left(\frac{t}{p}\right) \geq \theta'(t)$$

The property (2) implies $\frac{d\theta_p(t)}{dt} \bigg|_{t=0} > 0$.

**Theorem 2** Let $\theta_p(t): R \rightarrow R^+$ be a function given as (6), then $\lim_{t \to +\infty} \theta_p(t) = \max\{0, t\}$. That is, the function $\theta_p(t)$ uniformly converges to the max-function, when the adjustable parameter $p$ approaches 0.

**Proof** For $t$ fixed, we can compute

$$\lim_{p \to 0} \theta_p(t) = \lim_{p \to 0} p \theta\left(\frac{t}{p}\right)$$

If $t < 0$, then $\lim_{p \to 0} p \theta\left(\frac{t}{p}\right) = 0$; otherwise, change variable $z = \frac{t}{p}$,

$$\lim_{p \to 0} \theta_p(t) = \lim_{z \to +\infty} t \cdot \frac{\theta(z)}{x} = t$$

OPEN ACCESS
Hence, \( \lim_{p \to +0} \theta_p(t) = \max \{0, t\} \) for all \( t \).

From the discussion above, we introduce the following function in the paper
\[
\theta(t) = \ln(1 + e^t)
\]
and
\[
\theta_p(t) = p\theta\left(\frac{t}{p}\right) = p\ln\left(1 + e^{t/p}\right)
\]

It is easily known that satisfies the properties (1) to (4), together with theorem 1 and 2. Thus, the max-function in (3) replaced by \( \theta_p(t) = p\ln\left(1 + e^{t/p}\right) \), and the fixed-point iteration which basing on smoothing function is obtained.

**Algorithm**

Step 1 Set an initial point \( u_0 \in \mathbb{R}^{m+n} \), choose \( \varepsilon > 0 \), \( 0 < p_0 \leq 1 \), \( l \in (0,1) \), let \( k = 0 \);

Step 2 Apply fixed-point algorithms to solve

\[
u_{j+1}^k = \theta_{p_0}\left(u_j^k - r_k\left(Mu_j^k + q_j\right)\right)
\]

\( j = 1, 2, \ldots, n + m \) \hspace{2cm} (7)

Step 3 Terminal condition:

\[\|u^k\|_T^T(Mu^k + q)| < \varepsilon \], stop; otherwise, go to Step 4;

Step 4 Let \( p_{k+1} = lp_k \), \( k = k+1 \), go to step 2.

The Jacobian matrix in (7) is shown

\[B = V(I - rM^T)\] \hspace{2cm} (8)

Where

\[
V = \begin{pmatrix}
    v_1 \\
    \vdots \\
    v_{n+m}
\end{pmatrix}, \quad v_j = \frac{e^{\frac{1}{p}r_j - q_j\left(Mu_j^k + q_j\right)}}{1 + e^{\frac{1}{p}r_j - q_j\left(Mu_j^k + q_j\right)}}
\]

3. Convergence Analyses

Now we discuss the convergence of the algorithm. To begin with, two simple lemmas are introduced that is the basis of our theoretical analysis. Since their proof can be found in reference [12], we here omit the proof due to space.

**Lemma 1** [12] A necessary and sufficient condition for the fixed-point iteration method to be convergent is that the spectral radius of Jacobian matrix \( B \) satisfies \( \rho(B) < 1 \), where the spectral radius \( \rho(B) \) be defined as

\[
\rho(B) = \max_{i \geq 1} \lambda_i, \quad \lambda_i \text{ denotes the eigenvalue of matrix } B^T B.
\]

**Lemma 2** [12] Suppose \( M \) is a matrix, whose eigenvalues can be ordered from the smallest to largest, i.e. \( |\lambda_1| \leq |\lambda_2| \leq \cdots \leq |\lambda_{n+m}| \), and \( V \) is diagonal matrix with nonnegative diagonal elements, then the maximum norm of the matrix \( VM \) ’s eigenvalue such that \( \|V\| \leq 1 \).

**Theorem 3** suppose \( \lambda_1, \lambda_2, \ldots, \lambda_{n+m} \) be the eigenvalues of the Jacobian matrix \( M \), if one of the following conditions is true, then we can select the proper parameter \( r_k \) makes the algorithm convergent:

(1) The real part of \( \lambda_j \) is positive, that is \( \Re \lambda_j > 0 \) \( (j = 1, 2, \ldots, n + m) \);

(2) All of the eigenvalues \( \lambda_j (j = 1, 2, \ldots, n + m) \) of matrix \( M \) are real.

**Proof** Suppose \( \mu_j (j = 1, 2, \ldots, n + m) \) are eigenvalues of the Jacobian matrix in (8), and matrix \( M \) ’s eigenvalues denote as \( \lambda_j (j = 1, 2, \ldots, n + m) \).

Let \( \lambda = x + yi \) (where \( i = \sqrt{-1} \)).
Note that

\[ |1 - r\lambda| \leq 1 \iff r\left(x^2 + y^2\right) \leq 2x \]

If \( \lambda = 0 \), for all positive number of \( r \); while \( \lambda \neq 0 \), we know \( r \leq \frac{2x}{x^2 + y^2} \).

1) If \( \lambda_j \) \((j = 1, 2, \cdots, n+m)\) is positive, we let \( r = \frac{2\min_j \{\text{Re}\lambda_j\}}{\max_j \{\text{Re}\lambda_j\}} \), and have \( |1 - r\lambda_j| \leq 1 \)

\((j = 1, 2, \cdots, n+m)\);

2) If all of the eigenvalues \( \lambda_j \) \((j = 1, 2, \cdots, n+m)\) of matrix \( M \) are real, we let \( r = \frac{2}{\max_j \{\text{Re}\lambda_j\}} \), and have \( |1 - r\lambda_j| \leq 1 \)

For all discussed above, associate with Lemma 1, we know that the matrix \( M \) has the eigenvalues of \( \mu_j \) \((j = 1, 2, \cdots, n+m)\), such that

\[ |\mu_j| \leq \max_j \{v_j\} |1 - r\lambda_j| \leq \max_j \{v_j\} < 1 \]

According to the theorem above, the algorithm is always convergent if the parameter \( r \) is chosen properly. Thus the recent relevant results in reference [8] are extended.

4. Numerical Experiment and Conclusion

In this section, we implement algorithm for solving system of inequalities in MATLAB7.0 in order to demonstrate the behavior of the algorithm. Due to page limit, we omit to test the effectiveness of our method in generating highly quadratic programming problems.

Example 1 Consider

\[
\begin{align*}
\min z &= x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 + x_3 \\
\text{s.t.} & \quad x_1 + x_2 + x_3 = 4 \\
& \quad 2x_1 - x_2 + x_3 = 2 \\
& \quad x_1 - 2x_2 - x_3 \leq 4 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]

The exact optimal solution is \( x^* = \left( \begin{array}{c} 21 \\ 11 \\ 22 \end{array} \right) \), and the optimal value of objective function is \( z^* = \frac{175}{44} \).

Example 2 Consider

\[
\begin{align*}
\max z &= x_1^2 + x_2^2 + x_3^2 \\
\text{s.t.} & \quad 3x_1 + 2x_2 + x_3 = 6 \\
& \quad 2x_1 - 4x_2 + x_3 \leq -8 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]

The exact optimal solution is \( x^* = \left( \begin{array}{c} 1 \\ 9 \\ -4 \end{array} \right) \), and the optimal value of objective function is \( z^* = \frac{55}{4} \).

The results are shown in Table 1.

From the table, we can see that the algorithm is very simple and only applies the traditional fixed-point method. The implementation of this algorithm is rather easy.

Example 3 Consider

\[
\min \frac{1}{2} x^T Qx + c^T x
\]
Table 1. Experiments and calculated data.

<table>
<thead>
<tr>
<th>Example</th>
<th>Optimal Solution</th>
<th>Objective Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\begin{bmatrix} 1.909091 \ 1.954546 \ 0.136364 \ 0.500000 \ 2.250000 \ 0.000000 \end{bmatrix}$</td>
<td>3.977273</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>13.750000</td>
</tr>
</tbody>
</table>

Table 2. Experiments and calculated data.

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Time</th>
<th>Objective Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5.2s</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

And

$$Ax \geq b$$

And

$$Q = \begin{bmatrix} 100.0429 & 0.9681 & 1.0672 & 1.4239 & 0.2477 & 0.7917 \\ 0.9681 & 100.0988 & 1.1956 & 0.7964 & 0.50000 & 0.00000 \\ 1.0672 & 1.1956 & 100.0919 & 0.7258 & 0.3490 & 4.7373 \\ 1.4239 & 0.7964 & 0.7258 & 101.7046 & 1.1653 & 0.8368 \\ 0.2477 & 0.3543 & 0.3490 & 1.1653 & 101.3110 & 1.1222 \\ 0.7917 & 0.8045 & 1.2315 & 0.8368 & 1.1122 & 100.3080 \end{bmatrix}$$


$$c = (4.5647 & 0.1850 & 8.2141 & 4.4470 & 6.1543 & 7.9194)^T$$

$$b = (-4.6091 & -3.6910 & -0.8813 & -2.0285)^T$$

The exact optimal solution is

$$x^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

and the optimal value of objective function is $z^* = 0$.

Applying the method proposed, the iterations consuming time are shown in Table 2.

Acknowledgements

This work is supported by the Beijing Municipal Commission of Education Science and Technology Program (KM201310017006) and the Beijing URT Program (2012J00018).

REFERENCES


