一类二次规划逆问题的交替方向数值方法*

卢 越[1] 张继宏[1] 张立卫[1,†]

摘要 考虑求解一类二次规划逆问题的交替方向数值算法. 首先给出矩阵变量子问题解的显示表达式，而后构造了两个求解向量变量子问题近似解的数值算法，其中一个算法基于不动点原理，另一算法则应用半光滑牛顿法. 数值实验表明，所提出的算法能够快速高效地求解二次规划逆问题.

关键词 逆问题，交替方向法，二次规划，半光滑牛顿法

中图分类号 O221

2010 数学分类号 90C20, 90C25

An alternating direction numerical method for a type of inverse quadratic programming problem*

LU Yue[1] ZHANG Jihong[1] ZHANG Liwei[1,†]

Abstract An alternating direction numerical method for a type of inverse quadratic programming problem is considered, we first give an explicit formula of the solution to the matrix-variable sub-problem, and provide two algorithms for finding an approximate solution to the vector-variable subproblem. One of these two algorithms is based on the fixed point theorem and the other is a semi-smooth Newton method. Numerical experiments show the efficiency and effectiveness of the proposed algorithm for inverse quadratic programming problems.

Keywords inverse problem, alternating direction method, quadratic programming, semi-smooth Newton method

Chinese Library Classification O221

2010 Mathematics Subject Classification 90C20, 90C25

0 引 言

最优化问题往往由一些参数定义，通常的优化问题指在这些参数已知的前提下，求出问题的最优解，这样的问题可以被称为正问题. 最优化中的问题大都是这样的正问题，比如线性规划、二次规划、非线性规划、几何规划、二阶锥规划、半定规划、矩阵锥优化、随机规划、均衡约束优化，等等.

然而在现实中，有大量的最优化问题，其参数并非完全已知，只知道某些估计值，可

收稿日期：2012年11月6日

* 基金项目：国家自然科学基金 (Nos. 91130007, 91330206)

1. 大连理工大学数学科学学院, 辽宁大连 116024; Institute of Operations Research and Control Theory, School of Mathematics Sciences, Dalian University of Technology, Dalian 116024, Liaoning, China

† 通讯作者 Corresponding author, Email: lwzhang@dlut.edu.cn

以通过理论研究和现实观察得到原问题的最优解，需要在这些条件下，确定优化问题的参数，这就是逆优化问题.

20世纪80年代，逆问题首先得到了地球物理学家的重视[1]，将其应用到对地震运动的预测. 在最优化领域中，1992年和1994年Burton和Toint[2,3]在Mathematical Programming上发表了两篇论文，对地震层析成像的最短路逆问题进行了研究，带动了组合优化领域的长足发展. 比如, 可以将其应用到投资组合优化[4,5]、生成树逆问题[6]和网络流逆问题[7]等. 组合优化逆问题的详细论述可以参考Heuberger[8]在2004年的综述报告. 在连续优化领域，Zhang和Liu[9,10]研究了一类线性规划逆问题，将线性规划逆问题转化为一个新的线性规划问题并对之进行求解. Ahuja和Orlin[11]将对偶理论应用到逆问题研究中. 目前，连续优化的逆问题的数值算法的研究取得重要进展，Zhang和他的合作者们比较早地讨论目标函数为1模和∞模的线性规划逆问题. 针对二次规划逆问题，文献[12]发现这一问题的对偶问题是一SC1的凸优化问题 (即目标函数是连续可微函数，它的梯度是半光滑的映射)，他们用增广Lagrange方法求解该对偶问题，并给出增广Lagrange方法的收敛速度的分析，这一分析涉及投影算子$\Pi_{\mathcal{S}_+^n}$的微分性质. 文献[13]给出这一对偶问题的光滑牛顿法. 关于半定规划逆问题和半定约束二次规划逆问题的的研究，Xiao[14]给出了较为完整的理论分析和数值实验.

尽管上面的工作涉及了线性规划逆问题、二次规划逆问题、半定规划逆问题和半定约束二次规划逆问题的研究. 但数值算法均为增广Lagrange方法和光滑牛顿方法，两种算法都是对对偶问题进行求解得到的，而且涉及到半正定对称矩阵锥的投影的微分性质等非光滑矩阵分析的专门知识，不利于实际工程技术人员理解和使用. 基于这样的认识，本文将构造一类二次规划逆问题的交替方向方法.

本文的结构如下：第1部分将介绍求解二次规划逆问题的交替方向方法的结构，第2部分讨论算法的终止准则. 关于子问题的求解算法将在第3部分给出，第4部分是数值实验，最后给出结论.

0.1 符号说明

在本文中，设$\mathbb{R}^{m\times n}$表示$m\times n$矩阵组成的空间. $N_{\mathcal{S}_+^n}(\cdot)$ 表示集合$\mathcal{S}_+^n$的法锥, 它的定义如下

$$N_{\mathcal{S}_+^n}(X) := \{Y : \langle Y, X' - X\rangle \leqslant 0, \quad \text{对任意的 } X' \in \mathcal{S}_+^n\}.$$

类似地，$N_{\mathcal{S}_+^n\times\mathcal{S}_+^{m-r}}(G^*,\Gamma^*)$ 表示集合$\mathcal{S}_+^n\times\mathcal{S}_+^{m-r}$在 (G^*,Γ^*) 处的法锥. 由集合$\mathcal{S}^n$到半正定对称矩阵锥$\mathcal{S}_+^n$的投影记为$\Pi_{\mathcal{S}_+^n}(\cdot)$, 它是下述优化问题的最优解

$$\begin{aligned} \min \quad & \|X - Y\|_F^2 \\ \text{s. t.} \quad & X \in \mathcal{S}_+^n. \end{aligned}$$

1 交替方向法

考虑下述二次规划问题

$$\mathrm{QP}(G,c)\quad \begin{array}{ll}\min & Q(x)=c^{\mathrm{T}}x+\frac{1}{2}x^{\mathrm{T}}Gx\\ \text{s. t.} & Ax\geqslant b,\\ & x\in\mathbb{R}^n.\end{array}$$

其中$A\in\mathbb{R}^{m\times n},b\in\mathbb{R}^m$是给定的矩阵和向量，$G\in\mathcal{S}_+^n,c\in\mathbb{R}^n$是要确定的对称矩阵和向量.

给定$G^0\in\mathcal{S}_+^n,c^0\in\mathbb{R}^n$，它们是参考矩阵和向量，通过实验或观察得到. 给定向量$x^0\in\mathbb{R}^n$满足$Ax^0\geqslant b$，这里考虑的二次规划逆问题是确定$(G,c)\in\mathcal{S}_+^n\times\mathbb{R}^n$，满足向量$x^0$是$\mathrm{QP}(G,c)$的解，使$(G,c)$与$(G^0,c^0)$的距离最近. 记$x^0\in\mathrm{SOL}(\mathrm{QP}(G,c))$，其中$\mathrm{SOL}(\mathrm{QP}(G,c))$表示$\mathrm{QP}(G,c)$的最优解集，因此问题可表述如下

$$(\mathrm{IQP})\quad \begin{array}{ll}\min & \frac{1}{2}||G-G^0||_F^2+\frac{1}{2}||c-c^0||^2\\ \text{s. t.} & x^0\in\mathrm{SOL}(\mathrm{QP}(G,c)),\\ & G\in\mathcal{S}_+^n.\end{array}$$

其中$||\cdot||_F$和$||\cdot||$分别表示矩阵的Frobenius范数 (即矩阵全部元素的平方和的平方根) 和向量的欧式范数. 由于$G\in\mathcal{S}_+^n$，问题$\mathrm{QP}(G,c)$是一严格凸的二次规划问题，因此逆问题 (IQP) 的约束$x^0\in\mathrm{SOL}(\mathrm{QP}(G,c))$等价于下述KKT条件

$$\begin{cases}c+Gx^0-A^{\mathrm{T}}l=0,\\ 0\leqslant l\perp Ax^0-b\geqslant 0.\end{cases}\tag{1.1}$$

不妨假设

$$I(x^0)=\{i:a_i^{\mathrm{T}}x^0=b_i\}=\{1,2,\cdots,m_0\},$$

记$A_0=(a_1,a_2,\cdots,a_{m_0})$，则 (1.1) 等价表示为

$$\begin{cases}c+Gx^0-A_0^{\mathrm{T}}u=0,\\ 0\leqslant u\in\mathbb{R}^{m_0}.\end{cases}\tag{1.2}$$

在这样的记号之下，逆问题 (IQP) 可简化成下述形式

$$\begin{array}{ll}\min & \frac{1}{2}||G-G^0||_F^2+\frac{1}{2}||A_0^{\mathrm{T}}u-Gx^0-c^0||^2\\ \text{s. t.} & 0\leqslant u\in\mathbb{R}^{m_0},\ G\in\mathcal{S}_+^n.\end{array}\tag{1.3}$$

本论文的主要目的是给出一交替方向方法求解问题 (1.3). 如果求出 (1.3) 的最优解是(G^*,u^*)，则$(G^*,A_0^{\mathrm{T}}u^*-G^*x^0)$即为 (IQP) 的解.

问题 (1.3) 可以通过引入辅助矩阵$Z\in\mathcal{S}_+^n$等价地表示为

$$\begin{array}{ll}\min & \frac{1}{2}||G-G^0||_F^2+\frac{1}{2}||A_0^{\mathrm{T}}u-Zx^0-c^0||^2\\ \text{s. t.} & Z-G=0,\ 0\leqslant u\in\mathbb{R}^{m_0},\ G\in\mathcal{S}_+^n.\end{array}\tag{1.4}$$

问题 (1.4) 的增广Lagrange函数为

$$L_\beta(G,u,Z,\Gamma)=\frac{1}{2}||G-G^0||_F^2+\frac{1}{2}||A_0^{\mathrm{T}}u-Zx^0-c^0||^2+\langle\Gamma,Z-G\rangle+\frac{\beta}{2}||Z-G||_F^2.$$

下面的交替方向方法是基于增广Lagrange函数在两组变量的极小化问题上交替求解的思想设计的.

交替方向法

步0 取 $(u^1,Z^1,\Gamma^1)\in\mathbb{R}^{m_0}\times\mathcal{S}_+^n\times\mathcal{S}^n$，置$k=1$.

步1 计算

$$G^{k+1}=\mathrm{argmin}\left\{L_\beta(G,u^k,Z^k,\Gamma^k):G\in\mathcal{S}_+^n\right\}. \tag{1.5}$$

步2 计算

$$(u^{k+1},Z^{k+1})=\mathrm{argmin}\left\{L_\beta(G^{k+1},u,Z,\Gamma^k):u\geqslant 0\right\}. \tag{1.6}$$

步3 计算

$$\Gamma^{k+1}=\Gamma^k+\beta(Z^{k+1}-G^{k+1}).$$

步4 若 $(G^{k+1},u^{k+1},Z^{k+1},\Gamma^{k+1})$ 满足终止准则，则终止计算.

步5 更新β，置$k:=k+1$，转到步1.

注1.1 上述步4中没有给出具体的终止准则，问题 (1.5) 中的G^{k+1}的求解方法与问题 (1.6) 中的求解方法也没有具体给出. 在具体算法实现中我们要逐项讨论. 上述方法还只是一概念性方法.

后面的讨论表明，求解问题 (1.5) 不困难，它有显式的表达式，但问题 (1.6) 的求解不是一个容易的问题，需要设计具体的算法. 我们将给出两个算法求解这一问题，一个算法基于不动点原理属于简单迭代方法; 另一方法则是半光滑牛顿法.

2 终止准则的讨论

由于问题 (1.3) 是一凸优化问题，设 (G^*,u^*) 是这一问题的解，则下述条件是解的充分必要条件

$$0\in D\varphi(G^*,u^*)+N_{\mathcal{S}_+^n\times\mathbb{R}_+^{m_0}}(G^*,u^*), \tag{2.1}$$

其中

$$\varphi(G,u)=\frac{1}{2}||G-G^0||_F^2+\frac{1}{2}||A_0^{\mathrm{T}}u-Gx^0-c^0||^2$$

是问题 (1.3) 的目标函数, $D\varphi(G^*,u^*)$ 是φ在 (G^*,u^*) 点处的导数，有

$$D\varphi(G^*,u^*)=(D_G\varphi(G^*,u^*),D_u\varphi(G^*,u^*)).$$

通过简单计算，我们可以得到

$$D_G\varphi(G^*,u^*)=G-G^0+\frac{(c^0+Gx^0-A_0^{\mathrm{T}}u)(x^0)^{\mathrm{T}}+(x^0)(c^0+Gx^0-A_0^{\mathrm{T}}u)^{\mathrm{T}}}{2},$$
$$D_u\varphi(G^*,u^*)=(A_0^{\mathrm{T}}u-Gx^0-c^0)^{\mathrm{T}}A_0^{\mathrm{T}}=\nabla_u\varphi(G,u)^{\mathrm{T}},$$

则关系式 (2.1) 等价于

$$0 \in G^* - G^0 + \frac{(c^0 + G^*x^0 - A_0^{\mathrm{T}}u^*)(x^0)^{\mathrm{T}} + (x^0)(c^0 + G^*x^0 - A_0^{\mathrm{T}}u^*)^{\mathrm{T}}}{2} + N_{\mathcal{S}_+^n}(G^*),$$
$$0 \in A_0(A_0^{\mathrm{T}}u^* - G^*x^0 - c^0) + N_{\mathbb{R}_+^{m_0}}(u^*).$$

由投影算子和其对应法锥的关系，上述包含关系可表达为

$$G^* - \Pi_{\mathcal{S}_+^n}\left(G^0 - \frac{(c^0 + G^*x^0 - A_0^{\mathrm{T}}u^*)(x^0)^{\mathrm{T}} + (x^0)(c^0 + G^*x^0 - A_0^{\mathrm{T}}u^*)^{\mathrm{T}}}{2}\right) = 0,$$
$$u^* - \Pi_{\mathbb{R}_+^{m_0}}\left(u^* - A_0(A_0^{\mathrm{T}}u^* - G^*x^0 - c^0)\right) = 0.$$

可见步4 的终止准则可选取为

$$\max\left\{r_G^{k+1}, r_u^{k+1}\right\} \leqslant \varepsilon,$$

其中

$$r_G^{k+1} = ||G^{k+1} - \Pi_{\mathcal{S}_+^n}\left(G^0 - TG^{k+1}\right)||_F,$$
$$TG^{k+1} = \frac{(c^0 + G^{k+1}x^0 - A_0^{\mathrm{T}}u^{k+1})(x^0)^{\mathrm{T}} + (x^0)(c^0 + G^{k+1}x^0 - A_0^{\mathrm{T}}u^{k+1})^{\mathrm{T}}}{2},$$
$$r_u^{k+1} = ||u^{k+1} - \Pi_{\mathbb{R}_+^{m_0}}\left(u^{k+1} - A_0(A_0^{\mathrm{T}}u^{k+1} - G^{k+1}x^0 - c^0)\right)||. \tag{2.2}$$

3 子问题的求解

首先我们考虑求解G^{k+1}的表达式. 由上面的交替方向法知，G^{k+1}为下述问题的最优解

$$\begin{aligned} \min \quad & L_\beta(G, u^k, Z^k, \Gamma^k) \\ \text{s. t.} \quad & G \in \mathcal{S}_+^n, \end{aligned}$$

则有

$$0 \in [D_G L_\beta(G, u^k, Z^k, \Gamma^k)]\,|_{G=G^{k+1}} + N_{\mathcal{S}_+^n}(G^{k+1}),$$

其中

$$[D_G L_\beta(G, u^k, Z^k, \Gamma^k)]\,|_{G=G^{k+1}} = G^{k+1} - G^0 - \Gamma^k - \beta(Z^k - G^{k+1}),$$

因此

$$G^{k+1} = \Pi_{\mathcal{S}_+^n}\left(\frac{1}{1+\beta}(G^0 + \Gamma^k + \beta Z^k)\right). \tag{3.1}$$

通过 (3.1) 很容易得到G^{k+1}的值. 设$\frac{1}{1+\beta}(G^0 + \Gamma^k + \beta Z^k)$的谱分解是

$$\frac{1}{1+\beta}(G^0 + \Gamma^k + \beta Z^k) = P^k \Lambda^k (P^k)^{\mathrm{T}},$$

其中P^k是正交矩阵，对角矩阵$\Lambda^k = \mathrm{diag}(\lambda_1^k, \lambda_2^k, \cdots, \lambda_n^k)$，其中$\lambda_j^k\ (j = 1, 2, \cdots, n)$是$\frac{1}{1+\beta}(G^0 + \Gamma^k + \beta Z^k)$的$n$个特征值. 定义

$$\Lambda_+^k = \mathrm{diag}([\lambda_1^k]_+, [\lambda_2^k]_+, \cdots, [\lambda_n^k]_+),\ [a]_+ = \max(a, 0),$$

则

$$G^{k+1} = P^k \Lambda_+^k (P^k)^{\mathrm{T}}, \tag{3.2}$$

即问题 (1.5) 的解.

下面我们应用二次规划来求解 (u^{k+1}, Z^{k+1}). 由于 (u^{k+1}, Z^{k+1}) 是问题 (1.6) 的解，可将 (1.6) 表示为

$$\min_{u \geqslant 0} \min_{Z} \ L_\beta(G^{k+1}, u, Z, \Gamma^k). \tag{3.3}$$

引入记号

$$Z_k(u) = \mathrm{argmin}_Z \ L_\beta(G^{k+1}, u, Z, \Gamma^k),$$

则问题 (3.3) 可表示为

$$\min_{u \geqslant 0} \ f_k(u), \tag{3.4}$$

其中

$$f_k(u) = L_\beta(G^{k+1}, u, Z_k(u), \Gamma^k). \tag{3.5}$$

注意，对固定的$u, Z \mapsto L_\beta(G^{k+1}, u, Z, \Gamma^k)$ 是凸函数，有$Z_k(u)$ 是下述方程组的解

$$D_z L_\beta(G^{k+1}, u, Z_k(u), \Gamma^k) = 0.$$

为了方便推导，记$\phi_k(Z, u) = L_\beta(G^{k+1}, u, Z, \Gamma^k)$，则有$D_z \phi_k(Z_k(u), u) = 0$，从而

$$\begin{aligned} &\Gamma^k - \beta G^{k+1} + \frac{(c^0 - A_0^{\mathrm{T}} u)(x^0)^{\mathrm{T}} + (x^0)(c^0 - A_0^{\mathrm{T}} u)^{\mathrm{T}}}{2} \\ &+\beta Z_k(u) + \frac{x^0 (x^0)^{\mathrm{T}} Z_k(u) + Z_k(u) x^0 (x^0)^{\mathrm{T}}}{2} = 0. \end{aligned} \tag{3.6}$$

由此解得

$$Z_k(u) = -L^{-1}_{(x^0(x^0)^{\mathrm{T}} + \beta I)} \left(\Gamma^k - \beta G^{k+1} + L_{x^0}(c^0 - A_0^{\mathrm{T}} u) \right), \tag{3.7}$$

其中

$$L_A(B) = \frac{1}{2}(AB + BA), \ A, B \in \mathcal{S}^n, \ L_w(\xi) = \frac{1}{2}(w\xi^{\mathrm{T}} + \xi w^{\mathrm{T}}), \ w, \xi \in \mathbb{R}^n.$$

由$f_k(u)$的定义可得$f_k(u) = \phi_k(Z_k(u), u)$，则$\nabla f_k(u)$可以通过下面的计算得到

$$\nabla f_k(u) = \nabla_u \phi_k(Z, u) \mid_{Z = Z_k(u)} + DZ_k(u)^* D_z \phi_k(Z, u)^* \mid_{Z = Z_k(u)},$$

其中$*$表示伴随运算. 由于$D_z \phi_k(Z_k(u), u) = 0$，有

$$\nabla f_k(u) = \nabla_u \phi_k(Z, u) \mid_{Z = Z_k(u)} = A_0(A_0^{\mathrm{T}} u - c^0) - A_0 Z_k(u) x^0. \tag{3.8}$$

定理 3.1 对于$u \in \mathbb{R}^n, \nabla^2 f_k(u)$ 是半正定矩阵，从而f_k是凸函数.

证明 我们计算$\nabla^2 f_k(u)$，为此对任何$h \in \mathbb{R}^{m_0}$，计算$\nabla^2 f_k(u)h$. 根据 (3.8)，有

$$\nabla^2 f_k(u)h = A_0 A_0^{\mathrm{T}} h - A_0 Z_k'(u)hx^0, \tag{3.9}$$

下面只需计算$A_0 Z_k'(u)hx^0$即可. 由 (3.6)，有

$$\begin{aligned}\Gamma^k - \beta G^{k+1} + \frac{(c^0 - A_0^{\mathrm{T}}(u+h))(x^0)^{\mathrm{T}} + (x^0)(c^0 - A_0^{\mathrm{T}}(u+h))^{\mathrm{T}}}{2} \\ +\beta Z_k(u+h) + \frac{x^0(x^0)^{\mathrm{T}} Z_k(u+h) + Z_k(u+h)x^0(x^0)^{\mathrm{T}}}{2} = 0.\end{aligned} \tag{3.10}$$

用式 (3.10) 减掉 (3.6) 得到

$$\begin{aligned}\frac{-A_0^{\mathrm{T}} h(x^0)^{\mathrm{T}} - x^0 h^{\mathrm{T}} A_0}{2} + \beta[Z_k(u+h) - Z_k(u)] \\ +\frac{x^0(x^0)^{\mathrm{T}}(Z_k(u+h) - Z_k(u)) + (Z_k(u+h) - Z_k(u))x^0(x^0)^{\mathrm{T}}}{2} = 0.\end{aligned}$$

注意$Z_k(\cdot)$是线性映射，$Z_k(u+h) - Z_k(u) = Z_k'(u)h$，上式即

$$\frac{-A_0^{\mathrm{T}} h(x^0)^{\mathrm{T}} - x^0 h^{\mathrm{T}} A_0}{2} + \beta Z_k'(u)h + \frac{x^0(x^0)^{\mathrm{T}} Z_k'(u)h + Z_k'(u)hx^0(x^0)^{\mathrm{T}}}{2} = 0. \tag{3.11}$$

对 (3.11) 的两端左乘A_0，右乘x^0可得

$$\begin{aligned}-\frac{(x^0)^{\mathrm{T}} x^0 A_0 A_0^{\mathrm{T}} h + A_0 x^0 h^{\mathrm{T}} A_0 x^0}{2} + \beta A_0 Z_k'(u)hx^0 \\ +\frac{A_0 x^0(x^0)^{\mathrm{T}} Z_k'(u)hx^0 + A_0 Z_k'(u)hx^0(x^0)^{\mathrm{T}} x^0}{2} = 0,\end{aligned}$$

即

$$\begin{aligned}((x^0)^{\mathrm{T}} x^0)A_0 A_0^{\mathrm{T}} h + (A_0 x^0)(A_0 x^0)^{\mathrm{T}} h = 2\beta(A_0 Z_k'(u)hx^0) + ((x^0)^{\mathrm{T}} x^0)(A_0 Z_k'(u)hx^0) \\ +A_0 x^0(x^0)^{\mathrm{T}} Z_k'(u)hx^0.\end{aligned} \tag{3.12}$$

对 (3.11) 的两端左乘$(x^0)^{\mathrm{T}}$，右乘x^0可得

$$((x^0)^{\mathrm{T}} x^0)(x^0)^{\mathrm{T}} A_0^{\mathrm{T}} h = (x^0)^{\mathrm{T}} Z_k'(u)hx^0\beta + ((x^0)^{\mathrm{T}} x^0)(x^0)^{\mathrm{T}} Z_k'(u)hx^0,$$

于是有

$$(x^0)^{\mathrm{T}} Z_k'(u)hx^0 = \frac{x^0(x^0)^{\mathrm{T}}}{(x^0)^{\mathrm{T}} x^0 + \beta}(x^0)^{\mathrm{T}} A_0^{\mathrm{T}} h. \tag{3.13}$$

将 (3.13) 代入 (3.11) 可得

$$\begin{aligned}[2\beta + (x^0)^{\mathrm{T}} x^0](A_0 Z_k'(u)hx^0) &= ((x^0)^{\mathrm{T}} x^0)A_0 A_0^{\mathrm{T}} h + \Big(1 - \frac{x^0(x^0)^{\mathrm{T}}}{(x^0)^{\mathrm{T}} x^0 + \beta}\Big)(Ax^0)(Ax^0)^{\mathrm{T}} h \\ &= ((x^0)^{\mathrm{T}} x^0)A_0 A_0^{\mathrm{T}} h + \frac{\beta}{(x^0)^{\mathrm{T}} x^0 + \beta}(Ax^0)(Ax^0)^{\mathrm{T}} h,\end{aligned}$$

从而有

$$A_0 Z_k'(u)hx^0 = \frac{1}{2\beta+(x^0)^{\mathrm{T}}x^0}\left[(x^0)^{\mathrm{T}}x^0 A_0A_0^{\mathrm{T}} + \frac{\beta}{(x^0)^{\mathrm{T}}x^0+\beta}(A_0x^0)(A_0x^0)^{\mathrm{T}}\right]h.$$

于是由 (3.9) 可得

$$\begin{aligned}\nabla^2 f_k(u)h &= \left\{\left(1-\frac{(x^0)^{\mathrm{T}}x^0}{2\beta+(x^0)^{\mathrm{T}}x^0}\right)A_0A_0^{\mathrm{T}} - \frac{\beta}{(2\beta+(x^0)^{\mathrm{T}}x^0)(\beta+(x^0)^{\mathrm{T}}x^0)}(A_0x^0)(A_0x^0)^{\mathrm{T}}\right\}h\\ &= \frac{2\beta}{2\beta+(x^0)^{\mathrm{T}}x^0}\left[A_0A_0^{\mathrm{T}} - \frac{1}{2(\beta+(x^0)^{\mathrm{T}}x^0)}(A_0x^0)(A_0x^0)^{\mathrm{T}}\right]h\\ &= \frac{2\beta}{2\beta+(x^0)^{\mathrm{T}}x^0}A_0\left[I - \frac{1}{2(\beta+(x^0)^{\mathrm{T}}x^0)}x^0(x^0)^{\mathrm{T}}\right]A_0^{\mathrm{T}}h.\end{aligned}$$

所以有

$$\nabla^2 f_k(u) = \frac{2\beta}{2\beta+(x^0)^{\mathrm{T}}x^0}A_0\left[I - \frac{1}{2(\beta+(x^0)^{\mathrm{T}}x^0)}x^0(x^0)^{\mathrm{T}}\right]A_0^{\mathrm{T}}. \tag{3.14}$$

注意对称矩阵$I - \dfrac{1}{2(\beta+(x^0)^{\mathrm{T}}x^0)}x^0(x^0)^{\mathrm{T}}$有$n-1$个1特征值和最小特征值

$$1 - \frac{||x^0||^2}{2(\beta+||x^0||^2)} > 0,$$

这一矩阵是正定对称矩阵，可见由 (3.14) 知$\nabla^2 f_k(u)$是与u无关的半正定矩阵，$f_k(u)$是一凸函数.

注3.1 由于$Z_k(u)$是u的线性函数，$f_k(u)$是二次函数，上述定理表明这一二次函数还是凸函数，其Hesse 阵由公式 (3.14) 给出. 现在给出$f_k(u)$的具体表达式. 因极小化$f_k(u)$与常数项无关，无需给出常数项的值，不妨设$f_k(u)$的表达式如下

$$f_k(u) = \frac{1}{2}u^{\mathrm{T}}Hu + (p^k)^{\mathrm{T}}u + \mathrm{const},$$

其中H的表达式即 (3.14) 中右端的矩阵，即

$$H = \frac{2\beta}{2\beta+(x^0)^{\mathrm{T}}x^0}A_0\left[I - \frac{1}{2(\beta+(x^0)^{\mathrm{T}}x^0)}x^0(x^0)^{\mathrm{T}}\right]A_0^{\mathrm{T}}. \tag{3.15}$$

下面给出p^k的表达式. 注意到$p^k = \nabla f_k(0)$，由 (3.8) 得

$$p^k = -A_0c^0 - A_0Z_k(0)x^0, \tag{3.16}$$

只需要计算$A_0Z_k(0)x^0$，即得到p^k的值. 仿照上述定理中的证明，可得到

$$A_0Z_k(0)x^0 = \frac{-2}{2\beta+(x^0)^{\mathrm{T}}x^0}A_0\left[I - \frac{1}{2(\beta+(x^0)^{\mathrm{T}}x^0)}x^0(x^0)^{\mathrm{T}}\right]W^kx^0,$$

其中W^k的定义如下

$$W^k = \Gamma^k - \beta G^{k+1} + \frac{c^0(x^0)^{\mathrm{T}} + x^0(c^0)^{\mathrm{T}}}{2}. \tag{3.17}$$

于是由 (3.16) 得到

$$p^k = -A_0c^0 + \frac{2}{2\beta+(x^0)^{\mathrm{T}}x^0}A_0\left[I - \frac{1}{2(\beta+(x^0)^{\mathrm{T}}x^0)}x^0(x^0)^{\mathrm{T}}\right]W^kx^0. \tag{3.18}$$

推论 3.1 函数$f_k(u)$是一二次凸函数，它的一次项系数向量p^k由(3.18)给出，Hesse阵H由(3.15)给出. 如果A_0是行满秩矩阵，则函数$f_k(u)$是强凸二次函数.

所以u^{k+1}是下述简单凸二次极小化问题的解

$$\begin{array}{ll}\min & \frac{1}{2}u^{\mathrm{T}}Hu+(p^k)^{\mathrm{T}}u\\ \text{s. t.} & u\geqslant 0.\end{array}\tag{3.19}$$

而$Z^{k+1}=Z_k(u^{k+1})$由方程(3.6)确定，或由(3.7)给出.

下面我们讨论问题(3.19)的解法. 为了简化讨论，我们往往作下述假设：A_0是行满秩的矩阵. 在这一假设下，$f_k(u)$是一强凸二次函数.

首先，我们给出简单迭代方法. 如果A_0是行满秩的矩阵，问题(3.19)等价于下述线性互补问题

$$0\leqslant u\perp(Hu+p^k)\geqslant 0.\tag{3.20}$$

(3.20)等价于

$$0\in Hu+p^k+N_{\mathbb{R}^{m_0}_+}(u).\tag{3.21}$$

由投影映射和法锥的关系有

$$u-\Pi_{\mathbb{R}^{m_0}_+}(u-\alpha(Hu+p^k))=0,\tag{3.22}$$

其中$\alpha>0$是常数. 设H的最大特征值的上界为$M>0$，它的最小特征值的下界为$m>0$，即

$$m\leqslant\lambda_i(H)\leqslant M,\ \forall i=1,2,\cdots,m_0.\tag{3.23}$$

记

$$F_\alpha(u)=u-\Pi_{\mathbb{R}^{m_0}_+}(u-\alpha(Hu+p^k)).\tag{3.24}$$

定理 3.2 如果条件(3.23)成立，且$\alpha\in(0,\frac{2}{M})$，则$F_\alpha(u)$是一压缩映射，满足

$$||F_\alpha(u)-F_\alpha(u')||\leqslant q||u-u'||,\ \forall u,u'\in\mathbb{R}^{m_0},\tag{3.25}$$

其中$q=\max\{|1-\alpha m|,|1-\alpha M|\}<1$.

证明 证明类似于文献[15]的定理2.3.4.

可以验证，当α选取为

$$\alpha=\frac{2}{M+m}$$

时，q取最小值，为

$$q=\frac{M-m}{M+m}=\frac{\kappa-1}{\kappa+1},$$

其中κ是H的条件数的一个上界，它越接近1，q越接近0. 根据定理3.2，若$\alpha\in(0,\frac{2}{M})$，则$F_\alpha(u)$是一压缩映射，u^{k+1}为问题(3.19)的解，它是$F_\alpha(u)$的不动点. 可采用下述简单迭代格式生成这一不动点.

推论 3.2 如果在步2中不设定终止准则，下述不动点算法生成无穷序列$\{\nu^j\}$，则

$$||\nu^j-u^{k+1}||\leqslant\frac{q^j}{1-q}||\nu^1-\nu^0||.\tag{3.26}$$

不动点算法

步0 计算H, p^k，估计H特征值的上下界$M, m > 0$，置$\nu^0 = u^k, j = 0$，取

$$\alpha = \frac{2}{M+m},$$

步1 计算$\nu^{j+1} = F_\alpha(\nu^j)$.

步2 若ν^{j+1}满足精度，取u^{k+1}的近似解为ν^{j+1}，终止计算.

步3 置$j := j + 1$，转到步1.

根据 (3.26)，只要加大迭代次数，可以使ν^j任意程度地近似u^{k+1}，因此求解u^{k+1}的不动点算法的终止准则很容易选取.

上面的方法虽然简单，方便实现，但其收敛速度是线性的，收敛速度由矩阵H的条件数确定. 接下来我们给出求解 (3.20) 的半光滑牛顿法，其具有局部的二阶收敛速度.

利用NCP函数可以将 (3.20) 转化为下述方程组

$$\Phi_k(u) = \begin{bmatrix} \phi(u_1, (Hu + p^k)_1) \\ \vdots \\ \phi(u_{m_0}, (Hu + p^k)_{m_0}), \end{bmatrix}$$

其中$\phi(\cdot, \cdot)$为NCP函数，满足

$$0 \leqslant a \perp b \geqslant 0 \Leftrightarrow \phi(a, b) = 0.$$

通常$\phi(a, b)$不是光滑函数，而是半光滑函数，例如

$$\phi_{\min}(a, b) = \min\{a, b\}$$

与

$$\phi_{FB}(a, b) = \sqrt{a^2 + b^2} - a - b$$

都是半光滑函数. 实际上$\phi_{\min}$在集合$\{(a, b) : a \neq b\}$上是光滑的，而ϕ_{FB}除了$(0, 0)$点外都是光滑的. 特别地，如果$\Phi_k(u)$在解点u^{k+1}处满足强半光滑性，即

$$\Phi_k(u^{k+1} + h) - \Phi_k(u^{k+1}) - Vh = O(||h||^2), \ \forall V \in \partial\Phi_k(u^{k+1}),$$

且满足强正则性条件，即对任何$V \in \partial\Phi_k(u^{k+1})$均是非奇异的，则可采用半光滑牛顿法 (或广义牛顿方法)，

$$\nu^{j+1} = \nu^j - [V^j]^{-1}\Phi_k(\nu^j), \ V^j \in \partial\Phi_k(\nu^j),$$

生成的序列$\{\nu^j\}$局部二次收敛到u^{k+1}. 这要求初始点ν^0 (通常取为u^k) 距离u^{k+1}很接近才能保证算法的收敛性以及二阶收敛速度. 而事先不能保证初始点ν^0距离u^{k+1}很接近，因此局部算法在实际计算时会遇到困难，为此需要设计全局收敛的半光滑牛顿法.

全局牛顿法保证每次的迭代使势函数的函数值下降，最终迭代点距离u^{k+1}很接近使得全局牛顿法变成局部的步长为1的广义牛顿法. 这一问题的势函数通常取为

$$f_k(u) = \frac{1}{2}||\Phi_k(u)||^2. \tag{3.27}$$

全局牛顿法的迭代格式如下

$$\nu^{j+1}=\begin{cases}\nu^j-\alpha_j[V^j]^{-1}\Phi_k(\nu^j), & \text{若}V^j\text{非奇异},\\ \nu^j-\alpha_j\nabla f_k(\nu^j), & \text{其他},\end{cases}$$

其中α_j是步长. 上述迭代要求$\nabla f_k(\nu^j)$必须存在，对上述的$\phi_{\min}$与ϕ_{FB}而言，ϕ_{FB}确定的$f_k(u)$是光滑的，因此这里我们采用$\phi(\cdot,\cdot)=\phi_{FB}(\cdot,\cdot)$.

以下关于ϕ_{FB}的微分性质由Facchinei和Pang的专著[16]得到.

引理 3.1 对于$(a,b)\in\mathbb{R}\times\mathbb{R}$，有

$$\partial\phi_{FB}(a,b)=\begin{cases}\left\{\left(\dfrac{a}{\sqrt{a^2+b^2}}-1,\dfrac{b}{\sqrt{a^2+b^2}}-1\right)\right\}, & a^2+b^2\neq 0,\\ \left\{(\nu_1-1,\nu_2-1):\nu_1^2+\nu_2^2\leqslant 1\right\}, & a=b=0.\end{cases}$$

引理 3.2 取$\phi(\cdot,\cdot)=\phi_{FB}(\cdot,\cdot)$，则$f_k(u)$是光滑函数，而且

$$\nabla f_k(u)=V^{\mathrm{T}}\Phi_k(u),\ \forall V\in\partial\Phi_k(u).$$

定理 3.3 设A_0是行满秩矩阵，则$V\in\partial\Phi_k(u)$都是非奇异的，即强BD正则性成立. 同时V有如下形式

$$V=D_a+D_bH,$$

其中

$$D_a=\mathrm{diag}(a_1,a_2,\cdots,a_{m_0}),\quad D_b=\mathrm{diag}(b_1,b_2,\cdots,b_{m_0}),$$

向量$(a_i,b_i)\in\partial\phi_{FB}(u_i,(Hu+p^k)_i),(i=1,2,\cdots,m_0)$的定义如下

$$(a_i,b_i)\begin{cases}=\dfrac{(u_i,(Hu+p^k)_i)}{\sqrt{u_i^2+(Hu+p^k)_i^2}}-(1,1), & \text{若}(u_i,(Hu+p^k)_i)\neq(0,0),\\ \in clB(0,1)-(1,1), & \text{其他}.\end{cases}$$

半光滑牛顿法

步0 取$\nu^0=u^k,\rho\in(0,\frac{1}{2}),\sigma\in(0,\frac{1}{8})$，置$j=0$.

步1 选取$V^j\in\partial\Phi_k(\nu^j)$，求$d^j$满足

$$\Phi_k(\nu^j)+V^jd^j=0.$$

步2 确定满足下式的最小的非负整数m_j使得

$$f_k(\nu^j+\rho^md^j)-f_k(v^j)\leqslant\sigma\rho^m\langle\nabla f_k(\nu^j),d^j\rangle.$$

置$t_j=\rho^m,\nu^{j+1}=\nu^j+t_jd^j$.

步3 置$j:=j+1$，转到步1.

注3.2 上述方法没有设定终止准则，实际计算时可选择

$$||\nabla f_k(\nu^j)||\leqslant\varepsilon\quad\text{或}\quad||\Phi_k(\nu^j)||\leqslant\varepsilon,$$

其中$\varepsilon>0$是事先给定的精度. 下面给出上述方法的收敛性定理.

定理 3.4 设A_0是行满秩矩阵，$\{\nu^j\}$是由半光滑牛顿法生成的无穷点列，则$\nu^j \to u^{k+1}$，且局部收敛速度是二阶的.

证明 根据文章[16]的命题9.1.20，存在常数$\mu_0 > 0$满足

$$\|\nu - u^{k+1}\| \leqslant \mu_0 \|\Phi_k(\nu)\|, \quad \forall \nu.$$

由于$f_k(\nu^j)$是递减序列，序列$\{\nu^j\}$是一个有界序列. 设$\{\nu^{j_l}\}$是$\{\nu^j\}$的一个收敛子列，它收敛到$\bar{\nu}$. 根据线搜索原则与d^{j_l}的定义, 可以得到

$$\rho^{m_{j_l}} \nabla f_k(\nu^{j_l})^{\mathrm{T}} d^{j_l} \to 0 (l \to +\infty), \quad \nabla f_k(\nu^{j_l})^{\mathrm{T}} d^{j_l} = -2 f_k(\nu^{j_l}).$$

假设$f_k(\bar{\nu}) \neq 0$, 由上式知$\rho^{m_{j_l}} \to 0$. 根据算法的线搜索原则及中值定理知

$$\int_0^1 \left(\nabla f_k(\nu^{j_l} + t\rho^{m_{j_l}-1} d^{j_l}) - \nabla f_k(\nu^{j_l})\right)^{\mathrm{T}} d^{j_l} \mathrm{d}t + \nabla f_k(\nu^{j_l})^{\mathrm{T}} d^{j_l} > \sigma \nabla f_k(\nu^{j_l})^{\mathrm{T}} d^{j_l}.$$

由$\nabla f_k(\cdot)$的连续性，当$l \to +\infty$时，上式变为$-2 f_k(\bar{\nu}) \geqslant -2\sigma f_k(\bar{\nu})$. 由于假设$f_k(\bar{\nu}) \neq 0$，上式与$\sigma \in (0, \frac{1}{8})$矛盾. 故必有$f_k(\bar{\nu}) = 0$成立. 因为$u^{k+1}$ 是$f_k(u) = 0$的唯一解，有$\bar{\nu} = u^{k+1}$. 可见$\{\nu^j\}$的任何聚点均是u^{k+1}，所以$\nu^j \to u^{k+1}$. 根据$\Phi_k(\nu)$是强BD正则性，可以推出对充分大的j有$\|[V^j]^{-1}\| \leqslant \beta_0$且

$$\|u^{k+1} - \nu^j\| \leqslant \|[V^j]^{-1} \Phi_k(\nu^j)\| + O(\|u^{k+1} - \nu^j\|^2) \leqslant 2\beta_0 \|\Phi_k(\nu^j)\|.$$

根据$\Phi_k(\nu)$的Lipschtiz连续性得到

$$\|\Phi_k(\nu^j + d^j)\| \leqslant O(\|\nu^j + d^j - u^{k+1}\|) = O(\|\nu^j - u^{k+1}\|^2) = O(\|\Phi_k(\nu^j)\|^2).$$

根据σ的选取与上式可得

$$\sigma \nabla f_k(\nu^j)^{\mathrm{T}} d^j = -2\sigma f_k(\nu^j) \geqslant -\frac{1}{2} f_k(\nu^j) \geqslant f_k(\nu^j + d^j) - f_k(\nu^j),$$

结合线搜索原则知$t_j = 1$，即$\nu^{j+1} = \nu^j + d^j$. 由ν^{j+1}的定义知

$$\|u^{k+1} - \nu^{k+1}\| = \|[V^j]^{-1} \Phi_k(\nu^j) - \nu^j + u^{k+1}\|^2 = \|[V^j]^{-1}(\Phi_k(\nu^j) + V^j(u^{k+1} - \nu^j))\|^2.$$

由于$\Phi_k(\nu)$在u^{k+1}附近是强半光滑的，即有

$$0 = \Phi_k(u^{k+1}) = \Phi_k(\nu^j) + V^j(u^{k+1} - \nu^j) + O(\|u^{k+1} - \nu^j\|^2),$$

从而

$$\|\Phi_k(\nu^j) + V^j(u^{k+1} - \nu^j)\| \leqslant \beta_1 \|u^{k+1} - \nu^j\|^2,$$

其中$\beta_1 > 0$是一常数. 由于$\|[V^j]^{-1}\| \leqslant \beta_0$，可以得到

$$\|u^{k+1} - \nu^{k+1}\| \leqslant \beta_0 \beta_1 \|u^{k+1} - \nu^j\|^2,$$

因此$\{\nu^j\}$局部二次收敛到u^{k+1}.

注3.3 本文算法的全局收敛性由综述性文章[17]可直接得到，这篇文章对下述形式问题的交替方向法的理论分析与数值计算进行了讨论：

$$\begin{aligned} \min \quad & f(x)+g(z) \\ \text{s. t.} \quad & Ax+Bz=c, \\ & x\in\mathcal{X}, z\in\mathcal{Z}. \end{aligned} \tag{3.28}$$

显然本文研究的问题 (1.4) 具有上述问题的形式，终止准则的确定也采用文章[16]的分析方法，得到了前文第2部分定义的相关残量r_G^k, r_u^k，其定义见 (2.2).

4 数值实验

本节将对文中给出的算法进行数值实验. 为了进行比较，编写了应用MATLAB自带的quadprog子程序求解u^k子问题的程序，我们将其编号为Method_1. 同时将应用简单迭代法和半光滑牛顿法求解上述子问题的程序分别编号为Method_2和Method_3.

由于逆问题 (1.3) 也可应用内点法来求解，我们应用了国际著名的优化软件CVX[18,19]中的Sedumi[20]和SDPT3[21]子程序对其进行求解，以便与本文提出的算法进行对比.

本文算法的测试环境为：

Ubuntu12.04操作系统，Aoc台式机，内存4G，主频3.3Hz，MATLAB 2009b(Linux版)，CVX 1.22(Unix)版.

算法中的参数设定如下：

(1) 算法中数据$A\in\mathbb{R}^{m\times n}, b\in\mathbb{R}^m, G\in\mathbb{R}^{n\times n}, c\in\mathbb{R}^n, G_0\in\mathbb{R}^{n\times n}, c^0\in\mathbb{R}^n$均为随机生成.

(2) 算法的初始迭代步中u^1, Z^1, Γ^1分别取值如下

$$\begin{aligned} & u_0=\text{ones}(mA_0,1), \text{其中}mA_0\text{为矩阵}A_0\text{的行数}, \\ & \Gamma^1=\text{ones}(n,n), \text{其中}n\text{为矩阵}G\text{的行数}, \\ & Z^1\text{为随机生成的}n\times n\text{对称半正定矩阵}. \end{aligned}$$

(3) 算法的停止准则为$\max(\min(r_G, r_G^k), r_u^k)\leqslant 10^{-3}$ 或 $k_{\max}\geqslant 500$, 其中$r_G=||G^{k+1}-G^k||, r_G^k, r_u^k$定义见 (2.2).

(4) 算法中关于Lagrange函数中参数β的更新准则, 仿照文献[16]中的 (2.13), 定义如下

$$\beta=\begin{cases} 2\beta, & \text{如果}\min(r_G^k, r_G)>10\cdot r_u^k, \\ \beta/2, & \text{如果}\min(r_G^k, r_G)<10\cdot r_u^k, \\ \beta, & \text{其他}. \end{cases}$$

注4.1 上述关于参数β的更新准则是为了得到如文献[17]中交替方向法的收敛性.

表格中的变量说明：

指标 (m,n) 表示矩阵A的行数和列数, 指标$k_i, rgk_i, ruk_i, t_i(i=1,2,3)$ 分别表示Method_i的迭代次数, $\min(r_G^k, r_G)$ 和r_u^k的值及计算时间 (单位为秒), 指标 (t_4,t_5) 表示Sedumi和SDPT3的计算时间 (单位为秒).

本文算法的计算结果如下 (见表1—3).

表 1 低维问题测试

m	n	k_1, k_2, k_3	rgk_1, rgk_2, rgk_3	ruk_1	ruk_2	ruk_3	t_1	t_2	t_3
10	10	11	7.45E-04	4.00E-06	0	6.00E-06	0.127	0.028	0.012
10	20	12	6.81E-04	0	0	1.00E-06	0.075	0.022	0.015
10	30	13	5.21E-04	1.00E-06	0	0	0.200	0.037	0.026
10	40	13	6.99E-04	0	0	2.00E-06	0.162	0.071	0.045
10	50	13	8.63E-04	0	1.00E-06	2.00E-06	0.341	0.088	0.071
10	60	14	5.09E-04	0	0	0	0.260	0.090	0.083
10	70	14	6.06E-04	0	0	0	0.185	0.110	0.109
10	80	14	6.95E-04	0	1.00E-06	0	0.314	0.160	0.142
10	90	14	7.84E-04	0	1.00E-06	0	0.373	0.185	0.178
10	100	14	8.68E-04	0	1.00E-06	0	0.435	0.237	0.222

表 2 高维问题测试1

m	n	k_1, k_2, k_3	rgk_1, rgk_2, rgk_3	ruk_1	ruk_2	ruk_3	t_1	t_2	t_3
100	100	14	8.66E-04	0	3.00E-06	1.00E-06	0.428	0.331	0.234
100	200	15	8.64E-04	0	4.00E-06	1.10E-05	1.343	1.060	0.977
100	300	16	6.48E-04	0	1.20E-05	3.00E-05	3.218	2.958	2.805
100	400	16	8.65E-04	0	1.20E-05	1.60E-05	6.411	6.346	6.211
100	500	17	5.40E-04	0	2.10E-05	1.20E-05	12.193	12.049	11.805
100	600	17	8.63E-04	0	3.10E-05	2.80E-05	21.450	20.245	20.198
100	700	17	7.54E-04	0	2.00E-05	3.00E-06	40.192	39.830	39.733
100	800	17	8.67E-04	0	1.50E-05	5.00E-06	65.877	65.330	65.151
100	900	17	8.63E-04	0	2.80E-05	2.10E-05	105.555	103.715	101.823
100	1000	18	4.31E-04	0	2.00E-05	3.00E-06	164.208	162.760	160.489

表 3 高维问题测试2

m	n	k_1, k_2, k_3	rgk_1, rgk_2, rgk_3	ruk_1	ruk_2	ruk_3	t_1	t_2	t_3
500	500	17	4.41E-04	0	3.70E-05	0	12.09	11.74	10.68
500	600	17	5.58E-04	0	5.20E-05	0	25.65	23.83	21.86
500	700	17	4.09E-04	1.00E-06	4.50E-05	1.00E-06	41.83	40.26	39.83
500	800	17	7.76E-04	0	3.80E-05	0	68.73	67.34	66.57
500	900	17	8.93E-04	0	8.90E-05	2.00E-06	112.89	111.97	110.62
500	1000	17	6.81E-04	1.00E-06	1.12E-04	3.00E-06	171.54	170.65	169.04

为了与本文算法相比较，应用CVX中的程序Sedumi和SDPT3求解问题 (1.3)，计算结果如下表所示，其中F表示计算失败，这里的计算失败包括程序无法求解和计算时间超过定义的$t_{\max} = 200$(秒) 这两种情况.

表 4 Sedumi和SDPT3测试

m	n	t_4	t_5	m	n	t_4	t_5
10	10	0.161	0.315	10	70	58.772	7.811
10	20	0.211	0.396	10	80	124.314	16.426
10	30	0.485	0.677	10	90	F	22.379
10	40	1.356	1.377	10	100	F	33.867
10	50	6.867	2.087	100	100	F	38.943
10	60	18.731	4.064	100	200	F	F

通过上面的计算结果 (见表1—3)，不难发现，三类算法的迭代次数和最终 $\min(r_G^k, r_G)$ 值均相同，r_u^k的值也相当接近. 关于计算时间，我们有以下几点说明：

(1) 由表1不难发现，在求解低维问题时，本文设计的两类求解子问题的算法与MATLAB自带的quadprog相比能够更快的求解子问题. 在求解高维问题时，Method_3相比其他两种算法更具竞争力 (见表2和表3);

(2) 与内点法程序Sedumi和SDPT3的计算结果相比 (见表4)，本文算法均具有较高的计算效率，并且都能够求解高维数的问题;

(3) 应用MATLAB自带的quadprog程序求解子问题会极大地降低算法实现的难度，整个算法的MATLAB代码仅仅需几行，比以前的方法要简短的多;

(4) 应用交替方向法求解问题 (1.4) 除本文算法外，还存在下述框架，即将本文算法的步1和步2替换为下述步骤,

步1 计算 $(G^{k+1}, u^{k+1}) = \text{argmin}\left\{L_\beta(G, u, Z^k, \Gamma^k) : G \in \mathcal{S}_+^n, u \geqslant 0\right\}$,

步2 计算 $Z^{k+1} = \text{argmin}\left\{L_\beta(G^{k+1}, u^{k+1}, Z, \Gamma^k)\right\}$.

此时G方向和Z方向均有显示解，u方向转化为如 (3.19) 的简单二次规划问题，这一框架使理论分析得到了简化. 然而我们应用上述框架进行数值实验发现：此框架能够较快地求解低维问题 (例如$m = 10, n = 100$)，但测试高维问题 (例如$m = 100, n = 200$) 时会出现求解速度下降，甚至发生无法求解的情况. 与上述框架相比，本文算法虽然理论分析复杂，但能够快速地求解测试问题.

5 结 论

本文考虑了一类二次规划逆问题，首次提出一个交替方向方法求解这类问题. 在两个方向变量极小化问题的求解中，关于矩阵的问题的求解可以得到显示的表达式. 而另外一方向变量的求解被证明本质是一非负约束的严格凸二次规划的求解，或等价的线性互补问题的求解，不可能有显示的表达式. 本文提出两种迭代方法来求解此二次规划或线性互补问题，一个是基于不动点原理的简单迭代方法，另一个是半光滑牛顿法，我们还给出了这两种迭代方法的收敛性和收敛速度的分析. 我们应用所提出的方法做了大量的数值实验，数值结果表明，本文提出的交替方向方法是非常有效的. 本文的算法框架简单且易于编程实现，有利于工程技术人员的使用.

参 考 文 献

[1] Tarantola A. *Inverse Problem Theory: Method for Data Fatting and Model Parameter Estimation* [M]. Amsterdam: Elsevier, 1987.

[2] Burton D, Toint Ph L. On an instance of the inverse shortest paths problem [J]. *Mathematical Programming*, 1992, **53**: 45-61.

[3] Burton D, Toint Ph L. On the use of an inverse shortest paths algorithm for recovering linearly correlated costs [J]. *Mathematical Programming*, 1994, **63**: 1-22.

[4] Carr S C, Lovejoy W S. The inverse newsvendor problem: choosing an optimal demand portfolio for capacitated resources [J]. *Management Science*, 2000, **46**7: 912-927.

[5] Dembo R, Merkoulovitch L, Rosen D. Images from a portfolio [R]. Toronto: Algorithmics, 1998.

[6] Ahuja R K, Orlin J B. A faster algorithm for the inverse spanning tree problem [J]. *Journal of Algorithms*, 2000, **34**: 177-193.

[7] Ahuja R K, Orlin J B. Combinatorial algorithms for inverse network flow problems [J]. *Networks*, 2002, **40**: 181-187.

[8] Heuberger C. Inverse combinatorial optimization: a survey on problem, methods and results [J]. *Journal of Combinatorial Optimization*, 2004, **8**: 329-361.

[9] Zhang J, Liu Z. Calculating some inverse linear programming problems [J]. *Journal of Computational and Applied Mathematics*, 1996, **72**: 261-273.

[10] Zhang J, Liu Z. A further study on inverse linear programming problems [J]. *Journal of Computational and Applied Mathematics*, 1999, **106**: 345-359.

[11] Ahuja R K, Orlin J B. Inverse optimization [J]. *Operations Research*, 2001, **49**: 771-783.

[12] Zhang J, Zhang L W. An augmented Lagrangian method for a class of inverse quadratic programming problems [J]. *Applied Mathematics & Optimization*, 2010, **61**: 57-83.

[13] Xiao X, Zhang L W. A smoothing Newton method for a type of IQP problems [J]. *Journal of Computational Mathematics*, 2009, **27**: 787-801.

[14] 肖现涛. 求解半定约束二次规划逆问题的数值方法 [D]. 大连: 大连理工大学，2009.

[15] 张立卫, 单峰. 最优化方法 [M]. 北京: 科学出版社, 2010.

[16] Facchinei F, Pang J S. *Finite-Dimensional Variational Inequalities and Complementarity Problem* [M]. New York: Springer-Verlag, 2003.

[17] Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers [J]. *Foundations and Trends in Machine Learning*, 2010, **3**: 1-122.

[18] Grant M, Boyd S. CVX: Matlab software for disciplined convex programming, version 1.22 [EB/OL]. [2013-11-21]. http://cvxr.com/cvx.

[19] Grant M, Boyd S, Graph implementations for nonsmooth convex programs [M]//*Recent Advances in Learning and Control*, London: Springer-Verlag, 2008, 95-110.

[20] Strum J F. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones [J]. *Optimization Methods and Software*, 1999, **11**: 625-653.

[21] Tutuncu R H, Toh K C, Todd M J. Solving semidefinite-quadratic-linear programs using SDPT3 [J]. *Mathematical Programming*, 2005, **95**: 189-217.