

求解一类二次规划反问题的同伦交替方向法

高峰, 宇振盛

(上海理工大学 理学院, 上海 200093)

摘要: 对一类带不等式约束的二次规划反问题的求解方法进行研究。首先表示出此类二次规划对应的反问题形式, 将该反问题转化为目标函数变量可分离优化问题, 将其中约束写成 KKT 条件的形式之后, 该反问题等同于一个等式约束优化问题。综合以上, 考虑使用交替方向乘子法进行迭代, 在此基础上, 将同伦思想应用于算法每步迭代的子问题中, 以此避免近端算子选取的敏感性, 又可保证算法的收敛速度。针对子问题, 使用逐次超松弛法进行求解, 并获取算法的收敛性。最后, 将该算法与 SDPT3 和 Sedumi 两种方法进行比较, 数值结果表明, 该算法无论在速度上还是效率上都优于以上两种方法。

关键词: 反问题; 二次规划; 同伦交替方向法; 超松弛迭代法

中图分类号: O 221.2

文献标志码: A

A homotopy-based ADMM for a class of inverse quadratic programming problems

GAO Feng, YU Zhensheng

(College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: The solution of the inverse quadratic programming problem with inequality constraint was studied. Firstly, the form of inverse problem corresponding to this kind of quadratic programming was written, and then the inverse problem was transformed into an optimization problem with separable objective functions. After the constraints were written in the form of KKT conditions, the inverse problem was equivalent to an equality-constrained optimization problem. Based on above transformation, the alternating direction method of multipliers (ADMM) was considered for iteration. On this basis, the homotopy idea was applied to the subproblems of each iteration of our algorithm, so as to avoid the sensitivity of the selection of proximal operators and ensure the convergence speed of the algorithm. For the subproblems, the successive over-relaxation method was used to solve them, and the convergence of the algorithm was obtained. Finally, comparing the proposed method with SDPT3 and Sedumi, the results show that our algorithm is superior to the above two methods both in speed and efficiency.

Keywords: inverse problems; quadratic programming; homotopy-based ADMM; successive over-relaxation method

收稿日期: 2021-09-02

基金项目: 教育部行指委教育改革创新项目 (HBKC213014)

第一作者: 高峰 (1997-), 女, 硕士研究生。研究方向: 最优化方法。E-mail: 3177486063@qq.com

通信作者: 宇振盛 (1973-), 男, 教授。研究方向: 最优化方法。E-mail: zhsh-yu@163.com

反问题研究在投资组合优化和生成树逆问题中具有广泛而极高的应用价值，特别是在网络流逆问题中，为解决信息资源配置问题提供了理论依据和支持。针对反问题，已有大量学者在理论上和算法上进行了深入的研究，并取得了丰硕的成果。早在1996年，Zhang等^[1-2]就已着手研究线性规划反问题，Iyengar等^[3]则在2005年讨论了锥规划反问题。2018年，Khan等^[4]研究了拟变分不等式中参数辨识的反问题，并提出了一种抽象的非光滑正则化方法。具体到二次规划反问题上，Zhang等^[5]于2010年发现此类问题的对偶问题是一个SC¹凸问题(即目标函数连续可微并且梯度半光滑)，并且提出使用增广拉格朗日法求解此类问题。针对相同问题，Xiao等^[6]于2009年提出了一种光滑牛顿法进行求解。Lu等^[7]在2019年提出了非凸交替乘子方向法求解一类稀疏的半定逆二次规划问题。李丽丹等^[8]在2021年提出了G-ADMM法对一类二次规划逆问题进行求解，目标函数是矩阵谱范数与向量无穷范数之和的最小化问题。上述的一些算法虽然全局收敛，但在实际迭代过程中，保证算法线性收敛的前提条件太强而难以满足。除此之外，反问题的目标函数中会出现矩阵范数与向量范数之和，这导致其对偶形式不易求出，因此，就有必要考虑直接对原问题进行求解。为更好解决这类问题，本文使用交替方向乘子法(ADMM)，该方法是解决变量可分离问题的有效方法之一。交替方向乘子法最早由Gabay等^[9]在1976年提出，并且Boyd等^[10]在2011年验证了此方法可用来求解大规模的分布式优化问题。同年，文献[11]基于传统交替方向法，在生成迭代步骤的过程中添加了近似二次项，这种方法被称为近端交替方向法(PADM)，但这种近端交替方向法在近端参数的选择上较为敏感。本文提出了一种基于同伦法的ADMM方法，这种方法将同伦思想融入迭代的每一子问题中，既可克服近端参数选取的敏感性这一缺点，又可加快ADMM收敛速度。

1 问题的提出

考虑凸二次规划问题 $QP(G, c, A, b)$ ：

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} \geqslant \mathbf{b} \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{1}$$

式中： $\mathbf{G} \in \mathbb{S}_+^n$ 和 $\mathbf{c} \in \mathbb{R}^n$ 分别为矩阵和向量； $\mathbb{S}_+^n$ 为对称半定矩阵集合；且 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 和 $\mathbf{b} \in \mathbb{R}^m$ 的值已给定。

传统的正向优化过程是从所有可行解中找到一个最优解 $\mathbf{x}$ ，前提是其中参数 $\mathbf{G} \in \mathbb{S}_+^n$ ， $\mathbf{c} \in \mathbb{R}^n$ 的值会精确给定。但是，反过来讲，有这样一类优化问题，只知道参数 $\mathbf{G}$ 和 $\mathbf{c}$ 的估计值，从经验、观察或者实验中可以知道此问题的可行解，这种情况下，需要找到使给定解成为最优解时的参数 $\mathbf{G}$ 和 $\mathbf{c}$ 的值，并且求得的参数值应该尽可能靠近之前各自的估计值。该类问题称为优化反问题。

反问题在股票市场应用居多，常见的投资组合优化问题可以描述如下：

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} - \lambda' \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} \leqslant \mathbf{b} \\ & \mathbf{x} \geqslant 0 \end{aligned} \tag{2}$$

式中： $\lambda' > 0$ 是控制收益和风险的权重参数； $\mathbf{A} \mathbf{x} \leqslant \mathbf{b}$ 代表投资组合的约束条件。

一般来说，在一个股票市场里，投资者需要根据目标预期收益来决定每只股票的权重，这就是最优投资组合问题。

而在现实问题中，每只股票的预期收益以及不同股票之间的协方差会随市场的变化而变化。这里用 $(\mathbf{G}^0, \mathbf{c}^0)$ 代表 $(\mathbf{G}, \mathbf{c})$ 的最新估计值，用 $\mathbf{x}^0$ 代表问题(2)在 $(\mathbf{G}, \mathbf{c}) = (\mathbf{G}^0, \mathbf{c}^0)$ 时的最优解。在这种股市变动情况下，投资者先前的投资组合 $\mathbf{x}^0$ 不再是新股市场的最优组合，因为，此时新市场的期望收益变为 $\mathbf{c}^0$ ，协方差变为 $\mathbf{G}^0$ ，投资者如果继续使用此投资组合将会蒙受经济损失。因此，为了维持经济效益，并且在不改变投资组合的前提下实现原本投资组合 $\mathbf{x}^0$ 的价值，需要找出更适用此种投资组合的投资者，而这样的投资者所持股票的期望收益率 $\mathbf{c}$ 和股票之间协方差 $\mathbf{G}$ 应当尽可能接近第一个投资者股票的期望收益率和协方差的估计值，除此之外，要使得现存投资组合 $\mathbf{x}^0$ 为新投资者所持股票组合最优解。因此，找到合适的 $\mathbf{G}$ 和 $\mathbf{c}$ 值极具实际意义。

问题(2)可写成如下的优化反问题：

$$\begin{aligned} \min \quad & f(\mathbf{G}, \mathbf{c}) = \|\mathbf{G} - \mathbf{G}^0\| + \|\mathbf{c} - \mathbf{c}^0\| \\ \text{s.t.} \quad & (\mathbf{G}, \mathbf{c}) \in \psi(\mathbf{x}^0) \\ & (\mathbf{G}, \mathbf{c}) \in \mathbb{S}_+^n \times \mathbb{R}^n \end{aligned} \tag{3}$$

式中： $\|\cdot\|$ 代表矩阵和向量的范数； $\psi(\mathbf{x}^0)$ 表示 $\mathbf{x}^0$ 为问题(2)的最优解时所有 $(\mathbf{G}, \mathbf{c})$ 的集合。

因此,如果 $(\mathbf{G}, \mathbf{c}) \in \psi(\mathbf{x}^0)$,即 $f(\mathbf{G}^0, \mathbf{c}^0) = 0$,那么,此时 $\mathbf{x}^0$ 也是问题(3)在参数 $(\mathbf{G}, \mathbf{c}) = (\mathbf{G}^0, \mathbf{c}^0)$ 时的最优解,这意味着投资者可以继续使用投资组合 $\mathbf{x}^0$ 。否则,如果 $f(\mathbf{G}^0, \mathbf{c}^0) > 0$,要保证交易成本和回报之间的平衡就要保证估计值 $(\mathbf{G}^0, \mathbf{c}^0)$ 和集合 $\psi(\mathbf{x}^0)$ 之间的偏差尽可能小。总之,研究二次规划反问题具有极大的现实意义。

对于问题(1),为方便表示,令

$$\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m)^T, \quad \mathbf{a}_i \in \mathbb{R}^n, \quad i = 1, 2, \dots, m$$

式中, $\mathbf{S}_+^n$ 定义了一个 $n \times n$ 的半定矩阵锥的空间。本文假设 $m \leq n$ 。

从观测和实验中可以得到参数 $\mathbf{G}$ 和 $\mathbf{c}$ 估计值 $\mathbf{G}^0 \in \mathbf{S}_+^n$ 和 $\mathbf{c}^0 \in \mathbb{R}^n$,给定一个可行点 $\mathbf{x}^0$ 使之满足约束条件 $\mathbf{A}\mathbf{x}^0 \geq \mathbf{b}$ 。本文用 $S(Q)$ 表示 Q 的最优解,所以,本文研究二次规划反问题的目的即为求出 $(\mathbf{G}, \mathbf{c}) \in \mathbf{S}_+^n \times \mathbb{R}^n$ 的值,使之满足 $\mathbf{x}^0 \in S(QP(\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}))$,且 $(\mathbf{G}, \mathbf{c})$ 要尽可能接近 $(\mathbf{G}^0, \mathbf{c}^0)$ 。因此,相应的二次规划反问题 $IQP(\mathbf{G}, \mathbf{c})$ 可写为

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{G} - \mathbf{G}^0\|_2^2 + \frac{1}{2} \|\mathbf{c} - \mathbf{c}^0\|_2^2 \\ \text{s.t.} \quad & \mathbf{x}^0 \in S(QP(\mathbf{G}, \mathbf{c})) \\ & (\mathbf{G}, \mathbf{c}) \in \mathbf{S}_+^n \times \mathbb{R}^n \end{aligned} \quad (4)$$

式中, $\|\cdot\|_2$ 表示矩阵的谱范数和向量的欧几里得范数,即 $\|\mathbf{G}\|_2 = \max\{\sqrt{\lambda} | \lambda \in \lambda(\mathbf{G}^T \mathbf{G})\}$ 。

众所周知,问题 $QP(\mathbf{G}, \mathbf{c})$ 是严格凸二次规划问题,因此,式(4)中的约束,写成KKT条件为^[12]

$$\begin{cases} \mathbf{G}\mathbf{x}^0 + \mathbf{c} - \mathbf{A}^T \mathbf{u} = 0 \\ 0 \leq \mathbf{u} \perp (\mathbf{A}\mathbf{x}^0 - \mathbf{b}) \geq 0 \end{cases}$$

式中, $\mathbf{u} \in \mathbb{R}^m$ 。

不失一般性,假设前 p 个约束在 $\mathbf{x}^0$ 处是有效约束,即 $\mathbf{x}^0$ 的积极约束集等价于

$$I(\mathbf{x}^0) := \{i | \mathbf{a}_i^T \mathbf{x}^0 = b_i, i = 1, 2, \dots, m\} = \{1, 2, \dots, p\}$$

令 $\mathbf{A}_0 := (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p)^T, \mathbf{a}_i \in \mathbb{R}^n (i = 1, 2, \dots, p)$,即 $\mathbf{A}_0$ 表示矩阵 $\mathbf{A}$ 的前 p 行。同时引入向量 $\mathbf{y}$,其由向量 $\mathbf{u}$ 的前 p 个元素构成,所以, $\mathbf{x}^0 \in S(QP(\mathbf{G}, \mathbf{c}))$ 可以等同于

$$\begin{cases} \mathbf{G}\mathbf{x}^0 + \mathbf{c} - \mathbf{A}_0^T \mathbf{y} = 0 \\ \mathbf{y} \geq 0, \mathbf{y} \in \mathbb{R}^p \end{cases}$$

令 $\delta_{\mathbb{R}_+^p}(\cdot)$ 代表在 $\mathbb{R}_+^p$ 上的指标函数,因此,反问

题 $IQP(\mathbf{G}, \mathbf{c})$ 可被重写为如下变量分离形式:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{G} - \mathbf{G}^0\|^2 + \frac{1}{2} \|\mathbf{c} - \mathbf{c}^0\|^2 + \delta_{\mathbb{R}_+^p}(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{G}\mathbf{x}^0 + \mathbf{c} - \mathbf{A}_0^T \mathbf{y} = 0 \\ & (\mathbf{G}, \mathbf{c}, \mathbf{y}) \in \mathbf{S}_+^n \times \mathbb{R}^n \times \mathbb{R}^p \end{aligned} \quad (5)$$

其拉格朗日函数为

$$\begin{aligned} L_\lambda(\mathbf{G}, \mathbf{c}, \mathbf{y}, \lambda) = & \frac{1}{2} \|\mathbf{G} - \mathbf{G}^0\|^2 + \frac{1}{2} \|\mathbf{c} - \mathbf{c}^0\|^2 + \delta_{\mathbb{R}_+^p}(\mathbf{y}) - \\ & \lambda^T (\mathbf{c} - \mathbf{A}_0^T \mathbf{y} + \mathbf{G}\mathbf{x}^0) + \frac{\beta}{2} \|\mathbf{c} - \mathbf{A}_0^T \mathbf{y} + \mathbf{G}\mathbf{x}^0\|^2 \end{aligned}$$

交替方向法是求解目标函数可分离优化问题的重要工具。本文为求解一类二次规划反问题,提出了一种基于同伦法的交替方向乘子法,该方法将同伦思想与经典的ADMM方法相结合。同伦方法最早于1930年被提出,是解决非线性问题的有力工具,在凸优化和非凸优化方面都做出了巨大贡献。2017年,Yang等^[13]提出了一种基于同伦的交替方向乘子方法来求解线性约束可分离凸极小化问题。本文将同伦方法应用到迭代的每一个子问题上,以此避免敏感近端参数的选取,同时加快了传统ADMM的速度。

2 同伦ADMM算法

现介绍外迭代使用的近端ADMM算法以及内迭代使用的逐次超松弛法。

2.1 外迭代

已知 $(\mathbf{G}^k, \mathbf{c}^k, \mathbf{y}^k, \lambda^k)$ 的前提下,由ADMM产生 $(\mathbf{G}^{k+1}, \mathbf{c}^{k+1}, \mathbf{y}^{k+1}, \lambda^{k+1})$ 的迭代结构如下:

(S1) 已知 $(\mathbf{c}^k, \mathbf{y}^k, \lambda^k)$,则有

$$\mathbf{G}^{k+1} = \arg \min_{\mathbf{G} \in \mathbf{S}_+^n} \{L_\beta(\mathbf{G}, \mathbf{c}^k, \mathbf{y}^k, \lambda^k)\}$$

(S2) 已知 $\mathbf{G}^{k+1}, \mathbf{y}^k$ 和 λ^k ,则有

$$\mathbf{c}^{k+1} = \arg \min_{\mathbf{c} \in \mathbb{R}^n} \{L_\beta(\mathbf{G}^{k+1}, \mathbf{c}, \mathbf{y}^k, \lambda^k)\}$$

(S3) 已知 $\mathbf{G}^{k+1}, \mathbf{c}^{k+1}$ 和 λ^k ,则有

$$\mathbf{y}^{k+1} = \arg \min_{\mathbf{y} \in \mathbb{R}^p} \{L_\beta(\mathbf{G}^{k+1}, \mathbf{c}^{k+1}, \mathbf{y}, \lambda^k)\}$$

(S4) 拉格朗日乘子 λ 的更新为

$$\lambda^{k+1} = \lambda^k - \beta (\mathbf{c}^{k+1} - \mathbf{A}_0^T \mathbf{y}^{k+1} + \mathbf{G}^{k+1} \mathbf{x}^0)$$

上述步骤等同于

$$\begin{cases} \mathbf{G}^{k+1} = \arg \min_{\mathbf{G}} \left\{ \frac{1}{2} \|\mathbf{G} - \mathbf{G}^0\|^2 + \frac{\beta}{2} \|\mathbf{c}^k - \mathbf{A}_0^T \mathbf{y}^k + \mathbf{G} \mathbf{x}^0 - \frac{1}{\beta} \lambda^k\|^2 \right\} \\ \mathbf{c}^{k+1} = \arg \min_{\mathbf{c}} \left\{ \frac{1}{2} \|\mathbf{c} - \mathbf{c}^0\|^2 + \frac{\beta}{2} \|\mathbf{c} - \mathbf{A}_0^T \mathbf{y}^k + \mathbf{G}^{k+1} \mathbf{x}^0 - \frac{1}{\beta} \lambda^k\|^2 \right\} \\ \mathbf{y}^{k+1} = \arg \min_{\mathbf{y}} \left\{ \frac{\beta}{2} \|\mathbf{c}^{k+1} - \mathbf{A}_0^T \mathbf{y} + \mathbf{G}^{k+1} \mathbf{x}^0 - \frac{1}{\beta} \lambda^k\|^2 \right\} \\ \lambda^{k+1} = \lambda^k - \beta (\mathbf{c}^{k+1} - \mathbf{A}_0^T \mathbf{y}^{k+1} + \mathbf{G}^{k+1} \mathbf{x}^0) \end{cases} \quad (6)$$

为确保迭代过程收敛性,在上述子问题中分别加入近端项 $\frac{r_k}{2} \|\mathbf{G} - \mathbf{G}^k\|^2$, $\frac{s_k}{2} \|\mathbf{c} - \mathbf{c}^k\|^2$ 和 $\frac{t_k}{2} \|\mathbf{y} - \mathbf{y}^k\|^2$,其中, r_k , s_k 和 t_k 为近端算子。因此,上述迭代过程可重新描述为

$$\begin{cases} \mathbf{G}^{k+1} = \arg \min_{\mathbf{G}} \left\{ \frac{1}{2} \|\mathbf{G} - \mathbf{G}^0\|^2 + \frac{\beta}{2} \|\mathbf{c}^k - \mathbf{A}_0^T \mathbf{y}^k + \mathbf{G} \mathbf{x}^0 - \frac{1}{\beta} \lambda^k\|^2 + \frac{r_k}{2} \|\mathbf{G} - \mathbf{G}^k\|^2 \right\} \\ \mathbf{c}^{k+1} = \arg \min_{\mathbf{c}} \left\{ \frac{1}{2} \|\mathbf{c} - \mathbf{c}^0\|^2 + \frac{\beta}{2} \|\mathbf{c} - \mathbf{A}_0^T \mathbf{y}^k + \mathbf{G}^{k+1} \mathbf{x}^0 - \frac{1}{\beta} \lambda^k\|^2 + \frac{s_k}{2} \|\mathbf{c} - \mathbf{c}^k\|^2 \right\} \\ \mathbf{y}^{k+1} = \arg \min_{\mathbf{y}} \left\{ \frac{\beta}{2} \|\mathbf{c}^{k+1} - \mathbf{A}_0^T \mathbf{y} + \mathbf{G}^{k+1} \mathbf{x}^0 - \frac{1}{\beta} \lambda^k\|^2 + \frac{t_k}{2} \|\mathbf{y} - \mathbf{y}^k\|^2 \right\} \\ \lambda^{k+1} = \lambda^k - \beta (\mathbf{c}^{k+1} - \mathbf{A}_0^T \mathbf{y}^{k+1} + \mathbf{G}^{k+1} \mathbf{x}^0) \end{cases} \quad (7)$$

实际上,迭代式(7)等同于求解下述方程:

$$\begin{cases} (\mathbf{G} - \mathbf{G}^0) + \beta \left(\mathbf{c}^k - \mathbf{A}_0^T \mathbf{y}^k + \mathbf{G} \mathbf{x}^0 - \frac{\lambda^k}{\beta} \right) (\mathbf{x}^0)^T + \\ r_k (\mathbf{G} - \mathbf{G}^k) = 0 \\ (\mathbf{c} - \mathbf{c}^0) + \beta \left(\mathbf{c} - \mathbf{A}_0^T \mathbf{y}^k + \mathbf{G}^{k+1} \mathbf{x}^0 - \frac{\lambda^k}{\beta} \right) + \\ s_k (\mathbf{c} - \mathbf{c}^k) = 0 \\ \beta \mathbf{A}_0 \left(-\mathbf{c}^{k+1} + \mathbf{A}_0^T \mathbf{y} - \mathbf{G}^{k+1} \mathbf{x}^0 + \frac{\lambda^k}{\beta} \right) + t_k (\mathbf{y} - \mathbf{y}^k) = 0 \\ \lambda^{k+1} = \lambda^k - \beta (\mathbf{c}^{k+1} - \mathbf{A}_0^T \mathbf{y}^{k+1} + \mathbf{G}^{k+1} \mathbf{x}^0) \end{cases} \quad (8)$$

由问题(5)的一阶最优条件可知,对于任意

$$(\mathbf{G}^*, \mathbf{c}^*, \mathbf{y}^*) \in \mathbf{S}_n^+ \times \mathbb{R}^n \times \mathbb{R}^p, \text{ 存在 } \lambda^* \in \mathbb{R}^n, \text{ 使得 } \begin{pmatrix} (\mathbf{G}^* - \mathbf{G}^0) - \lambda^* (\mathbf{x}^0)^T \\ (\mathbf{c}^* - \mathbf{c}^0) - \lambda^* \\ \mathbf{A}_0 \lambda^* \\ \mathbf{c}^* - \mathbf{A}_0^T \mathbf{y}^* + \mathbf{G}^* \mathbf{x}^0 \end{pmatrix} = 0 \quad (9)$$

为方便表示,引入符号

$$\mathbf{W} := \mathbf{S}_n^+ \times \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^n, \quad \omega = (\mathbf{G}, \mathbf{c}, \mathbf{y}, \lambda) \in \mathbf{W},$$

$$\mathbf{W}^* = \{\omega^* \in \mathbf{W} : (\mathbf{G}^*, \mathbf{c}^*, \mathbf{y}^*, \lambda^*)\}$$

则有

$$F(\omega) = \begin{pmatrix} (\mathbf{G} - \mathbf{G}^0) - \lambda (\mathbf{x}^0)^T \\ (\mathbf{c} - \mathbf{c}^0) - \lambda \\ \mathbf{A}_0 \lambda \\ \mathbf{c} - \mathbf{A}_0^T \mathbf{y} + \mathbf{G} \mathbf{x}^0 \end{pmatrix} \quad (10)$$

综上所述,求解 $\omega^* \in \mathbf{W}^*$ 本质上就是求解非线性方程组 $F(\omega) = 0$ 。

定义一个广义同伦映射为^[13]

$$T(\omega, \alpha) = \alpha F(\omega) + (1 - \alpha)(\mathbf{H}\omega - \mathbf{a})$$

同伦方法是通过选择合适的矩阵 $\mathbf{H}$ 和向量 $\mathbf{a}$,再将 α_k 的值由0迭代1,得到 $T(\omega, \alpha_k) = 0$ 的一系列解的过程,特别是,最终 $\alpha_k \rightarrow 1$ 时即可求得 $F(\omega) = 0$ 的近似解。总的来说,同伦方法的目的是将求解一个给定的较难问题转化为求解简单问题。原问

题 $F(\omega) = 0$ 求解起来较为困难,但求解问题 $(\mathbf{H}\omega - \mathbf{a}) = 0$ 则相对容易。

对于给定同伦参数 $\alpha_k \in (0, 1)$,令 $r_k, s_k, t_k = \frac{1 - \alpha_k}{\alpha_k}$,罚参数 $\beta_k = \frac{\beta}{\alpha_k}$,因此,关于 $\mathbf{G}$ 的子问题等价于

$$\begin{aligned} & (\mathbf{G} - \mathbf{G}^0) + \beta_k \left(\mathbf{c}^k - \mathbf{A}_0^T \mathbf{y}^k + \mathbf{G} \mathbf{x}^0 - \frac{\lambda^k}{\beta_k} \right) (\mathbf{x}^0)^T + r_k (\mathbf{G} - \mathbf{G}^k) = 0 \\ & \Rightarrow (\mathbf{G} - \mathbf{G}^0) + \beta_k \left(\mathbf{c}^k - \mathbf{A}_0^T \mathbf{y}^k + \mathbf{G} \mathbf{x}^0 - \frac{\lambda^k}{\beta_k} \right) (\mathbf{x}^0)^T + \\ & \frac{1 - \alpha_k}{\alpha_k} (\mathbf{G} - \mathbf{G}^k) = 0 \end{aligned}$$

$$\begin{aligned} & \Rightarrow \alpha_k \left[(\mathbf{G} - \mathbf{G}^0) - \lambda^k (\mathbf{x}^0)^T \right] + \\ & (1 - \alpha_k) \left[\mathbf{G} \left(\frac{\alpha_k \beta_k}{1 - \alpha_k} \mathbf{x}^0 (\mathbf{x}^0)^T + \mathbf{I} \right) - \mathbf{G}^k + \right. \\ & \left. \frac{\alpha_k \beta_k}{1 - \alpha_k} (\mathbf{c}^k - \mathbf{A}_0^T \mathbf{y}^k) (\mathbf{x}^0)^T \right] = 0 \end{aligned}$$

$$\begin{aligned} & \text{同理,关于}\mathbf{c}\text{的子问题等价于} \\ & (\mathbf{c} - \mathbf{c}^0) + \beta_k \left(\mathbf{c} - \mathbf{A}_0^T \mathbf{y}^k + \mathbf{G}^{k+1} \mathbf{x}^0 - \frac{\lambda^k}{\beta_k} \right) + s_k (\mathbf{c} - \mathbf{c}^k) = 0 \end{aligned}$$

$$\begin{aligned} & \Rightarrow (\mathbf{c} - \mathbf{c}^0) + \beta_k \left(\mathbf{c} - \mathbf{A}_0^T \mathbf{y}^k + \mathbf{G}^{k+1} \mathbf{x}^0 - \frac{\lambda^k}{\beta_k} \right) + \frac{1 - \alpha_k}{\alpha_k} (\mathbf{c} - \mathbf{c}^k) = 0 \\ & \Rightarrow \alpha_k (\mathbf{c} - \mathbf{c}^0) - \alpha_k \lambda^k + (1 - \alpha_k) \cdot \end{aligned}$$

$$\begin{aligned} & \left[\left(\frac{\alpha_k \beta_k}{1 - \alpha_k} \mathbf{I} + \mathbf{I} \right) \mathbf{c} - \mathbf{c}^k + \frac{\alpha_k \beta_k}{1 - \alpha_k} (-\mathbf{A}_0^T \mathbf{y}^k + \mathbf{G}^{k+1} \mathbf{x}^0) \right] = 0 \\ & \text{关于}\mathbf{y}\text{的子问题等价于} \end{aligned}$$

$$\begin{aligned} & \beta_k \mathbf{A}_0 \left(-\mathbf{c}^{k+1} + \mathbf{A}_0^T \mathbf{y} - \mathbf{G}^{k+1} \mathbf{x}^0 + \frac{\lambda^k}{\beta_k} \right) + t_k (\mathbf{y} - \mathbf{y}^k) = 0 \\ & \Rightarrow \beta_k \mathbf{A}_0 \left(-\mathbf{c}^{k+1} + \mathbf{A}_0^T \mathbf{y} - \mathbf{G}^{k+1} \mathbf{x}^0 + \frac{\lambda^k}{\beta_k} \right) + \frac{1 - \alpha_k}{\alpha_k} (\mathbf{y} - \mathbf{y}^k) = 0 \\ & \Rightarrow \alpha_k \mathbf{A}_0 \lambda^k + (1 - \alpha_k) \cdot \\ & \left[\left(\frac{\alpha_k \beta_k}{1 - \alpha_k} \mathbf{A}_0 \mathbf{A}_0^T + \mathbf{I} \right) \mathbf{y} - \mathbf{y}^k + \frac{\alpha_k \beta_k}{1 - \alpha_k} \mathbf{A}_0 (-\mathbf{c}^{k+1} - \mathbf{G}^{k+1} \mathbf{x}^0) \right] = 0 \end{aligned}$$

现将基于同伦的近端 ADMM(HADMM)应用于问题 (6), 下面介绍算法。

算法 1

给定 $\mathbf{z}^k = (\mathbf{G}^k, \mathbf{c}^k, \mathbf{y}^k, \lambda^k)$, 令 $\mu \in (0, 1)$, $\varepsilon > 0$, 同伦因子 $\alpha_k \in (0, 1)$, 惩罚参数 $\beta_k = \frac{\beta}{\alpha_k}$ ($\beta > 0$, 为一常数)。则由 HADMM 产生的新的迭代点 $\mathbf{z}^{k+1} = (\mathbf{G}^{k+1}, \mathbf{c}^{k+1}, \mathbf{y}^{k+1}, \lambda^{k+1})$ 的过程由下面的步骤组成。

步骤 1 给定初始矩阵 $\mathbf{G} \in \mathbf{S}_+^n$, 初始向量 $\mathbf{c} \in \mathbb{R}^n$, 估计值 $\mathbf{G}^0 = \text{rand}(n) + 300\text{eye}(n)$, $\mathbf{c}^0 = \text{rand}(n, 1)$, 矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$, 向量 $\mathbf{b} \in \mathbb{R}^m$, 可行点 $\mathbf{x}^0 \in \mathbb{R}^n$, 同伦因子初始值 $\alpha_0 \in (0, 1)$, $\beta > 0$, 令罚参数初始值 $\beta_0 = \frac{\beta}{\alpha_0}$, $\mu \in (0, 1)$, $\varepsilon > 0$;

步骤 2 计算出 p 和 $\mathbf{A}_0$ 的值;

步骤 3 对于给定的 $\mathbf{z}^k = (\mathbf{G}^k, \mathbf{c}^k, \mathbf{y}^k, \lambda^k)$, 由 HADMM 产生的新的迭代点 $\mathbf{z}^{k+1} = (\mathbf{G}^{k+1}, \mathbf{c}^{k+1}, \mathbf{y}^{k+1}, \lambda^{k+1})$ 。

$\mathbf{G}^{k+1}$ 为方程

$$\alpha_k \left[(\mathbf{G} - \mathbf{G}^0) - \lambda^k (\mathbf{x}^0)^T \right] + (1 - \alpha_k) (\mathbf{G}\mathbf{T} - \boldsymbol{\xi}) = \mathbf{0} \quad (11)$$

的解, 其中, $\boldsymbol{\xi} = \mathbf{G}^k - \tau_k (\mathbf{c}^k - \mathbf{A}_0^T \mathbf{y}^k) (\mathbf{x}^0)^T$, $\mathbf{T} = \tau_k \mathbf{x}^0 (\mathbf{x}^0)^T +$

$\mathbf{I}$, $\tau_k = \frac{\alpha_k \beta_k}{1 - \alpha_k}$;

$\mathbf{c}^{k+1}$ 为方程

$$\alpha_k \left[(\mathbf{c} - \mathbf{c}^0) - \lambda^k \right] + (1 - \alpha_k) (\mathbf{P}\mathbf{c} - \mathbf{f}) = \mathbf{0} \quad (12)$$

的解, 其中, $\phi = \mathbf{c}^k - \tau_k (-\mathbf{A}_0^T \mathbf{y}^k + \mathbf{G}^{k+1} \mathbf{x}^0)$, $\mathbf{P} = \tau_k \mathbf{I} + \mathbf{I}$;

$\mathbf{y}^{k+1}$ 为方程

$$\alpha_k \mathbf{A}_0 \lambda^k + (1 - \alpha_k) (\mathbf{Q}\mathbf{y} - \boldsymbol{\varphi}) = \mathbf{0} \quad (13)$$

的解, 其中, $\boldsymbol{\varphi} = \mathbf{y}^k - \tau_k \mathbf{A}_0 (-\mathbf{c}^{k+1} - \mathbf{G}^{k+1} \mathbf{x}^0)$, $\mathbf{Q} = \tau_k \mathbf{A}_0 \mathbf{A}_0^T + \mathbf{I}$;

λ^{k+1} 为方程

$$\lambda^{k+1} = \lambda^k - \frac{\tau_k (1 - \alpha_k)}{\alpha_k} (\mathbf{c}^{k+1} - \mathbf{A}_0^T \mathbf{y}^{k+1} + \mathbf{G}^{k+1} \mathbf{x}^0) \quad (14)$$

的解;

步骤 4 若有 $\max \{ \|\mathbf{G}^{k+1} - \mathbf{G}^k\|, \|\mathbf{c}^{k+1} - \mathbf{c}^k\|, \|\mathbf{y}^{k+1} - \mathbf{y}^k\| \} \leq \varepsilon$, 则停止迭代; 否则, 令 $\alpha_{k+1} := (1 - \alpha_k) \mu + \alpha_k$,

$\beta_{k+1} = \frac{\beta}{\alpha_{k+1}}$, $k := k + 1$, 返回步骤 3。

2.2 内迭代

现引入逐次超松弛法对上述外迭代过程中的子问题进行内迭代求解。

逐次超松弛法是 Gauss-Seidel 法的一种演变方

法。它最初是 Frankel 在 1950 年为了在计算机上求解线性方程而提出的。逐次超松弛迭代法在高斯-赛德尔迭代基础上加入松弛因子以加快迭代收敛速度。

考虑 Gauss-Seidel 迭代法

$$\begin{aligned} \mathbf{x}_i^{k+1} &= -\frac{1}{a_{ii}} \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k+1)} - \frac{1}{a_{ii}} \sum_{j=i+1}^n a_{ij} \mathbf{x}_j^{(k)} + \frac{1}{a_{ii}} \mathbf{b}_i = \\ &= \frac{1}{a_{ii}} \mathbf{b}_i - \frac{1}{a_{ii}} \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k+1)} - \frac{1}{a_{ii}} \sum_{j=i+1}^n a_{ij} \mathbf{x}_j^{(k)} = \\ &= \frac{1}{a_{ii}} \mathbf{b}_i - \frac{1}{a_{ii}} \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k+1)} - \frac{1}{a_{ii}} \sum_{j=i}^n a_{ij} \mathbf{x}_j^{(k)} + \mathbf{x}_i^{(k)} = \\ &= \mathbf{x}_i^{(k)} + \frac{1}{a_{ii}} \left(\mathbf{b}_i - \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k+1)} - \sum_{j=i}^n a_{ij} \mathbf{x}_j^{(k)} \right) \end{aligned}$$

令

$$\begin{aligned} \mathbf{r}_i^{(k)} &= \mathbf{x}_i^{(k+1)} - \mathbf{x}_i^{(k)} = \\ &= \frac{1}{a_{ii}} \left(\mathbf{b}_i - \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k+1)} - \sum_{j=i}^n a_{ij} \mathbf{x}_j^{(k)} \right) \end{aligned}$$

其中, $\mathbf{r}_i^{(k)}$ 是 $\mathbf{x}_i$ 在第 $k+1$ 次迭代时的改变量, 所以, 有

$$\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} + \mathbf{r}_i^{(k)}$$

在该变量 $\mathbf{r}_i^{(k)}$ 前面添加因子 ω ($0 < \omega < 2$), 则有

$$\begin{aligned} \mathbf{x}_i^{(k+1)} &= \mathbf{x}_i^{(k)} + \omega \mathbf{r}_i^{(k)} = \\ &= \mathbf{x}_i^{(k)} + \frac{\omega}{a_{ii}} \left(\mathbf{b}_i - \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k+1)} - \sum_{j=i}^n a_{ij} \mathbf{x}_j^{(k)} \right) \end{aligned}$$

合并 $\mathbf{x}_i^{(k)}$ 可得

$$\begin{aligned} \mathbf{x}_i^{(k+1)} &= (1 - \omega) \mathbf{x}_i^{(k)} + \frac{\omega}{a_{ii}} \left(\mathbf{b}_i - \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} \mathbf{x}_j^{(k)} \right), \\ i &= 1, 2, \dots, n \quad k = 0, 1, 2, \dots \end{aligned}$$

通过算法 1 可得

$$\begin{aligned} \alpha_k \left[(\mathbf{c}^{k+1} - \mathbf{c}^0) - \lambda^k \right] + (1 - \alpha_k) \left[(\tau_k \mathbf{I} + \mathbf{I}) \mathbf{c}^{k+1} - \right. \\ \left. \mathbf{c}^k + \tau_k (-\mathbf{A}_0^T \mathbf{y}^k + \mathbf{G}^{k+1} \mathbf{x}^0) \right] = \mathbf{0} \end{aligned} \quad (15)$$

和

$$\begin{aligned} \alpha_k \mathbf{A}_0 \lambda^k + (1 - \alpha_k) \left[(\tau_k \mathbf{A}_0 \mathbf{A}_0^T + \mathbf{I}) \mathbf{y}^{k+1} - \mathbf{y}^k + \right. \\ \left. \tau_k \mathbf{A}_0 (-\mathbf{c}^{k+1} - \mathbf{G}^{k+1} \mathbf{x}^0) \right] = \mathbf{0} \end{aligned} \quad (16)$$

将式 (14) 中 λ^{k+1} 的更新公式分别与式 (15) 和式 (16) 合并, 可得

$$\alpha_k \left[(\mathbf{c}^{k+1} - \mathbf{c}^0) - \lambda^{k+1} \right] + (1 - \alpha_k) (\mathbf{c}^{k+1} - \mathbf{c}^k) + \tau_k (1 - \alpha_k) \mathbf{A}_0^T (\mathbf{y}^{k+1} - \mathbf{y}^k) = 0$$

和

$$\alpha_k \mathbf{A}_0 \lambda^{k+1} (1 - \alpha_k) (\mathbf{I} + \tau_k \mathbf{A}_0 \mathbf{A}_0^T) (\mathbf{y}^{k+1} - \mathbf{y}^k) - \tau_k (1 - \alpha_k) \mathbf{A}_0 \mathbf{A}_0^T (\mathbf{y}^{k+1} - \mathbf{y}^k) = 0$$

并且，式(14)可重写为

$$\alpha_k (\mathbf{c}^{k+1} - \mathbf{A}_0^T \mathbf{y}^{k+1} + \mathbf{G}^{k+1} \mathbf{x}^0) + \frac{\alpha_k^2}{\tau_k (1 - \alpha_k)} (\lambda - \lambda^k) = 0$$

为简单起见，在收敛分析中引入矩阵

$$\mathbf{P}_k = \begin{pmatrix} (1 - \alpha_k) \mathbf{I} & 0 & 0 \\ 0 & (1 - \alpha_k) (\mathbf{I} + \tau_k \mathbf{A}_0 \mathbf{A}_0^T) & 0 \\ 0 & 0 & \frac{\alpha_k^2}{\tau_k (1 - \alpha_k)} \mathbf{I} \end{pmatrix}$$
$$\mathbf{Q}_k = \begin{pmatrix} (1 - \alpha_k) \mathbf{I} & 0 & 0 \\ 0 & (1 - \alpha_k) (\mathbf{I} + \tau_k \mathbf{A}_0 \mathbf{A}_0^T) & 0 \\ 0 & -\alpha_k \mathbf{A}_0^T & \frac{\alpha_k^2}{\tau_k (1 - \alpha_k)} \mathbf{I} \end{pmatrix}$$
$$\mathbf{R}_k = \begin{pmatrix} \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & 0 \\ 0 & -\frac{\tau_k (1 - \alpha_k)}{\alpha_k} \mathbf{A}_0^T & \mathbf{I} \end{pmatrix}$$

其中， $\mathbf{Q}_k = \mathbf{P}_k \mathbf{M}_k$ 。令 $\mathbf{U} := \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^n$ ， $\mathbf{u} = (\mathbf{c}, \mathbf{y}, \lambda) \in \mathbf{U}$ ，已知 $\mathbf{u}^k$ ，用 $\mathbf{u}^{k+1}$ 表示由本文算法生成的下一步迭代点。如此便有

$$\alpha_k F(\mathbf{u}^{k+1}) + \mathbf{P}_k (\mathbf{u}^{k+1} - \mathbf{u}^k) - \begin{pmatrix} -\tau_k (1 - \alpha_k) \mathbf{A}_0^T (\mathbf{y}^{k+1} - \mathbf{y}^k) \\ \tau_k (1 - \alpha_k) \mathbf{A}_0 \mathbf{A}_0^T (\mathbf{y}^{k+1} - \mathbf{y}^k) \\ 0 \end{pmatrix} = 0$$

定理 1 由算法 1 生成的序列 $\{\mathbf{z}^k\}$ 是收敛的，并且序列极限满足一阶最优性条件式(9)。

定理 1 的证明可参考文献 [13]，其中也证明了最坏情况下的收敛速度为 $O(1/k)$ 。

3 数值实验

现对本文提出的算法进行数值实验，并与国际知名的优化软件 CVX 的子程序 SDPT3 和 Sedumi 求解式(5)的数值结果进行比较。

实验中设置同伦因子初始值， μ 及 β 为
 $\alpha_0 = 0.1, \mu = 0.2, \beta = 0.3$

取 $\mathbf{A}$ ， $\mathbf{x}^0$ 和 $\mathbf{b}$ 的值为

$$\mathbf{x}^0 = \text{ones}(n, 1)$$
$$\mathbf{b} = \text{ones}(m, 1)$$

$$\mathbf{A} = [\text{eye}(400, n); \text{zeros}(m - 400, n)]$$

并计算出 p 和 $\mathbf{A}_0$ 的值。

参数 $(\mathbf{G}, \mathbf{c}, \mathbf{G}^0, \mathbf{c}^0)$ 设置为

$$\mathbf{G} = \text{eye}(n), \mathbf{c} = 2\text{ones}(n, 1)$$

$$\mathbf{G}^0 = \text{rand}(n) + 300\text{eye}(n), \mathbf{c}^0 = \text{rand}(n, 1)$$

算法的数值结果如表 1 和表 2 所示。

表 1 HADMM 数值结果
Tab.1 Numerical results of HADMM

<i>m</i>	<i>n</i>	<i>p</i>	<i>e</i>	<i>t/s</i>	<i>r</i>
100	300	60	56	4.06	1.29E-05
100	400	60	56	7.93	9.80E-06
100	500	60	71	14.09	9.99E-06
100	600	60	60	18.95	9.99E-06
100	700	60	66	27.79	1.09E-05
100	800	60	60	37.35	1.09E-05
100	900	60	82	59.99	1.09E-05
100	1000	60	73	81.47	1.09E-05
500	500	200	84	16.67	1.00E-06
500	600	200	103	27.79	1.09E-06
500	700	200	80	32.37	1.09E-06
500	800	200	113	56.41	9.99E-06
500	900	200	68	55.2	9.99E-06
500	1000	200	73	83.13	8.80E-06
1000	1000	400	50	88.33	9.80E-05
1000	1500	400	48	261.7	9.80E-05
1000	2000	400	50	645.65	9.80E-05
2000	2000	500	49	631.68	9.80E-05

表 1 中， m ， n 分别表示矩阵或向量的行和列。此处将终止条件设置为 $\varepsilon = 10^{-4}$ 。 e ， t 和 r 分别代表迭代次数，CPU时间和精度。表 2 中，字母 F 代表计算失败，即迭代次数超过 300 次。下标 $i(i = 1, 2, 3)$ 分别代表 HADMM，SDPT3 和 Sedumi 这 3 个算法。

从表 1 和表 2 中可以看出，在解决相同维度的问题时，本文的算法无论是在精度还是 CPU 时间上，都要优于 SDPT3 和 Sedumi。除此之外，本文算法能够高效、快速地求解更高维的二次规划反问题。

表 2 数值比较

Tab.2 Numerical Comparison

维数			迭代次数			时间			精度		
m	n	p	e_1	e_2	e_3	t_1/s	t_2/s	t_3/s	r_1	r_2	r_3
10	20	5	52	10	9	0.04	0.45	0.3	4.78E-07	1.49E-08	4.41E-10
10	30	5	55	10	9	0.07	0.48	1.1	4.73E-07	1.49E-08	7.98E-10
10	40	5	56	10	10	0.12	1.2	5.8	4.80E-07	1.49E-08	6.73E-11
10	50	5	61	10	10	0.2	2.95	45.1	8.99E-07	1.49E-08	7.97E-11
50	60	30	58	10	10	0.25	6.54	282	6.93E-07	1.49E-08	1.00E-10
50	70	30	107	10	10	0.42	13.86	F	1.99E-06	1.49E-08	1.00E-10
50	80	30	80	10	F	0.46	23.38	F	1.67E-06	1.49E-08	F
50	90	30	118	10	F	0.63	42.38	F	2.18E-06	1.49E-08	F
50	100	30	77	10	F	0.59	68.3	F	5.95E-06	1.49E-08	F
100	100	30	77	10	F	0.6	70.14	F	5.35E-06	1.49E-08	F
100	200	60	56	F	F	1.7	F	F	4.99E-06	F	F

4 结 论

采用一种基于同伦的 ADMM 方法求解一类二次规划反问题。将该问题转化为目标函数可分离的线性约束问题后, 将同伦思想应用于每一次迭代的子问题中, 不仅可以实现对传统 ADMM 方法的加速, 而且避免了选取近端参数时的敏感性。最后给出了算法的数值实验结果。与 SDPT3 和 Sedumi 的数值结果相比, 本文的算法无论是在 CPU 时间还是残差方面都要优于以上两种方法。最重要的是本文的算法能够更高效求解更高维的问题。

参考文献:

[1] ZHANG J Z, LIU Z H. Calculating some inverse linear programming problems[J]. *Journal of Computational and Applied Mathematics*, 1996, 72(2): 261–273.

[2] ZHANG J Z, LIU Z H. A further study on inverse linear programming problems[J]. *Journal of Computational and Applied Mathematics*, 1999, 106(2): 345–359.

[3] IYENGAR G, KANG W M. Inverse conic programming with applications[J]. *Operations Research Letters*, 2005, 33(3): 319–330.

[4] KHAN A A, MOTREANU D. Inverse problems for quasi-variational inequalities[J]. *Journal of Global Optimization*, 2018, 70(2): 401–411.

[5] ZHANG J Z, ZHANG L W. An augmented Lagrangian method for a class of inverse quadratic programming problems[J]. *Applied Mathematics and Optimization*, 2010, 61(1): 57–83.

[6] XIAO X T, ZHANG L W. Solving a class of inverse QP problems by a smoothing Newton method[J]. *Journal of Computational Mathematics*, 2009, 27(6): 787–801.

[7] LU Y, HUANG M, ZHANG Y, et al. A nonconvex ADMM for a class of sparse inverse semidefinite quadratic programming problems[J]. *Optimization*, 2019, 68(6): 1075–1105.

[8] 李丽丹, 张立卫, 张宏伟. 一类不可微二次规划逆问题 [J]. *计算数学*, 2021, 43(2): 227–240.

[9] GABAY D, MERCIER B. A dual algorithm for the solution of nonlinear variational problems via finite element approximation[J]. *Computers & Mathematics with Applications*, 1976, 2(1): 17–40.

[10] 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*, 2011, 3(1): 1–122.

[11] XU M H, WU T. A class of linearized proximal alternating direction methods[J]. *Journal of Optimization Theory and Applications*, 2011, 151(2): 321–337.

[12] 李丽丹, 张宏伟, 张立卫. Gauss 回代交替方向法求解一类二次规划逆问题 [J]. *应用数学进展*, 2020, 9(1): 60–71.

[13] YANG J, DAI Y Q, PENG Z, et al. A homotopy alternating direction method of multipliers for linearly constrained separable convex optimization[J]. *Journal of the Operations Research Society of China*, 2017, 5(2): 271–290.