利用统一框架设计凸优化的分裂收缩算法*

何炳生
(南京大学数学系, 南京 210093)

USING A UNIFIED FRAMEWORK TO DESIGN THE SPLITTING AND CONTRACTION METHODS FOR CONVEX OPTIMIZATION

He Bingsheng
(Department of Mathematics, Nanjing University, Nanjing 210093)

Abstract　In order to solve the linearly constrained convex optimization problems, by using the concepts of variational inequality (VI) and proximal point algorithm (PPA), we have gradually summed up and developed a simple and powerful algorithmic unified framework. By verifying the conditions in this framework, the convergence proofs of many existing algorithms becomes very simple. Based on this framework, we can construct the appropriate algorithms for different problems. Under the guidance of the framework, this paper presents some algorithms which widely expand the classical algorithms of the ALM and/or ADMM-like classes.

Key words　convex optimization, variational inequality, augmented Lagrangian method, alternating direction methods of multipliers, unified framework, splitting and contraction methods.
AMS(2000) subject classifications　90C25, 90C30, 90C33
中图法分类号　O221.2, O224

1 引言

应用领域内提出的大量优化问题中, 包含了许多可以归结为(或者松弛成)如下典型的线性约束凸优化问题:

* 国家自然科学基金委NSFC Grant 11871029资助项目.
收稿日期: 2021-05-23.

1. 线性约束的单块凸优化问题

$$\min\{\theta(x)|Ax=b\,(\text{or}\geq b),\,x\in\mathcal{X}\}. \tag{1.1}$$

2. 线性等式约束的可分离成两块的凸优化问题

$$\min\{\theta_1(x)+\theta_2(y)|Ax+By=b,\,x\in\mathcal{X},\,y\in\mathcal{Y}\}. \tag{1.2}$$

3. 线性等式约束的可分离成三块的凸优化问题

$$\min\{\theta_1(x)+\theta_2(y)+\theta_3(z)|Ax+By+Cz=b,\,x\in\mathcal{X},\,y\in\mathcal{Y},\,z\in\mathcal{Z}\}. \tag{1.3}$$

以上的 $\theta_1(x),\theta_2(y),\theta_3(z)$ 都是凸函数, A,B,C 都是矩阵, b 是相应的向量, $\mathcal{X},\mathcal{Y},\mathcal{Z}$ 都是简单闭凸集. 通篇, 我们都假设讨论的问题有解. 我们研究线性约束凸优化问题的算法, 着眼于求它们的拉格朗日函数的鞍点. 建立了约束凸优化问题的拉格朗日函数鞍点和变分不等式解集之间的等价关系, 然后聚焦于寻找变分不等式解点的收缩算法 (迭代点在一定范数意义下到解集距离越来越近的算法). 这些工作可以追溯到 1991 年作者和何旭初先生合作的文章 [17]. 后继的研究中, 我们逐步总结出了一个统一的算法框架 [14–16]. 这个算法框架的收敛性证明相当简单, 也容易用其中的收敛性条件去验证一些方法能否确保收敛, 更重要的是能用来指导设计凸优化问题的求解方法.

增广拉格朗日乘子法 (Augmented Lagrangian Method, 简称ALM) [32, 35] 和乘子交替方向法(Alternating Directions Method of Multipliers, 简称ADMM) [10–12] 分别是求解问题 (1.1) 和 (1.2) 的被广泛认可的算法. 直接推广的 ADMM 用来求解三块的可分离问题 (1.3) 却不能保证收敛 [7]. 对典型凸优化问题, 本文介绍如何利用我们的算法框架去构造更加丰富又便于操作的 ALM, ADMM 类分裂收缩算法, 并说明这些算法之间的来龙去脉和内在关系. S. Becker 在一篇短文[2] 的正文第一句话说: “Recent works such as [HY12] have proposed a very simple yet powerful technique for analysing optimization methods.” 这里的[HY12]就是我们的论文 [28], 本文将比较系统地介绍这类简单而有效的技术, 希望能让中文读者掌握构造求解凸优化一阶算法的基本技术.

内容安排上, 我们在第二节介绍预备知识, 包括线性约束凸优化问题与对应的单调变分不等式, 以及邻近点算法及其主要性质. 第三节中介绍算法的统一框架, 为了文章的完整性, 仍附上了简短的收敛性证明. 第四节, 第五节和第六节分别介绍如何在统一框架指导下构造求解第一、第二和第三类问题的方法. 最后, 研究这些方法的体会和总结在第七节中做了陈述. 在导师何旭初先生诞生百年之际, 谨以此文献给离开我们30年的何先生.

2 预备知识

凸优化的分裂收缩算法所需要的数学基础知识不多. 先说可微凸优化问题

$$\min\{f(x)\mid x\in\mathcal{X}\}, \tag{2.1}$$

其中 $f(x)$ 是可微凸函数, $\mathcal{X}$ 是 $\Re^n$ 中的闭凸集. 我们记问题的解集为 $\mathcal{X}^*$. 用比较通俗的话说, 如果 x^* 是问题(2.1)的解点, 除了它必须属于 $\mathcal{X}$ 外, 同时从这一点出发的任何可行方向都不再是目标函数的下降方向. 我们用

$$S_f(x^*) = \{s \in \Re^n \mid s = x - x^*,\ x \in \mathcal{X}\} \quad \text{和} \quad S_d(x^*) = \{s \in \Re^n \mid s^T \nabla f(x^*) < 0\}$$

分别代表 x^* 处的可行方向集和下降方向集, 那么 $x^* \in \mathcal{X}^*$ 的充分必要条件是

$$x^* \in \mathcal{X} \quad \text{并且} \quad S_f(x^*) \cap S_d(x^*) = \emptyset.$$

等价的形式就是

$$x^* \in \mathcal{X}, \quad (x - x^*)^T \nabla f(x^*) \ge 0, \quad \forall x \in \mathcal{X}.$$

如果用算子 $F(x)$ 代替 $\nabla f(x)$, 就得到变分不等式的一般形式. 基于上面的分析, 用变分不等式判定是否找到 (2.1) 的解点, 可以说成是盲人爬山判定是否已经到达山顶的数学表达形式(只是把求最高变成求最小). 下面的常识性引理是本文算法收敛性分析的基础.

引理2.1 设 $\mathcal{X} \subset \Re^n$ 是简单闭凸集, $\theta(x)$ 和 $f(x)$ 都是 $\Re^n \to \Re$ 的凸函数. 如果 $f(x)$ 在包含 $\mathcal{X}$ 的某个开集上可微并且 $\min\{\theta(x) + f(x) \,|\, x \in \mathcal{X}\}$ 有解, 那么

$$x^* \in \arg\min\{\theta(x) + f(x) \,|\, x \in \mathcal{X}\} \tag{2.2a}$$

的充分必要条件是

$$x^* \in \mathcal{X}, \quad \theta(x) - \theta(x^*) + (x - x^*)^T \nabla f(x^*) \ge 0, \quad \forall x \in \mathcal{X}. \tag{2.2b}$$

引理2.1中对 $\theta(x)$ 是否可微不作要求. 感兴趣的读者可以在作者主页 [38] 第三个报告的第一部分找到完整的证明.

2.1 线性约束凸优化问题和等价的变分不等式

设 $\mathcal{U} \subset \Re^n$ 是简单闭凸集, $\theta: \Re^n \to \Re$ 是(并非一定光滑的)凸函数, $\mathcal{A} \in \Re^{m\times n}$, $b \in \Re^m$. 我们考虑线性约束的凸优化问题

$$\min\{\theta(u) \mid \mathcal{A}u = b \ (\text{or} \ge b), u \in \mathcal{U}\}, \tag{2.3}$$

并假设它有解. 问题(2.3)的拉格朗日 (Lagrange) 函数

$$L(u, \lambda) = \theta(u) - \lambda^T(\mathcal{A}u - b) \tag{2.4}$$

定义在 $\mathcal{U} \times \Lambda$ 上, 其中

$$\Lambda = \begin{cases} \Re^m, & \text{if } \mathcal{A}u = b, \\ \Re^m_+, & \text{if } \mathcal{A}u \ge b. \end{cases}$$

这里的 $\Re^m_+$ 表示 $\Re^m$ 中的非负卦限. 如果一对 (u^*, λ^*) 满足

$$(u^*, \lambda^*) \in \mathcal{U} \times \Lambda, \quad L(u^*, \lambda) \le L(u^*, \lambda^*) \le L(u, \lambda^*), \quad \forall (u, \lambda) \in \mathcal{U} \times \Lambda, \tag{2.5}$$

就称它为 Lagrange 函数(2.4) 的鞍点, 鞍点中的 u^* 就是(2.3)的解. 定义鞍点的不等式(2.5)的两部分写开来就是

$$\begin{cases} u^* \in \mathcal{U}, \quad L(u, \lambda^*) - L(u^*, \lambda^*) \ge 0, \quad \forall\, u \in \mathcal{U}, \\ \lambda^* \in \Lambda, \quad L(u^*, \lambda^*) - L(u^*, \lambda) \ge 0, \quad \forall\, \lambda \in \Lambda. \end{cases}$$

利用拉格朗日函数(2.4)的表达式, 从上式得到下面的变分不等式组

$$\begin{cases} u^* \in \mathcal{U}, \quad \theta(u) - \theta(u^*) + (u - u^*)^T(-\mathcal{A}^T\lambda^*) \ge 0, \quad \forall\, u \in \mathcal{U}, \\ \lambda^* \in \Lambda, \qquad\qquad\qquad\qquad (\lambda - \lambda^*)^T(\mathcal{A}u^* - b) \ge 0, \quad \forall\, \lambda \in \Lambda. \end{cases} \tag{2.6}$$

将上式两部分加在一起写成紧凑的形式就是变分不等式

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \ge 0, \quad \forall\, w \in \Omega, \tag{2.7a}$$

其中

$$w = \begin{pmatrix} u \\ \lambda \end{pmatrix}, \quad F(w) = \begin{pmatrix} -\mathcal{A}^T\lambda \\ \mathcal{A}u - b \end{pmatrix}, \quad \Omega = \mathcal{U} \times \Lambda. \tag{2.7b}$$

对(2.7)中任意的 $w \in \Omega$ 分别取 $w = (u, \lambda^*)$ 和 $w = (u^*, \lambda)$, 就能反过来得到(2.6). 这样, 我们把鞍点的集合和变分不等式的解集建立了等价关系. 通篇, 我们都用 Ω^* 表示变分不等式(2.7)的解集 (因此也就是拉格朗日函数(2.4)鞍点的集合). 注意到(2.7b)中, 仿射算子

$$F(w) = \begin{pmatrix} -\mathcal{A}^T\lambda \\ \mathcal{A}u - b \end{pmatrix} = \begin{pmatrix} 0 & -\mathcal{A}^T \\ \mathcal{A} & 0 \end{pmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} - \begin{pmatrix} 0 \\ b \end{pmatrix}$$

中的矩阵为反对称矩阵, 因此有$(w - \tilde{w})^T(F(w) - F(\tilde{w})) \equiv 0$. 由此而得的恒等式

$$(\tilde{w} - w^*)^T F(\tilde{w}) \equiv (\tilde{w} - w^*)^T F(w^*). \tag{2.8}$$

这个恒等式在后面算法框架的收敛性证明中常常要用到.

2.2 凸优化和变分不等式的邻近点算法

邻近点算法 (Proximal Point Algorithm) 简称 PPA 算法, 是求解(2.2a)中的凸优化问题

$$\min\{\theta(x) + f(x) \mid x \in \mathcal{X}\} \tag{2.9}$$

的一类基本算法 [33,36]. 用 PPA 求解(2.9), 第 k-步迭代是从给定的 x^k 开始, 通过求解

$$\min\{\theta(x)+f(x)+\frac{r}{2}\|x-x^k\|^2 \mid x\in\mathcal{X}\}, \quad (r>0) \tag{2.10}$$

得到 x^{k+1}. PPA 算法的第 k-步迭代的子问题的目标函数, 是在优化问题原目标函数后面加上正则项 $\frac{r}{2}\|x-x^k\|^2$, 使得新的迭代点离原来的迭代点不要太远, 犹如探险过程中步步为营、稳扎稳打. PPA 所产生的序列 $\{x^k\}$ 满足

$$\|x^{k+1}-x^*\|^2 \le \|x^k-x^*\|^2-\|x^k-x^{k+1}\|^2, \tag{2.11}$$

其中 x^* 表示问题(2.9)的任何一个解点, 简单的证明页可以在[38]的第一部分查到. (2.11)式是证明凸优化 PPA 算法收敛的关键不等式. 为了介绍求解变分不等式(2.7)的 PPA 算法, 我们先说一个简单的结论.

引理2.2　设 $a,b\in\Re^n$, $H\in\Re^{n\times n}$ 是对称正定矩阵. 如果 $b^TH(a-b)\ge 0$, 则有

$$\|b\|_H^2 \le \|a\|_H^2-\|a-b\|_H^2. \tag{2.12}$$

证明　对任何给定的正定矩阵H, 有

$$\|a\|_H^2=\|b+(a-b)\|_H^2=\|b\|_H^2+2b^TH(a-b)+\|a-b\|_H^2.$$

引理的结论可以从上面的恒等式和引理的假设条件直接得到.

定义2.1　(求解变分不等式(2.7) H-模下的 PPA 算法)　对给定的正定矩阵$H\succ 0$ 和向量 w^k, 如果一个算法的第 k-步求得的新迭代点 $w^{k+1}\in\Omega$ 满足

$$\theta(u)-\theta(u^{k+1})+(w-w^{k+1})^TF(w^{k+1})\ge(w-w^{k+1})^TH(w^k-w^{k+1}),\ \ \forall\, w\in\Omega, \tag{2.13}$$

则称该算法是求解变分不等式(2.7)的 H-模下的 PPA 算法.

显然, 如果在(2.13)中有 $w^k=w^{k+1}$, 那么 w^{k+1} 就是问题(2.7)的解. PPA 算法的变分不等式(2.13)也可以写成

$$w^{k+1}\in\Omega,\ \theta(u)-\theta(u^{k+1})+(w-w^{k+1})^T\{F(w^{k+1})+H(w^{k+1}-w^k)\}\ge 0,\ \forall\, w\in\Omega, \tag{2.14}$$

其中的 $H(w^{k+1}-w^k)$ 是二次函数 $\frac{1}{2}\|w-w^k\|_H^2$ 在 w^{k+1} 处的梯度, 这也是把(2.13)称为求解变分不等式(2.7)的 H-模下邻近点算法的原因.

引理2.3　对给定的正定矩阵 $H\succ 0$ 和向量 w^k, 如果 w^{k+1} 是由(2.13)提供的, 那么有

$$\|w^{k+1}-w^*\|_H^2\le\|w^k-w^*\|_H^2-\|w^k-w^{k+1}\|_H^2, \quad \forall w^*\in\Omega^*. \tag{2.15}$$

证明　将(2.13)中任意的 $w\in\Omega$ 设为 w^*, 就有

$$(w^{k+1}-w^*)^TH(w^k-w^{k+1})\ge\theta(u^{k+1})-\theta(u^*)+(w^{k+1}-w^*)^TF(w^{k+1}). \tag{2.16}$$

利用$(w^{k+1}-w^*)^T F(w^{k+1}) = (w^{k+1}-w^*)^T F(w^*)$ (见(2.8)), 将(2.16)的右端改写为与它等同的

$$\theta(u^{k+1}) - \theta(u^*) + (w^{k+1}-w^*)^T F(w^*).$$

由 $w^{k+1} \in \Omega$ 和 w^* 是(2.7)的解, 上式非负. 因此从(2.16)得到

$$(w^{k+1}-w^*)^T H(w^k - w^{k+1}) \geq 0.$$

在引理2.2中令 $a = w^k - w^*$ 和 $b = w^{k+1} - w^*$, 从上式就得到引理的结论.

引理2.3说明迭代点离解集里的每一点都越来越近. 具有性质(2.15)的序列 $\{w^k\}$ 被称为在 H-模下 Fejér 单调的, 它是证明迭代序列 $\{w^k\}$ 收敛的关键不等式. 对变分不等式(2.7), 我们在 [28]中首次明确提出了H-模下的 PPA 算法.

3 凸优化分裂收缩算法的统一框架

第二节已经对线性约束的凸优化问题(2.3)和形如(2.7)的单调变分不等式

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w-w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \Omega \tag{3.1}$$

建立了对应关系. 凸优化分裂收缩算法的统一框架聚焦于求解上述变分不等式. 建立这个框架, 是受我们求解经典的单调变分不等式的投影收缩算法框架 [14,18,19] 的影响与启发. 作者一生的研究途径是从单调变分不等式的投影收缩算法到凸优化的分裂收缩算法.

3.1 变分不等式形式下的统一框架

迭代方法中把 v 称为核心变量(essential variables), 是指 k-次迭代可以从给定的 v^k 开始. v 可以是 w 本身, 也可以是 w 的部分分量. 我们把 w 中除了 v 以外的分量(如果有)称为中间变量(intermediate variables). 为了方便收敛性证明, 我们把算法的每步迭代理解成(有时是故意分拆成)预测和校正两部分, 只描述第 k-步迭代如何预测与校正.

[预测] 统一框架算法的第 k-步迭代从给定的核心变量 v^k 开始, 通过求解一些子问题, 产生预测点 $\tilde{w}^k$, 使得

$$\tilde{w}^k \in \Omega, \ \ \theta(u) - \theta(\tilde{u}^k) + (w-\tilde{w}^k)^T F(\tilde{w}^k) \geq (v-\tilde{v}^k)^T Q(v^k - \tilde{v}^k), \ \ \forall\, w \in \Omega, \tag{3.2}$$

成立. 其中 Q 叫做*预测矩阵*, $Q^T + Q$ 是原则上(in principal)正定的.

这里需要解释一下什么叫原则上正定. 在为求解凸优化问题(2.3)设计的分裂算法中, 矩阵 $\mathcal{A}$ 常常需要被分块成 $\mathcal{A} = (A, B)$ 或者 $\mathcal{A} = (A, B, C)$. 当这些分块矩阵都列满秩, 矩

阵 Q^T+Q 就正定, 这时我们就说 Q^T+Q 是原则上正定的. 当那些列满秩条件不满足时, 原则上正定的 Q^T+Q 必须是半正定. 在往后的叙述中, 为了方便, 我们往往会把原则上正定说成正定. 与问题(3.1)对照一下就知道, 如果预测(3.2)中有 $v^k=\tilde{v}^k$, 那么 $\tilde{w}^k$ 就是变分不等式(3.1)的解. 否则, 我们分别采用 *固定步长* 和 *计算步长* 的校正产生新的核心变量 v^{k+1}.

3.1.1　固定步长的校正公式

[**固定步长的校正公式**]. 根据预测得到的 $\tilde{v}^k$, 核心变量 v 的新的迭代点 v^{k+1} 由

$$v^{k+1}=v^k-\alpha M(v^k-\tilde{v}^k) \tag{3.3}$$

给出, 其中的 $\alpha>0$ 和 *校正矩阵* M 要满足以下收敛性条件.

收敛性条件　对 (3.2) 中的预测矩阵 Q 和 (3.3)中的校正矩阵 M, 有正定矩阵

$$H\succ 0 \qquad \text{使得} \qquad HM=Q. \tag{3.4a}$$

此外, 校正公式(3.3)中所取步长 $\alpha>0$ 能够保证

$$G=Q^T+Q-\alpha M^THM\succ 0. \tag{3.4b}$$

上面的 G 正定, 通常情况下也可以只要求原则上正定.

定义3.1　当(3.2)中的预测矩阵 Q 本身就是一个对称正定矩阵时, 我们直接把它记为 H. 在(3.3)中取校正矩阵 $M=I$, $\alpha\in(0,2)$.

$$v^{k+1}=v^k-\alpha(v^k-\tilde{v}^k), \qquad \alpha\in(0,2). \tag{3.5}$$

这样的校正我们称之为**平凡的松弛校正**. 公式(3.3)中校正矩阵 $M\neq I$ 的校正, 都称之为**非平凡的必要校正**.

在平凡的松弛校正中, $H=Q$ 和 $G=(2-\alpha)H$ 都是正定矩阵. 这时, 如果在(3.5)式中取 $\alpha=1$, 就是 H-模下的邻近点 (PPA) 算法. 然而, 计算实践发现, 采用 $\alpha\in[1.2,1.8]$ 的超松弛校正, 方法的效率一般都有 30-50% 的提高. 通常, 我们把预测矩阵 Q 设计成对称正定矩阵的方法称为 *按需定制的邻近点算法* (Customized PPA) [13].

3.1.2　计算步长的校正公式

计算步长的校正都在(3.2)中的预测矩阵 Q 不对称时采用, 因此也都是非平凡的必要校正. 将(3.2)中任意的 w 设为某个任意确定的 $w^*\in\Omega^*$, 我们得到

$$(\tilde{v}^k-v^*)^TQ(v^k-\tilde{v}^k)\geq\theta(\tilde{u}^k)-\theta(u^*)+(\tilde{w}^k-w^*)^TF(\tilde{w}^k). \tag{3.6}$$

如同引理2.3中证明的那样, 上式右端非负. 进而从(3.6)得到

$$(v^k-v^*)^TQ(v^k-\tilde{v}^k)\geq(v^k-\tilde{v}^k)^TQ(v^k-\tilde{v}^k), \quad \forall v^*\in\mathcal{V}^*. \tag{3.7}$$

注意到上式右端等于 $\frac{1}{2}\|v^k-\tilde{v}^k\|^2_{(Q^T+Q)}$. 因此, 对任意给定的正定矩阵$H$, (3.7)式表示

$$\Big\langle \nabla\big(\frac{1}{2}\|v-v^*\|_H^2\big)\big|_{v=v^k},\, H^{-1}Q(v^k-\tilde{v}^k)\Big\rangle \ge \frac{1}{2}\|v^k-\tilde{v}^k\|^2_{(Q^T+Q)}.$$

换句话说, 对任何确定却未知的v^*, 记$M = H^{-1}Q$, $M(v^k-\tilde{v}^k)$ 就是H模下距离函数 $\frac{1}{2}\|v-v^*\|_H^2$ 在v^k处的上升方向. 据此我们得到下面的计算步长的校正公式.

[计算步长的校正公式] 根据预测得到的$\tilde{v}^k$, 核心变量v新的迭代点v^{k+1}由

$$v^{k+1} = v^k - \alpha_k M(v^k-\tilde{v}^k), \quad \alpha_k = \gamma\alpha_k^*, \quad \gamma\in(0,2) \tag{3.8}$$

给出, 其中α_k^* 和校正矩阵 M 要满足下面的收敛性条件.

收敛性条件 对(3.2)中的预测矩阵Q和(3.8)中的校正矩阵M, 有正定矩阵

$$H\succ 0 \qquad \text{使得} \qquad HM = Q. \tag{3.9a}$$

此外, 校正公式(3.8)中,

$$\alpha_k^* = \frac{(v^k-\tilde{v}^k)^TQ(v^k-\tilde{v}^k)}{[M(v^k-\tilde{v}^k)]^TQ(v^k-\tilde{v}^k)}. \tag{3.9b}$$

在计算步长的方法中, 跟用固定步长的校正方法一样, 主要工作还是对预测矩阵Q, 找出一对矩阵H和M, 其中H原则上正定, 并有$HM=Q$. 有了这些, 计算 α_k^* 的工作量是很有限的. 对确定的Q, H和M, 设

$$\alpha_{\max} = \operatorname{argmax}\{\alpha \,|\, Q^T+Q-\alpha M^THM \succeq 0\}. \tag{3.10a}$$

这个大于零的$\alpha_{\max}$ 是下有界的. 另一方面, 由(3.9b)和$Q^TM = M^THM$, 我们有

$$\alpha_k^*(v^k-\tilde{v}^k)^T(M^THM)(v^k-\tilde{v}^k) = \frac{1}{2}(v^k-\tilde{v}^k)^T(Q^T+Q)(v^k-\tilde{v}^k),$$

经整理得,

$$(v^k-\tilde{v}^k)^T(Q^T+Q-2\alpha_k^*M^THM)(v^k-\tilde{v}^k) = 0.$$

因此, 由(3.10a)推得,

$$\alpha_k^* \ge \frac{1}{2}\alpha_{\max}, \quad \forall k>0. \tag{3.10b}$$

由于Q, H和M已经给定, 计算步长的校正公式中, 对所有的k, 步长α_k^*是下有界的.

3.2 统一框架中的算法收敛性质

统一框架采用相同的预测(3.2), 分别采用 固定步长 和 计算步长 的校正产生新的核心变量v^{k+1}, 我们对采用不同校正的方法讨论其收敛性质. 证明的关键技术, 在 [15,16,26] 和 [38] 中已经多次提到. 由于篇幅不长, 为了文章的完整性, 我们仍然给出简要的证明.

3.2.1 采用固定步长校正的算法收敛性

首先对预测-校正方法(3.2)-(3.3)在条件(3.4)成立的情况下证明有关收敛性质.

定理3.1 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式(3.1)的预测-校正方法(3.2)-(3.3)生成的序列. 如果条件(3.4)成立, 那么有

$$\begin{aligned} \tilde{w}^k \in \Omega, \quad & \alpha\big\{\theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^T F(\tilde{w}^k)\big\} \\ & \geq \frac{1}{2}\big(\|v-v^{k+1}\|_H^2-\|v-v^k\|_H^2\big)+\frac{\alpha}{2}\|v^k-\tilde{v}^k\|_G^2, \quad \forall w\in\Omega. \end{aligned} \tag{3.11}$$

证明 利用 $Q=HM$ 和校正公式(3.3)中的 $M(v^k-\tilde{v}^k)=\dfrac{1}{\alpha}(v^k-v^{k+1})$, 我们有

$$Q(v^k-\tilde{v}^k)=HM(v^k-\tilde{v}^k)=\frac{1}{\alpha}H(v^k-v^{k+1}).$$

代入预测公式(3.2)的右端就得到

$$\alpha\{\theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^T F(\tilde{w}^k)\}\geq (v-\tilde{v}^k)^T H(v^k-v^{k+1}), \quad \forall w\in\Omega. \tag{3.12}$$

对上式的右端利用恒等式

$$(a-b)^T H(c-d)=\frac{1}{2}\big(\|a-d\|_H^2-\|a-c\|_H^2\big)+\frac{1}{2}\big(\|b-c\|_H^2-\|b-d\|_H^2\big), \tag{3.13}$$

并令其中的 $a=v$, $b=\tilde{v}^k$, $c=v^k$ 和 $d=v^{k+1}$, 就有

$$\begin{aligned} & (v-\tilde{v}^k)^T H(v^k-v^{k+1}) \\ = \ & \frac{1}{2}\big(\|v-v^{k+1}\|_H^2-\|v-v^k\|_H^2\big)+\frac{1}{2}\big(\|\tilde{v}^k-v^k\|_H^2-\|\tilde{v}^k-v^{k+1}\|_H^2\big). \end{aligned} \tag{3.14}$$

对(3.14)式右端后一个圆括号中的部分利用校正公式(3.3), 得到

$$\begin{aligned} & \|\tilde{v}^k-v^k\|_H^2-\|\tilde{v}^k-v^{k+1}\|_H^2 \\ \overset{(3.3)}{=} \ & \|\tilde{v}^k-v^k\|_H^2-\|(\tilde{v}^k-v^k)+\alpha M(v^k-\tilde{v}^k)\|_H^2 \\ = \ & 2\alpha(v^k-\tilde{v}^k)^T HM(v^k-\tilde{v}^k)-\alpha^2\|M(v^k-\tilde{v}^k)\|_H^2 \\ \overset{(3.4a)}{=} \ & \alpha(v^k-\tilde{v}^k)^T(Q^T+Q-\alpha M^T HM)(v^k-\tilde{v}^k) \\ \overset{(3.4b)}{=} \ & \alpha\|v^k-\tilde{v}^k\|_G^2. \end{aligned} \tag{3.15}$$

将(3.14)和(3.15)代入(3.12), 就得到定理的结论.

定理3.1的证明中用到的“积化和差”恒等式(3.13)虽然很初等, 却非常关键. 这个恒等式也被A. Beck 在他的专著 [1]中采用(见该书pp.428-429)并予以特别标注. 由于 v^* 表示 w^* 的核心变量部分, 我们记 $\mathcal{V}^*=\{v^*|w^*\in\Omega^*\}$. 利用定理3.1, 可以马上得到序列 $\{\|v^k-v^*\|_H^2\}$ 的收缩性质.

定理3.2 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式(2.7)的预测-校正方法(3.2)-(3.3)生成的序列. 如果条件(3.4)成立, 那么有

$$\|v^{k+1}-v^*\|_H^2 \le \|v^k-v^*\|_H^2 - \alpha\|v^k-\tilde{v}^k\|_G^2, \quad \forall v^* \in \mathcal{V}^*. \tag{3.16}$$

证明 将(3.11)中任意的 $w \in \Omega$ 以任意固定的解点 w^* 代入, 我们得到

$$\begin{aligned} &\|v^k-v^*\|_H^2 - \|v^{k+1}-v^*\|_H^2 \\ &\quad \ge \quad \alpha\|v^k-\tilde{v}^k\|_G^2 + 2\alpha\{\theta(\tilde{u}^k)-\theta(u^*)+(\tilde{w}^k-w^*)^TF(\tilde{w}^k)\}. \end{aligned} \tag{3.17}$$

利用$(\tilde{w}^k-w^*)^TF(\tilde{w}^k)=(\tilde{w}^k-w^*)^TF(w^*)$ (见(2.8)) 和 w^* 是最优点的性质, 就有

$$\theta(\tilde{u}^k)-\theta(u^*)+(\tilde{w}^k-w^*)^TF(\tilde{w}^k)=\theta(\tilde{u}^k)-\theta(u^*)+(\tilde{w}^k-w^*)^TF(w^*)\ge 0.$$

代入(3.17)式得到

$$\|v^k-v^*\|_H^2 - \|v^{k+1}-v^*\|_H^2 \ge \alpha\|v^k-\tilde{v}^k\|_G^2.$$

定理得证.

定理3.2的结论是统一框架算法收敛的关键. 收缩不等式(3.16)和PPA 算法的收缩不等式(2.15)有类似的形式, 因此这个统一框架提供的算法也可以看作PPA 算法的延伸与发展. 利用定理3.1 还可以得到算法收敛速率方面的性质 [26, 27], 有兴趣的读者可以在作者主页的 [38] 中查到相关的论述.

3.2.2 采用计算步长校正的算法收敛性

采用计算步长的校正方法, 根据(3.8), v^{k+1} 依赖步长 α_k. 我们定义

$$\vartheta(\alpha_k)=\|v^k-v^*\|_H^2-\|v^{k+1}-v^*\|_H^2, \tag{3.18}$$

利用(3.7)和$HM=Q$, 就有

$$\begin{aligned} \vartheta(\alpha_k) \quad &= \quad \|v^k-v^*\|_H^2-\|(v^k-v^*)-\alpha_kM(v^k-\tilde{v}^k)\|_H^2 \\ &= \quad 2\alpha_k(v^k-v^*)^TQ(v^k-\tilde{v}^k)-\alpha_k^2\|M(v^k-\tilde{v}^k)\|_H^2 \\ &\overset{(3.7)}{\ge} \quad 2\alpha_k(v^k-\tilde{v}^k)^TQ(v^k-\tilde{v}^k)-\alpha_k^2\|M(v^k-\tilde{v}^k)\|_H^2 \\ &=: \quad q(\alpha_k). \end{aligned} \tag{3.19}$$

注意到$q(\alpha_k)$是 α_k 的二次函数并在

$$\alpha_k^*=\frac{(v^k-\tilde{v}^k)^TQ(v^k-\tilde{v}^k)}{\|M(v^k-\tilde{v}^k)\|_H^2}=\frac{(v^k-\tilde{v}^k)^TQ(v^k-\tilde{v}^k)}{(v^k-\tilde{v}^k)^TM^TQ(v^k-\tilde{v}^k)}$$

取到最大值, 这里的 α_k^* 刚好就是(3.9b)中定义的. 取步长 $\alpha_k=\gamma\alpha_k^*$ 的时候, 利用

$$\alpha_k^*(v^k-\tilde{v}^k)^TM^TQ(v^k-\tilde{v}^k)=(v^k-\tilde{v}^k)^TQ(v^k-\tilde{v}^k)$$

推得

$$\vartheta(\alpha_k) \ge q(\alpha_k) = q(\gamma\alpha_k^*) = \gamma(2-\gamma)\alpha_k^*(v^k - \tilde{v}^k)^T Q(v^k - \tilde{v}^k).$$

根据上面的分析, 就有下面的收缩性质的定理.

定理3.3 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式(2.7)由(3.2)预测和(3.8)校正产生的序列, 如果条件(3.9)成立, 那么有

$$\|v^{k+1} - v^*\|_H^2 \le \|v^k - v^*\|_H^2 - \frac{\gamma(2-\gamma)}{2}\alpha_k^*\|v^k - \tilde{v}^k\|_{(Q^T+Q)}^2, \quad \forall v^* \in \mathcal{V}^*. \tag{3.20}$$

由于 $\tilde{w}^k$ 是变分不等式(3.1)解的充分必要条件是 $v^k = \tilde{v}^k$, 定理3.2(或者定理3.3) 的结论就保证了统一框架的算法收敛. 注意到两种方法采用同样的预测(有同样的Q), 校正中也用同样的M(因而有同样的H), 不同的只是取固定步长和计算步长. 后面, 我们每提出和介绍一个方法, 只要验证固定步长算法的收敛条件(3.4). 由(3.10)得知, α_k^* 下有界, 如果采用固定步长校正的算法是收敛的, 对应的计算步长校正的算法同样是收敛的.

4 单块的线性约束凸优化问题

单块的线性约束凸优化问题相当于(2.3)中 $u = x$, $\mathcal{U} = \mathcal{X}$ 和 $\mathcal{A} = A$. 这一节介绍如何用第三节中的统一框架来分析和构造求解这类问题的算法. 对于凸优化问题

$$\min\{\theta(x) \mid Ax = b \text{ (or} \ge b), \ x \in \mathcal{X}\}. \tag{4.1}$$

根据第2.1节的分析, 它对应的拉格朗日函数是定义在 $\mathcal{X} \times \Lambda$ 上的

$$L(x, \lambda) = \theta(x) - \lambda^T(Ax - b). \tag{4.2}$$

相应的变分不等式是

$$w^* \in \Omega, \quad \theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \ge 0, \quad \forall w \in \Omega, \tag{4.3a}$$

其中

$$w = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A^T\lambda \\ Ax - b \end{pmatrix}, \quad \Omega = \mathcal{X} \times \Lambda. \tag{4.3b}$$

注意到, 对等式约束 $Ax = b$ 和不等式约束 $Ax \ge b$, Λ 分别为 $\Re^m$ 和 $\Re_+^m$. 我们用 $P_\Lambda[\cdot]$ 表示向量到 Λ 上的投影. m-维向量到 $\Re^m$ 上的投影是它自身；到 $\Re_+^m$ 上的投影, 只要把小于零的分量置零, 非负分量不变.

4.1 线性等式约束问题的增广拉格朗日乘子法及其平凡松弛

求解等式约束的问题(4.1), 一个被广泛接受的有效算法是增广拉格朗日乘子法 (Augmented Lagrangian Method) [32, 35]. 为此, 需要定义等式约束问题的增广拉格朗日函数

$$\mathcal{L}_\beta(x,\lambda) = \theta(x) - \lambda^T(Ax-b) + \frac{\beta}{2}\|Ax-b\|^2, \tag{4.4}$$

其中 $\beta > 0$ 是理论上可以任意选取, 实际上对算法效率影响较大的常数. 这篇文章对如何合理选取常数 β 不做专门讨论. 增广拉格朗日乘子法的 k-步迭代从给定的 λ^k 开始, 通过

$$\text{(ALM)} \quad \begin{cases} x^{k+1} \in \arg\min\{\mathcal{L}_\beta(x,\lambda^k) \mid x \in \mathcal{X}\}, & \text{(4.5a)} \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1}-b), & \text{(4.5b)} \end{cases}$$

求得 $w^{k+1} = (x^{k+1}, \lambda^{k+1})$. 根据引理2.1和 $\mathcal{L}_\beta(x,\lambda)$ 的表达式, 我们有 $w^{k+1} \in \Omega$ 和

$$\begin{cases} \theta(x) - \theta(x^{k+1}) \ + \ (x - x^{k+1})^T\{-A^T\lambda^k + \beta A^T(Ax^{k+1}-b)\} \geq 0, \ \ \forall x \in \mathcal{X}, \\ \qquad\qquad (\lambda - \lambda^{k+1})^T\{(Ax^{k+1}-b) + (1/\beta)(\lambda^{k+1}-\lambda^k)\} \geq 0, \ \ \forall \lambda \in \Re^m. \end{cases}$$

利用(4.5b)和(4.3), 上述关系式可以写成

$$w^{k+1} \in \Omega, \ \ \theta(x) - \theta(x^{k+1}) + (w - w^{k+1})^T F(w^{k+1}) \geq (\lambda - \lambda^{k+1})^T \frac{1}{\beta}(\lambda^k - \lambda^{k+1}), \ \ \forall\, w \in \Omega. \tag{4.6}$$

由这个与(2.13)类似的不等式, 采用与引理2.3中同样的简单推导, 我们得到

$$\|\lambda^{k+1} - \lambda^*\|^2 \leq \|\lambda^k - \lambda^*\|^2 - \|\lambda^k - \lambda^{k+1}\|^2. \tag{4.7}$$

这是增广拉格朗日乘子法收敛的关键不等式. (4.7)说明增广拉格朗日乘子法是对偶变量 λ 的 PPA 算法. 相对于罚函数方法, 增广拉格朗日乘子法有明显的优势(见 [34] 的第17章). 从博弈的角度看, 罚函数方法只从原始方利益考虑, 增广拉格朗日乘子法同时兼顾了对偶方的利益.

把增广拉格朗日乘子法(4.5)的输出结果记为 $\tilde{w}^k$, 不等式(4.6)就变为

$$\tilde{w}^k \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (\lambda - \tilde{\lambda}^k)^T(1/\beta)(\lambda^k - \tilde{\lambda}^k), \quad \forall\, w \in \Omega. \tag{4.8}$$

这就成了统一框架中形如(3.2)的预测, 其中 $u = x$, $v = \lambda$, $Q = (1/\beta)I_m$. 由于 $(1/\beta)I_m$ 是对称正定矩阵, 采用平凡的松弛校正的方法一般会比经典的增广拉格朗日乘子法收敛更快. 松弛方法具有定理3.2证明的收敛性质. 注意到子问题(4.5a)的目标函数中的二次项是 $\|Ax-b\|^2$, 有时会因为矩阵 A 的结构, 给求解带来一些困难.

4.2 单块线性约束的问题

单块线性 (等式或不等式) 约束的凸优化问题(4.1)对应的变分不等式(4.3)中,

$$\Omega = \mathcal{X} \times \Lambda, \qquad \Lambda = \Re^m \ \text{ or } \ \Re^m_+.$$

当 $\Lambda = \Re^m_+$ 时, 不能直接用第4.1节的方法求解. 我们在第4.2节要介绍的方法中, x-子问题中的二次项中有 $\|A(x-x^k)\|^2$, 求解代价与第4.1节中的增广拉格朗日乘子法相当. 算法中需要的参数 $\beta>0$ 和 $\delta>0$ 理论上都是任意的. 开始一次迭代要从给定的 $w^k=(x^k,\lambda^k)$ 开始, 因此 w 本身是核心变量.

4.2.1　单块线性约束问题平凡松弛的 PPA 算法

求解(与不等式约束凸优化问题(4.1) 对应的) 变分不等式(4.3), 我们设计一个预测矩阵是正定矩阵的 PPA 算法. 迭代从给定的 $w^k=(x^k,\lambda^k)$ 开始, 求得的预测点 $\tilde{w}^k$ 满足

$$\tilde{w}^k\in\Omega,\ \ \theta(x)-\theta(\tilde{x}^k)+(w-\tilde{w}^k)^T\big\{F(\tilde{w}^k)+H(\tilde{w}^k-w^k)\big\}\ge 0,\ \ \forall w\in\Omega, \tag{4.9a}$$

其中预测矩阵

$$H=\begin{pmatrix}(1+\delta)\beta A^TA & A^T\\ A & (1/\beta)I_m\end{pmatrix},\quad \delta>0. \tag{4.9b}$$

当 A 列满秩时上面的矩阵 H 是正定的. 对应于统一框架中的预测, $v=w$. 再用松弛校正(3.5)直接产生新迭代点 w^{k+1} 的算法是能保证收敛的. 问题归结为求满足(4.9)的预测点 $\tilde{w}^k$. 将预测(4.9) 按 F 和 H 的结构分拆开来, 可以写成 $\tilde{w}^k=(\tilde{x}^k,\tilde{\lambda}^k)\in\Omega$,

$$\begin{cases}\theta(x)-\theta(\tilde{x}^k)+\quad(x-\tilde{x}^k)^T\{-A^T\tilde{\lambda}^k\\ \qquad\qquad +(1+\delta)\beta A^TA(\tilde{x}^k-x^k)+A^T(\tilde{\lambda}^k-\lambda^k)\}\ge 0,\quad \forall x\in\mathcal{X},\\ \qquad (\lambda-\tilde{\lambda}^k)^T\{(A\tilde{x}^k-b)+A(\tilde{x}^k-x^k)+(1/\beta)(\tilde{\lambda}^k-\lambda^k)\}\ge 0,\quad \forall\lambda\in\Lambda.\end{cases}$$

归并简化后的形式是 $\tilde{w}^k=(\tilde{x}^k,\tilde{\lambda}^k)\in\Omega$,

$$\begin{cases}\theta(x)-\theta(\tilde{x}^k)+(x-\tilde{x}^k)^T\{-A^T\lambda^k+(1+\delta)\beta A^TA(\tilde{x}^k-x^k)\}\ge 0,\quad \forall x\in\mathcal{X},\\ \qquad (\lambda-\tilde{\lambda}^k)^T\{(A(2\tilde{x}^k-x^k)-b)+(1/\beta)(\tilde{\lambda}^k-\lambda^k)\}\ge 0,\quad \forall\lambda\in\Lambda.\end{cases} \tag{4.10}$$

根据引理2.1, 上式中的 $\tilde{x}^k$ 可以通过求解子问题

$$\min\{\theta(x)-x^TA^T\lambda^k+\frac{1+\delta}{2}\beta\|A(x-x^k)\|^2|x\in\mathcal{X}\}$$

得到. 同样根据引理2.1, (4.10)中的 $\tilde{\lambda}^k$ 是问题

$$\min\Big\{\frac{1}{2\beta}\|\lambda-[\lambda^k-\beta(A(2\tilde{x}^k-x^k)-b)]\|^2\ \Big|\ \lambda\in\Lambda\Big\}$$

的解, 可以通过

$$P_\Lambda[\lambda^k-\beta(A(2\tilde{x}^k-x^k)-b)]$$

直接给出. 所以, 预测点可以由

$$\begin{cases}\tilde{x}^k=\arg\min\{\theta(x)-x^TA^T\lambda^k+\frac{1+\delta}{2}\beta\|A(x-x^k)\|^2|x\in\mathcal{X}\}, & (4.11a)\\ \tilde{\lambda}^k=P_\Lambda[\lambda^k-\beta(A(2\tilde{x}^k-x^k)-b)] & (4.11b)\end{cases}$$

提供, 按照先 $\tilde{x}^k$, 后 $\tilde{\lambda}^k$ 的顺序顺利实现.

如果把(4.9b)中的预测矩阵 H 换成

$$H = \begin{pmatrix} (1+\delta)\beta A^TA & -A^T \\ -A & (1/\beta)I_m \end{pmatrix}, \quad \delta > 0 \tag{4.12}$$

此时矩阵 H 依然正定. 相应的预测点 $\tilde{w}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ 可以通过如下

$$\begin{cases} \tilde{\lambda}^k = P_\Lambda[\lambda^k - \beta(Ax^k - b)], & \text{(4.13a)} \\ \tilde{x}^k = \arg\min\{\theta(x) - x^TA^T(2\tilde{\lambda}^k - \lambda^k) + \dfrac{1+\delta}{2}\beta\|A(x - x^k)\|^2 | x \in \mathcal{X}\}, & \text{(4.13b)} \end{cases}$$

先 $\tilde{\lambda}^k$, 再 $\tilde{x}^k$ 的顺序产生. 采用平凡的松弛校正, 同样给出了一个求解(与不等式凸优化问题(4.1) 相对应的) 变分不等式(4.3) 的收敛算法.

注记 如果将(4.9b)和(4.12)的预测矩阵 H 中的 $(1+\delta)\beta A^TA$ 部分改成 $\beta A^TA + \delta I$, 不管 A 是否列满秩, 矩阵 H 都是正定的. 对于这样改了以后的 H, 只要将相应的 x-子问题目标函数中的 $\frac{1+\delta}{2}\beta\|A(x-x^k)\|^2$ 改成 $\big(\frac{1}{2}\beta\|A(x-x^k)\|^2 + \frac{1}{2}\delta\|x - x^k\|^2\big)$ 就可以.

4.2.2 单块线性约束问题带非平凡校正的算法

从给定的 $w^k = (x^k, \lambda^k)$ 出发, 考虑跟第4.2.1节中(4.13)略有不同的预测

$$\begin{cases} \tilde{\lambda}^k = P_\Lambda[\lambda^k - \beta(Ax^k - b)], & \text{(4.14a)} \\ \tilde{x}^k = \arg\min\{\theta(x) - x^TA^T\tilde{\lambda}^k + \dfrac{1+\delta}{2}\beta\|A(x - x^k)\|^2 | x \in \mathcal{X}\} & \text{(4.14b)} \end{cases}$$

按照先 $\tilde{\lambda}^k$, 后 $\tilde{x}^k$ 的顺序提供. 与预测(4.13)比, 子问题(4.14b)似乎比(4.13b)自然一些. 根据引理2.1, 由(4.14)生成的$\tilde{w}^k = (\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$ 满足

$$\begin{cases} \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{-A^T\tilde{\lambda}^k + (1+\delta)\beta A^TA(\tilde{x}^k - x^k)\} \ge 0, \quad \forall x \in \mathcal{X}, \\ (\lambda - \tilde{\lambda}^k)^T\{(A\tilde{x}^k - b) - A(\tilde{x}^k - x^k) + (1/\beta)(\tilde{\lambda}^k - \lambda^k)\} \ge 0, \quad \forall \lambda \in \Lambda. \end{cases}$$

利用(4.3)的变分不等式形式, 上式可以写成统一框架中的预测形式

$$\tilde{w}^k \in \Omega, \ \ \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T\big\{F(\tilde{w}^k) + Q(\tilde{w}^k - w^k)\big\} \ge 0, \ \forall w \in \Omega, \tag{4.15a}$$

其中预测矩阵

$$Q = \begin{pmatrix} (1+\delta)\beta A^TA & 0 \\ -A & (1/\beta)I_m \end{pmatrix}, \quad \delta > 0. \tag{4.15b}$$

对(4.14)生成的预测点, 我们采用步长 $\alpha = 1$ 的非平凡校正

$$w^{k+1} = w^k - M(w^k - \tilde{w}^k), \tag{4.16a}$$

生成新的迭代点, 其中校正矩阵

$$M=\begin{pmatrix} I_n & 0 \\ -\beta A & I_m \end{pmatrix}. \tag{4.16b}$$

用(3.4) 验证收敛性条件, 利用 $H=QM^{-1}$ 得到

$$H=\begin{pmatrix} (1+\delta)\beta A^TA & 0 \\ -A & (1/\beta)I_m \end{pmatrix}\begin{pmatrix} I_n & 0 \\ \beta A & I_m \end{pmatrix}=\begin{pmatrix} (1+\delta)\beta A^TA & 0 \\ 0 & (1/\beta)I_m \end{pmatrix}\succ 0$$

和

$$\begin{aligned} G &= Q^T+Q-M^THM=Q^T+Q-Q^TM \\ &= \begin{pmatrix} 2(1+\delta)\beta A^TA & -A^T \\ -A & (2/\beta)I_m \end{pmatrix}-\begin{pmatrix} (1+\delta)\beta A^TA & -A^T \\ 0 & (1/\beta)I_m \end{pmatrix}\begin{pmatrix} I_n & 0 \\ -\beta A & I_m \end{pmatrix} \\ &= \begin{pmatrix} \delta\beta A^TA & 0 \\ 0 & (1/\beta)I_m \end{pmatrix}. \end{aligned}$$

矩阵 H 和 G (当A列满秩时) 都正定. 分别用 (4.14) 做预测和 (4.16) 做校正, 我们就得到一个求解(与不等式凸优化问题(4.1)对应的) 变分不等式(4.3) 的收敛算法.

注记 如果将 x-子问题(4.14b)目标函数中的二次项 $\frac{1+\delta}{2}\beta\|A(x-x^k)\|^2$ 改成 $\big(\frac{1}{2}\beta\|A(x-x^k)\|^2+\frac{1}{2}\delta\|x-x^k\|^2\big)$, (4.15b)中预测矩阵 Q 的左上角 $(1+\delta)\beta A^TA$ 部分改成 $\beta A^TA+\delta I$. 保持校正矩阵 M 不变, 这样, 不管 A 是否列满秩, 由 $H=QM^{-1}$ 和 $G=Q^T+Q-M^THM$ 得到的 H 和 G 都是正定的.

4.3 单块线性约束问题简化子问题的方法

第4.1节介绍的求解线性等式约束凸优化问题和第4.2节中介绍求解线性约束凸优化问题(4.1)的算法, 它们的 x-子问题的目标函数中分别包含二次项 $\|Ax-b\|^2$ 和 $\|A(x-x^k)\|^2$. 这一节介绍的算法, 相应的二次项都用 $\|x-x^k\|^2$ 代替, 给出的求解凸优化问题(4.1), 对等式和不等式约束是通用的.

4.3.1 $Q=H$ 的 PPA 算法

为求解变分不等式(4.3), 我们设计一个预测-校正方法, 迭代从给定的 $w^k=(x^k,\lambda^k)$ 开始, 求得预测点满足

$$\tilde{w}^k\in\Omega,\ \ \theta(x)-\theta(\tilde{x}^k)+(w-\tilde{w}^k)^T\big\{F(\tilde{w}^k)+H(\tilde{w}^k-w^k)\big\}\ge 0,\ \ \forall w\in\Omega, \tag{4.17a}$$

其中预测矩阵

$$H=\begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix},\quad rs>\|A^TA\|. \tag{4.17b}$$

这就成了形如(3.2)的预测, 其中 $u = x$, $v = w = (x, \lambda)$. 由于 $Q = H$ 是对称正定矩阵, 就像在第3.1节中讨论的, 新的迭代点 w^{k+1} 由平凡的松弛校正(3.5) 提供. 问题就归结为如何构造满足(4.17) 的预测方法. 根据变分不等式(4.3)的表达式, 预测(4.17)是求 $\tilde{w}^k \in \Omega$ 使得

$$\begin{cases} \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{-A^T\tilde{\lambda}^k + r(\tilde{x}^k - x^k) + A^T(\tilde{\lambda}^k - \lambda^k)\} \ge 0, \quad \forall x \in \mathcal{X}, \\ \qquad\qquad (\lambda - \tilde{\lambda}^k)^T\{(A\tilde{x}^k - b) + A(\tilde{x}^k - x^k) + s(\tilde{\lambda}^k - \lambda^k)\} \ge 0, \quad \forall \lambda \in \Lambda. \end{cases}$$

进一步化简归并就是

$$\begin{cases} \tilde{x}^k \in \mathcal{X}, \quad \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{-A^T\lambda^k + r(\tilde{x}^k - x^k)\} \ge 0, \quad \forall x \in \mathcal{X}, \\ \tilde{\lambda}^k \in \Lambda, \qquad (\lambda - \tilde{\lambda}^k)^T\{[A(2\tilde{x}^k - x^k) - b] + s(\tilde{\lambda}^k - \lambda^k)\} \ge 0, \quad \forall \lambda \in \Lambda. \end{cases}$$

根据引理2.1, 这样的 $\tilde{w}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ 可以按顺序由

$$\begin{cases} \tilde{x}^k = \arg\min\big\{\theta(x) - x^TA^T\lambda^k + \dfrac{r}{2}\|x - x^k\|^2 \;\big|\; x \in \mathcal{X}\big\}, & (4.18a) \\ \tilde{\lambda}^k = \arg\min\big\{\lambda^T[A(2\tilde{x}^k - x^k) - b] + \dfrac{s}{2}\|\lambda - \lambda^k\|^2 \mid \lambda \in \Lambda\big\} & (4.18b) \end{cases}$$

提供. 因此, 通过(4.18)预测, 再由平凡的松弛校正, 就提供了求解变分不等式(4.3)的一个收敛的算法. 预测步的 x-子问题(4.18a)中, 二次项 $\|x - x^k\|^2$ 比增广拉格朗日乘子法中的 $\|Ax - b\|^2$ 简单. 但要保证(4.17b)中的矩阵 H 正定, 就需要 $rs > \|A^TA\|$, 这往往会影响收敛速度.

针对(4.1)中等式约束 $Ax = b$ 和不等式约束 $Ax \ge b$ 两种情形, 这一节子问题(4.18b)得到 $\tilde{\lambda}^k$ 的差别仅在

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A(2\tilde{x}^k - x^k) - b] \qquad 和 \qquad \tilde{\lambda}^k = \big\{\lambda^k - \frac{1}{s}[A(2\tilde{x}^k - x^k) - b]\big\}_+.$$

直接将(4.18)产生的预测点当新的迭代点的*一阶原始-对偶方法*, Chambolle 和 Pock 在 [5] 中已经提及, 也给了遍历意义下的收敛性证明. 我们在 PPA 意义下给出简单的收缩意义下的收敛性证明 [28], 并指出新迭代点可以通过进一步平凡松弛延拓得到. 这个证明思路得到了 Chambolle 和 Pock 的高度认可并帮助和促进了他们的后继研究 [6].

4.3.2 Q 为上三角阵的预测校正方法

原始对偶混合梯度法 (PDHG) [37], 也是 [5]中一阶原始-对偶方法的一个特例. 它的 k-步迭代, 从给定的 $w^k = (x^k, \lambda^k)$ 开始, 利用(4.2)中定义的拉格朗日函数, 通过

$$\begin{cases} x^{k+1} = \arg\min\big\{L(x, \lambda^k) + \dfrac{r}{2}\|x - x^k\|^2 \;\big|\; x \in \mathcal{X}\big\}, \\ \lambda^{k+1} = \arg\max\big\{L(x^{k+1}, \lambda) - \dfrac{s}{2}\|\lambda - \lambda^k\|^2 \;\big|\; \lambda \in \Lambda\big\} \end{cases} \tag{4.19}$$

得到 $w^{k+1} = (x^{k+1}, \lambda^{k+1}) \in \Omega$. 有例子说明, PDHG 求解线性规划不能保证收敛 [38]. 不过, 将 PDHG 的输出当作预测点, 再进行适当校正, 是可以改造成一个收敛算法的. 注意到

改变目标函数中的常数项并不影响问题的解这个事实, 利用 $L(x,\lambda)$ 的表达式并将 PDHG 的输出记作 $\tilde{w}^k=(\tilde{x}^k,\tilde{\lambda}^k)$, 就有

$$\left\{\begin{array}{ll}\tilde{x}^k=\arg\min\big\{\theta(x)-x^TA^T\lambda^k+\frac{r}{2}\|x-x^k\|^2 \;\big|\; x\in\mathcal{X}\big\}, & \text{(4.20a)}\\ \tilde{\lambda}^k=\arg\min\big\{\lambda^T(A\tilde{x}^k-b)+\frac{s}{2}\|\lambda-\lambda^k\|^2 \;\big|\; \lambda\in\Lambda\big\}. & \text{(4.20b)}\end{array}\right.$$

根据引理2.1, 这样的 $\tilde{w}^k=(\tilde{x}^k,\tilde{\lambda}^k)\in\Omega$ 满足

$$\left\{\begin{array}{lr}\tilde{x}^k\in\mathcal{X}, \quad \theta(x)-\theta(\tilde{x}^k)+(x-\tilde{x}^k)^T\{-A^T\lambda^k+r(\tilde{x}^k-x^k)\}\ge 0, & \forall x\in\mathcal{X},\\ \tilde{\lambda}^k\in\Lambda, \quad (\lambda-\tilde{\lambda}^k)^T\{(A\tilde{x}^k-b)+s(\tilde{\lambda}^k-\lambda^k)\}\ge 0, & \forall\lambda\in\Lambda.\end{array}\right.$$

利用(4.3)中 $F(\tilde{w}^k)$ 的表达式, 就可以把上式写成统一框架中的预测形式

$$\tilde{w}^k\in\Omega,\ \ \theta(x)-\theta(\tilde{x}^k)+(w-\tilde{w}^k)^T\big\{F(\tilde{w}^k)+Q(\tilde{w}^k-w^k)\big\}\ge 0,\ \forall w\in\Omega, \tag{4.21a}$$

其中预测矩阵

$$Q=\begin{pmatrix} rI & A^T\\ 0 & sI\end{pmatrix} \tag{4.21b}$$

为上三角矩阵. 由于 Q 非对称, 我们采用步长 $\alpha=1$ 的非平凡的校正

$$w^{k+1}=w^k-M(w^k-\tilde{w}^k), \tag{4.22a}$$

去生成新的迭代点, 其中校正矩阵

$$M=\begin{pmatrix} I_n & \frac{1}{r}A^T\\ 0 & I_m\end{pmatrix}. \tag{4.22b}$$

算法的收敛性只要用条件(3.4)去验证. 由于

$$H=QM^{-1}=\begin{pmatrix} rI_n & A^T\\ 0 & sI_m\end{pmatrix}\begin{pmatrix} I_n & -\frac{1}{r}A^T\\ 0 & I_m\end{pmatrix}=\begin{pmatrix} rI_n & 0\\ 0 & sI_m\end{pmatrix}\succ 0$$

和

$$G = Q^T+Q-M^THM=Q^T+Q-Q^TM=\begin{pmatrix} rI_n & 0\\ 0 & sI_m-\frac{1}{r}AA^T\end{pmatrix}.$$

当 $rs>\|A^TA\|$ 时, G 就是正定的. 因此, 分别用(4.20)做预测和(4.22)做校正, 我们得到一个求解 (与凸优化问题(4.1)对应的) 变分不等式(4.3)的收敛算法.

我们也可以将校正公式(4.22)中的校正矩阵 M 改成

$$M=\begin{pmatrix} I_n & 0\\ -\frac{1}{s}A & I_m\end{pmatrix}.$$

这时, 对(4.21b)中的预测矩阵 Q, 我们有

$$H = QM^{-1} = \begin{pmatrix} rI_n & A^T \\ 0 & sI_m \end{pmatrix} \begin{pmatrix} I_n & 0 \\ \frac{1}{s}A & I_m \end{pmatrix} = \begin{pmatrix} rI_n + \frac{1}{s}A^TA & A^T \\ A & sI_m \end{pmatrix} \succ 0$$

和

$$G = Q^T + Q - M^THM = Q^T + Q - Q^TM = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix}.$$

同样的预测, 不同的校正, 方法都是收敛的.

同样, 针对(4.1)中等式约束 $Ax = b$ 和不等式约束 $Ax \geq b$ 两种情形, 子问题(4.20b)中给出 $\tilde{\lambda}^k$ 的差别仅在

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}(A\tilde{x}^k - b) \qquad \text{和} \qquad \tilde{\lambda}^k = \left[\lambda^k - \frac{1}{s}(A\tilde{x}^k - b)\right]_+.$$

4.3.3 Q 是对角正定矩阵和反对称矩阵之和的情形

第4.3.1节和第4.3.2节中介绍的方法, 都要求 $rs > \|A^TA\|$, 当矩阵 A^TA 的最大特征值和平均特征值之比大一些的时候, 往往会导致收敛很慢. 如果预测的格式满足

$$\tilde{w}^k \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T\left\{F(\tilde{w}^k) + Q(\tilde{w}^k - w^k)\right\} \geq 0, \quad \forall w \in \Omega, \tag{4.23a}$$

其中预测矩阵

$$Q = \begin{pmatrix} rI_n & A^T \\ -A & sI_m \end{pmatrix} \tag{4.23b}$$

是块对角数量矩阵和反对称矩阵之和. 对任意的 $r, s > 0$, $Q^T + Q$ 都是正定的. 利用 $F(\tilde{w}^k)$ 的表达式, 将(4.23)写开来就是

$$\begin{cases} \tilde{x}^k \in \mathcal{X}, & \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{-A^T\lambda^k + r(\tilde{x}^k - x^k)\} \geq 0, \quad \forall x \in \mathcal{X}, \\ \tilde{\lambda}^k \in \Lambda, & (\lambda - \tilde{\lambda}^k)^T\{(Ax^k - b) + s(\tilde{\lambda}^k - \lambda^k)\} \geq 0, \quad \forall \lambda \in \Lambda. \end{cases}$$

根据引理2.1, 通过

$$\begin{cases} \tilde{x}^k = \arg\min\left\{\theta(x) - x^TA^T\lambda^k + \frac{r}{2}\|x - x^k\|^2 \;\middle|\; x \in \mathcal{X}\right\}, & (4.24a) \\ \tilde{\lambda}^k = \arg\min\left\{\lambda^T(Ax^k - b) + \frac{s}{2}\|\lambda - \lambda^k\|^2 \;\middle|\; \lambda \in \Lambda\right\}, & (4.24b) \end{cases}$$

就能得到这样的预测点. (4.23) 中预测矩阵 Q 的具体形式给构造满足统一框架条件的矩阵 H 和相应的校正矩阵 M 提供了许多可能性. 例如, 取

$$H = \begin{pmatrix} rI_n & 0 \\ 0 & sI_m \end{pmatrix} \quad \text{和} \quad M = \begin{pmatrix} I_n & \frac{1}{r}A^T \\ -\frac{1}{s}A & I_m \end{pmatrix}$$

就有$HM=Q$, 剩下的事就可以按统一框架第3.2.2节中指出的用计算步长的方法去校正. 同样, 针对(4.1)中等式约束$Ax=b$和不等式约束$Ax\ge b$两种情形, 子问题(4.24b)中给出$\tilde\lambda^k$的差别仅在

$$\tilde\lambda^k=\lambda^k-\frac{1}{s}(Ax^k-b)\qquad 和 \qquad \tilde\lambda^k=\big[\lambda^k-\frac{1}{s}(Ax^k-b)\big]_+.$$

注记　在第4节的算法中, 预测分别由(4.18),(4.20)和(4.24)实现. 注意到它们的x-子问题完全相同. λ-子问题的线性项部分则由

$$\lambda^T[A(2\tilde x^k-x^k)-b],\quad 到\quad \lambda^T[A\tilde x^k-b],\quad 再到\quad \lambda^T[Ax^k-b].$$

它们可以分别被看成

$$\lambda^T[(Ax^k-b)+lA(\tilde x^k-x^k)],\quad l=2,1,0,$$

表明了这几个算法在统一框架下内在的联系. 如果对(4.17b)中的矩阵H, 以及(4.21b) 和(4.23b)中的预测矩阵Q, 改成

$$H=\begin{pmatrix} rI_n & -A^T\\ -A & sI_m\end{pmatrix},\quad Q=\begin{pmatrix} rI_n & 0\\ -A & sI_m\end{pmatrix}\quad 和\quad Q=\begin{pmatrix} rI_n & -A^T\\ A & sI_m\end{pmatrix},$$

读者可以根据第4.3节的分析和演绎, 设计相应的收敛算法.

5　求解可分离两块凸优化问题的乘子交替方向法

这一节讨论可分离成两块线性约束的凸优化问题

$$\min\{\theta_1(x)+\theta_2(y)\mid Ax+By=b,\, x\in\mathcal{X}, y\in\mathcal{Y}\}\tag{5.1}$$

的求解方法. 问题相当于在(2.3)中, 置$n=n_1+n_2$, $\mathcal{X}\subset\Re^{n_1}$, $\mathcal{Y}\subset\Re^{n_2}$, $\mathcal{U}=\mathcal{X}\times\mathcal{Y}$. $\theta(u)=\theta_1(x)+\theta_2(y)$, $\theta_1(x):\Re^{n_1}\to\Re$, $\theta_2(y):\Re^{n_2}\to\Re$. 矩阵$\mathcal{A}=(A,B)$, 其中$A\in\Re^{m\times n_1}$, $B\in\Re^{m\times n_2}$. 对这个问题, 变分不等式一般形式(2.7)就具体化为

$$w^*\in\Omega,\quad \theta(u)-\theta(u^*)+(w-w^*)^TF(w^*)\ge 0,\quad \forall\, w\in\Omega,\tag{5.2a}$$

其中

$$w=\begin{pmatrix} x\\ y\\ \lambda\end{pmatrix},\quad w=\begin{pmatrix} x\\ y\end{pmatrix},\quad F(w)=\begin{pmatrix} -A^T\lambda\\ -B^T\lambda\\ Ax+By-b\end{pmatrix},\quad \Omega=\mathcal{X}\times\mathcal{Y}\times\Re^m.\tag{5.2b}$$

利用问题(5.1)的增广拉格朗日函数

$$\mathcal{L}_\beta^{[2]}(x,y,\lambda)=\theta_1(x)+\theta_2(y)-\lambda^T(Ax+By-b)+\frac{\beta}{2}\|Ax+By-b\|^2, \tag{5.3}$$

经典的乘子交替方向法 [10–12] 可以简单地表示成

$$\text{(ADMM)}\quad \left\{\begin{array}{ll} x^{k+1}=\arg\min\big\{\mathcal{L}_\beta^{[2]}(x,y^k,\lambda^k)\ \big|\ x\in\mathcal{X}\big\}, & (5.4\text{a})\\ y^{k+1}=\arg\min\big\{\mathcal{L}_\beta^{[2]}(x^{k+1},y,\lambda^k)\ \big|\ y\in\mathcal{Y}\big\}, & (5.4\text{b})\\ \lambda^{k+1}=\lambda^k-\beta(Ax^{k+1}+By^{k+1}-b). & (5.4\text{c}) \end{array}\right.$$

因为k-次迭代从给定的 (y^k,λ^k) 就可以开始, $v=(y,\lambda)$ 被说成是核心变量, 而 x 只是迭代过程中的中间变量.

由于改变目标函数中的常数项不影响问题的解, 利用$\mathcal{L}_\beta^{[2]}(x,y,\lambda)$ 的表达式, 可以把迭代公式(5.4)写成等价的

$$\left\{\begin{array}{ll} x^{k+1}=\text{argmin}\big\{\theta_1(x)-x^TA^T\lambda^k+\dfrac{\beta}{2}\|Ax+By^k-b\|^2\big|x\in\mathcal{X}\big\}, & (5.5\text{a})\\ y^{k+1}=\text{argmin}\big\{\theta_2(y)-y^TB^T\lambda^k+\dfrac{\beta}{2}\|Ax^{k+1}+By-b\|^2\big|y\in\mathcal{Y}\big\}, & (5.5\text{b})\\ \lambda^{k+1}=\lambda^k-\beta(Ax^{k+1}+By^{k+1}-b). & (5.5\text{c}) \end{array}\right.$$

子问题(5.5a)和(5.5b)分别只含变量 x 和 y. 经典的 ADMM 算法有如下的收敛性关键结论.

定理5.1 设$\{w^{k+1}=(x^{k+1},y^{k+1},\lambda^{k+1})\}\in\Omega$ 是用算法(5.4)求解变分不等式(5.2) 生成的迭代序列, 那么有

$$\|v^{k+1}-v^*\|_H^2\le\|v^k-v^*\|_H^2-\|v^k-v^{k+1}\|_H^2,\quad \forall\, v^*\in\mathcal{V}^*. \tag{5.6}$$

其中

$$v=\begin{pmatrix} y\\ \lambda\end{pmatrix},\quad H=\begin{pmatrix}\beta B^TB & 0\\ 0 & \frac{1}{\beta}I_m\end{pmatrix},\quad \mathcal{V}^*=\{v^*=(y^*,\lambda^*)|(x^*,y^*,\lambda^*)\in\Omega^*\}.$$

定理5.1中的矩阵 H 就是原则上正定的, 比较简单的证明可以在 [38]中找到. Boyd 等在他们的综述文章 [3] 中对 ADMM 算法做了详细的介绍. 在算法(5.4)的基础上, 我们在[4,20,21]中发表了一些进一步提高计算效率的 ADMM 类算法.

这一节根据第3节中的统一框架, 介绍如何设计不同的 ADMM 类算法. 在算法(模型)设计中, 我们统一把(5.4)中容易求解的那个子问题安排成 x-子问题并假设(5.5a)的求解没有什么困难. 都把核心变量定成 $v=(y,\lambda)$, k-次迭代从给定的 $v^k=(y^k,\lambda^k)$ 开始, 完成一步迭代得到 $v^{k+1}=(y^{k+1},\lambda^{k+1})$. 后面三小节中提出的算法, x-子问题都跟经典 ADMM 中的(5.5a)一样. 注意到 ADMM 中 y-子问题(5.5b)的等价形式也可以写成

$$y^{k+1}=\text{argmin}\big\{\theta_2(y)-y^TB^T[\lambda^k-\beta(Ax^{k+1}+By^k-b)]+\frac{1}{2}\beta\|B(y-y^k)\|^2\big|y\in\mathcal{Y}\big\}. \tag{5.7}$$

目标函数中的二次项是 $\|B(y-y^k)\|^2$.

5.1 利用统一框架设计的 PPA 类算法

求解变分不等式(5.2)的 PPA 型算法要求(3.2)中的预测矩阵 Q 本身就是一个对称正定矩阵. 这时, 我们把相应的矩阵 Q 记为 H. 这类方法中, 我们用平凡松弛的校正(3.5) 给出 v^{k+1}. 实际运算中, 一般取 $\alpha\in[1.2,1.8]$.

5.1.1 预测中 y-子问题中二次项是 $\|B(y-y^k)\|^2$ 的算法

如果我们为求解(5.2)构造的预测公式中 $\tilde{w}^k$ 满足

$$\tilde{w}^k\in\Omega,\quad \theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^TF(\tilde{w}^k)\ge(v-\tilde{v}^k)^TH(v^k-\tilde{v}^k),\quad \forall w\in\Omega, \tag{5.8a}$$

其中预测矩阵

$$H=\begin{pmatrix}(1+\delta)\beta B^TB & -B^T\\ -B & (1/\beta)I_m\end{pmatrix},\quad (\delta>0) \tag{5.8b}$$

那么当 B 列满秩时矩阵 H 正定. 我们用平凡松弛的校正(3.3)得到新的迭代点 v^{k+1}. 根据统一框架, 算法就是收敛的. 因此, 问题归结为如何实现满足(5.8)的预测. 用(5.2)中 $F(w)$ 的表达式, 把(5.8)的具体形式写出来就是 $\tilde{w}^k=(\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)\in\Omega$, 使得

$$\begin{cases}\theta_1(x)-\theta_1(\tilde{x}^k)+(x-\tilde{x}^k)^T\{\underline{-A^T\tilde{\lambda}^k}\}\ge 0,\quad \forall x\in\mathcal{X}, & (5.9\text{a})\\ \theta_2(y)-\theta_2(\tilde{y}^k)+(y-\tilde{y}^k)^T\{\underline{-B^T\tilde{\lambda}^k} & \\ \qquad\qquad +(1+\delta)\beta B^TB(\tilde{y}^k-y^k)-B^T(\tilde{\lambda}^k-\lambda^k)\}\ge 0,\quad \forall y\in\mathcal{Y}, & (5.9\text{b})\\ \qquad (\underline{A\tilde{x}^k+B\tilde{y}^k-b})\ -B(\tilde{y}^k-y^k)\quad +\quad (1/\beta)\,(\tilde{\lambda}^k-\lambda^k)=0. & (5.9\text{c})\end{cases}$$

上式中, 有下划线的凑在一起, 就是(5.8a)中的 $F(\tilde{w}^k)$. 按照下面 x,λ,y 的次序计算:

$$\begin{cases}\tilde{x}^k=\operatorname{argmin}\{\theta_1(x)-x^TA^T\lambda^k+\frac{1}{2}\beta\|Ax+By^k-b\|^2\mid x\in\mathcal{X}\}, & (5.10\text{a})\\ \tilde{\lambda}^k=\lambda^k-\beta(A\tilde{x}^k+By^k-b), & (5.10\text{b})\\ \tilde{y}^k=\operatorname{argmin}\big\{\theta_2(y)-y^TB^T[2\tilde{\lambda}^k-\lambda^k]+\frac{1+\delta}{2}\beta\|B(y-y^k)\|^2\,\big|\,y\in\mathcal{Y}\big\}, & (5.10\text{c})\end{cases}$$

就得到满足条件(5.9)的预测点. 上式中 y-子问题(5.10c)和经典 ADMM 中(5.7)难度相当. 根据引理2.1, x-子问题(5.10a)的最优性条件是

$$\tilde{x}^k\in\mathcal{X},\quad \theta_1(x)-\theta_1(\tilde{x}^k)+(x-\tilde{x}^k)^T\{-A^T\lambda^k+\beta A^T(A\tilde{x}^k+By^k-b)\}\ge 0,\quad \forall x\in\mathcal{X}.$$

利用 $\tilde{\lambda}^k$ 的预测式(5.10b), 上式就可以改写成(5.9a). 注意到(5.10b)本身就是(5.9c). 剩下的(5.9b), 根据引理2.1, 可以通过求解 y-子问题(5.10c)去实现. 所以, 按照(5.10)中以 $\tilde{x}^k$, $\tilde{\lambda}^k$, $\tilde{y}^k$ 顺序生成的预测点 $\tilde{w}^k=(\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)$ 满足预测条件(5.8). 由于可以采用松弛校正, 在其他条件相同的情况下, 算法效率比经典的 ADMM (5.5) 有较明显的提高.

5.1.2 预测中 y-子问题简化的算法

要实现第5.1.1节中的预测(5.8), y-子问题的目标函数中含有二次项$\|B(y-y^k)\|^2$. 如果因此而给求解带来不小困难, 就将预测公式(5.8)改成

$$\tilde{w}^k \in \Omega, \quad \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (v - \tilde{v}^k)^T H(v^k - \tilde{v}^k), \quad \forall w \in \Omega, \tag{5.11a}$$

其中预测矩阵

$$H = \begin{pmatrix} rI_{n_2} & -B^T \\ -B & (1/\beta)I_m \end{pmatrix}. \tag{5.11b}$$

当$r > \beta\|B^TB\|$, 上面的矩阵H是正定的. 同样是用校正公式(3.5)给出新的迭代点v^{k+1}. 根据统一框架, 算法就是收敛的. 因此, 问题归结为如何实现满足(5.11)的预测. 如同第5.1.1节中的分析, 按照下面x, λ, y 的次序计算:

$$\left\{\begin{array}{ll} \tilde{x}^k = \text{argmin}\{\theta_1(x) - x^TA^T\lambda^k + \frac{1}{2}\beta\|Ax + By^k - b\|^2 \,|\, x \in \mathcal{X}\}, & (5.12\text{a}) \\ \tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k - b), & (5.11\text{b}) \\ \tilde{y}^k = \text{argmin}\big\{\theta_2(y) - y^TB^T[2\tilde{\lambda}^k - \lambda^k] + \frac{r}{2}\|y - y^k\|^2 \,\big|\, y \in \mathcal{Y}\big\}, & (5.12\text{c}) \end{array}\right.$$

就可以得到满足(5.11)的预测点. 这样做, y-子问题(5.12c)比(5.10c) 简单了一些.

需要指出的是, 如果第5.1.1节中y-子问题(5.10c)求解并不困难的话, 我们仍然主张用第5.1.1节中的方法. 因为(5.12c)中要求$r > \beta\|B^TB\|$, 往往会影响收敛速度.

5.2 非平凡校正的方法

求解变分不等式(5.2)的非平凡校正的方法允许预测(3.2)中的预测矩阵Q非对称, 并且一般是分块下三角矩阵. 这一节都采用步长$\alpha = 1$的校正公式(3.3), 其中校正矩阵

$$M = \begin{pmatrix} I_{n_2} & 0 \\ -\beta B & I_m \end{pmatrix}. \tag{5.13}$$

对这样的单位下三角矩阵M, 逆矩阵$M^{-1} = \begin{pmatrix} I_{n_2} & 0 \\ \beta B & I_m \end{pmatrix}$.

5.2.1 预测中 y-子问题中二次项是 $\|B(y-y^k)\|^2$ 的算法

预测中y-子问题难度和经典 ADMM 中(5.7)相当, 是指子问题目标函数中含有二次项$\|B(y-y^k)\|^2$. 设我们构造的预测公式产生的预测点$\tilde{w}^k$满足

$$\tilde{w}^k \in \Omega, \quad \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (v - \tilde{v}^k)^T Q(v^k - \tilde{v}^k), \quad \forall w \in \Omega, \tag{5.14a}$$

其中预测矩阵

$$Q = \begin{pmatrix} (1+\delta)\beta B^TB & 0 \\ -B & (1/\beta)I_m \end{pmatrix}, \quad (\delta > 0). \tag{5.14b}$$

当 B 列满秩时, 矩阵 Q^T+Q 正定. 然后用 $\alpha=1$ 的公式(3.3)进行校正, 其中校正矩阵 M 由(5.13) 给出. 用统一框架中的条件(3.4) 去验证收敛性. 矩阵 H 和 G 分别是

$$H=QM^{-1}=\begin{pmatrix}(1+\delta)\beta B^TB & 0\\ -B & (1/\beta)I_m\end{pmatrix}\begin{pmatrix}I_{n_2} & 0\\ \beta B & I_m\end{pmatrix}=\begin{pmatrix}(1+\delta)\beta B^TB & 0\\ 0 & (1/\beta)I_m\end{pmatrix}$$

和

$$G \;=\; Q^T+Q-M^THM=Q^T+Q-Q^TM=\begin{pmatrix}\delta\beta B^TB & 0\\ 0 & (1/\beta)I_m\end{pmatrix}.$$

在 B 列满秩时, 矩阵 H 和 G 就是正定的. 根据统一框架, 算法就是收敛的. 因此, 问题归结为如何实现满足(5.14)的预测. 使用(5.2)中 $F(w)$ 的表达式, 把(5.14)的具体形式写出来就是 $\tilde{w}^k=(\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)\in\Omega$, 使得

$$\left\{\begin{array}{lr}\theta_1(x)-\theta_1(\tilde{x}^k)+(x-\tilde{x}^k)^T\{\underline{-A^T\tilde{\lambda}^k}\}\ge 0, \quad \forall\, x\in\mathcal{X}, & (5.15a)\\ \theta_2(y)-\theta_2(\tilde{y}^k)+(y-\tilde{y}^k)^T\{\underline{-B^T\tilde{\lambda}^k}+(1+\delta)\beta B^TB(\tilde{y}^k-y^k)\}\ge 0,\ \forall y\in\mathcal{Y}, & (5.15b)\\ \qquad (\underline{A\tilde{x}^k+B\tilde{y}^k-b})-B(\tilde{y}^k-y^k)+(1/\beta)\,(\tilde{\lambda}^k-\lambda^k)=0. & (5.15c)\end{array}\right.$$

上式中, 有下划线的凑在一起, 就是(5.14a) 中的 $F(\tilde{w}^k)$. 就像第5.1.1节中解释的, 按照下面 x,λ,y 的次序计算:

$$\left\{\begin{array}{lr}\tilde{x}^k=\mathrm{argmin}\{\theta_1(x)-x^TA^T\lambda^k+\frac{1}{2}\beta\|Ax+By^k-b\|^2\,|\,x\in\mathcal{X}\}, & (5.16a)\\ \tilde{\lambda}^k=\lambda^k-\beta(A\tilde{x}^k+By^k-b), & (5.16b)\\ \tilde{y}^k=\mathrm{argmin}\big\{\theta_2(y)-y^TB^T\tilde{\lambda}^k+\frac{1+\delta}{2}\beta\|B(y-y^k)\|^2\,\big|\,y\in\mathcal{Y}\big\}, & (5.16c)\end{array}\right.$$

就得到满足条件(5.15)的预测点. 上式中 y-子问题(5.16c)和经典 ADMM 中(5.7)难度相当. 把预测(5.14)和步长 $\alpha=1$, M 由(5.13) 给出的校正(3.3)放在一起, 略去预测点记号, 这一节的迭代格式也可以直接写成

$$\left\{\begin{array}{rcl}x^{k+1} &=& \mathrm{argmin}\{\theta_1(x)-x^TA^T\lambda^k+\frac{1}{2}\beta\|Ax+By^k-b\|^2\,|\,x\in\mathcal{X}\},\\ \lambda^{k+\frac{1}{2}} &=& \lambda^k-\beta(Ax^{k+1}+By^k-b),\\ y^{k+1} &=& \mathrm{argmin}\big\{\theta_2(y)-y^TB^T\lambda^{k+\frac{1}{2}}+\frac{(1+\delta)}{2}\beta\|B(y-y^k)\|^2\,\big|\,y\in\mathcal{Y}\big\},\\ \lambda^{k+1} &=& \lambda^k-\beta(Ax^{k+1}+By^{k+1}-b).\end{array}\right.$$

5.2.2 预测中 y-子问题简化的算法

要实现第5.2.1节中的预测(5.14a), y-子问题的目标函数中含有二次项 $\|B(y-y^k)\|^2$. 如果因此而给求解带来不小困难, 就将预测公式(5.14)改成

$$\tilde{w}^k\in\Omega,\quad \theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^TF(\tilde{w}^k)\ge(v-\tilde{v}^k)^TQ(v^k-\tilde{v}^k),\quad \forall w\in\Omega, \qquad (5.17a)$$

其中预测矩阵

$$Q = \begin{pmatrix} rI_{n_2} & 0 \\ -B & (1/\beta)I_m \end{pmatrix}, \tag{5.17b}$$

当 $r > \beta\|B^TB\|$ 时矩阵 $Q^T + Q$ 正定. 同样用 $\alpha = 1$ 的公式(3.3)进行校正, 其中校正矩阵 M 由(5.13)给出. 我们用统一框架中的条件(3.4)去验证收敛性. 矩阵 H 和 G 分别是

$$H = QM^{-1} = \begin{pmatrix} rI_{n_2} & 0 \\ -B & (1/\beta)I_m \end{pmatrix}\begin{pmatrix} I_n & 0 \\ \beta B & I_m \end{pmatrix} = \begin{pmatrix} rI_{n_2} & 0 \\ 0 & (1/\beta)I_m \end{pmatrix},$$

和

$$G = Q^T + Q - M^THM = Q^T + Q - Q^TM = \begin{pmatrix} rI_{n_2} - \beta B^TB & 0 \\ 0 & (1/\beta)I_m \end{pmatrix}.$$

当$r > \beta\|B^TB\|$ 时矩阵H 和G 都是正定的. 根据统一框架, 算法就是收敛的. 因此, 问题归结为如何实现满足(5.17) 的预测. 如同第5.2.1节中的分析, 按照下面 x, λ, y 的次序计算:

$$\begin{cases} \tilde{x}^k = \mathrm{argmin}\{\theta_1(x) - x^TA^T\lambda^k + \frac{1}{2}\beta\|Ax + By^k - b\|^2 \,|\, x \in \mathcal{X}\}, & (5.18a) \\ \tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k - b), & (5.18b) \\ \tilde{y}^k = \mathrm{argmin}\big\{\theta_2(y) - y^TB^T\tilde{\lambda}^k + \frac{r}{2}\|y - y^k\|^2 \,\big|\, y \in \mathcal{Y}\big\}, & (5.18c) \end{cases}$$

就得到满足条件(5.17)的预测点. 把预测(5.17)和步长 $\alpha = 1$, M 由(5.13)给出的校正(3.3)放在一起, 略去预测点记号, 这一节的迭代格式也可以直接写成

$$\begin{cases} x^{k+1} = \mathrm{argmin}\{\theta_1(x) - x^TA^T\lambda^k + \frac{\beta}{2}\|Ax + By^k - b\|^2 \,|\, x \in \mathcal{X}\}, \\ \lambda^{k+\frac{1}{2}} = \lambda^k - \beta(Ax^{k+1} + By^k - b), \\ y^{k+1} = \mathrm{argmin}\big\{\theta_2(y) - y^TB^T\lambda^{k+\frac{1}{2}} + \frac{1}{2}r\|y - y^k\|^2 \,\big|\, y \in \mathcal{Y}\big\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). \end{cases} \tag{5.19}$$

如果 y-子问题的求解不会因为目标函数的二次项中有 $\|B(y - y^k)\|^2$ 而对求解带来不小的困难, 我们仍主张用第5.2.1节中的方法. 因为 $r > \beta\|B^TB\|$ 这样的要求, 往往会影响收敛速度.

注记 方法(5.19)还能进一步的改进. 所谓最优线性化ADMM [20], 实际上证明了对(5.19)的 y-子问题二次项 $\|y-y^k\|^2$ 的系数 r, 前面还能乘上一个区间 $(0.75, 1)$ 中的因子 τ.

方法变成

$$\begin{cases} x^{k+1} &= \operatorname{argmin}\{\theta_1(x) - x^TA^T\lambda^k + \frac{\beta}{2}\|Ax + By^k - b\|^2 \,|\, x \in \mathcal{X}\}, \\ \lambda^{k+\frac{1}{2}} &= \lambda^k - \beta(Ax^{k+1} + By^k - b), \\ y^{k+1} &= \operatorname{argmin}\big\{\theta_2(y) - y^TB^T\lambda^{k+\frac{1}{2}} + \frac{1}{2}\tau r\|y - y^k\|^2 \,\big|\, y \in \mathcal{Y}\big\}, \\ \lambda^{k+1} &= \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). \end{cases} \tag{5.20}$$

文章中我们用例子指出, 当上式中 $\tau r < 0.75\beta\|B^TB\|$ 的时候, 方法(5.20)可能不收敛. 论文 [20]中的证明要在第3节统一框架的基础上做进一步的延伸分析, 限于篇幅这里不做介绍.

另外我们注意到, 第5.1节的(5.8b)和(5.11b)中的矩阵 H 分别是

$$\begin{pmatrix} (1+\delta)\beta B^TB & -B^T \\ -B & \frac{1}{\beta}I_m \end{pmatrix} \quad 和 \quad \begin{pmatrix} rI_{n_2} & -B^T \\ -B & \frac{1}{\beta}I_m \end{pmatrix}.$$

第5.2节的(5.14b)和(5.17b) 中的预测矩阵 Q 分别是

$$\begin{pmatrix} (1+\delta)\beta B^TB & 0 \\ -B & \frac{1}{\beta}I_m \end{pmatrix} \quad 和 \quad \begin{pmatrix} rI_{n_2} & 0 \\ -B & \frac{1}{\beta}I_m \end{pmatrix}.$$

这四个分块矩阵的第二行完全相同. 左上角分别是 $(1+\delta)\beta B^TB$ 和 rI_{n_2} 两种选择, 右上角分别是 $-B^T$ 和 0 两种选择, 由此而得到四种方法.

6　可分离三块的凸优化问题的分裂收缩算法

可分离三块的线性等式约束约束凸优化问题, 其数学形式为

$$\min\{\theta_1(x) + \theta_2(y) + \theta_3(z) | Ax + By + Cz = b,\ x \in \mathcal{X}, y \in \mathcal{Y}, z \in \mathcal{Z}\}. \tag{6.1}$$

这相当于在(2.3)中, 置 $n = n_1 + n_2 + n_3$, $\mathcal{X} \subset \mathfrak{R}^{n_1}$, $\mathcal{Y} \subset \mathfrak{R}^{n_2}$, $\mathcal{Z} \subset \mathfrak{R}^{n_3}$, $\mathcal{U} = \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. $\theta(u) = \theta_1(x) + \theta_2(y) + \theta_3(z)$, $\theta_1(x)$: $\mathfrak{R}^{n_1} \to \mathfrak{R}$, $\theta_2(y)$: $\mathfrak{R}^{n_2} \to \mathfrak{R}$, $\theta_3(z)$: $\mathfrak{R}^{n_3} \to \mathfrak{R}$. 矩阵 $\mathcal{A} = (A, B, C)$, 其中 $A \in \mathfrak{R}^{m\times n_1}$, $B \in \mathfrak{R}^{m\times n_2}$, $C \in \mathfrak{R}^{m\times n_3}$. 这个问题的拉格朗日函数是

$$L^{[3]}(x, y, z, \lambda) = \theta_1(x) + \theta_2(y) + \theta_3(z) - \lambda^T(Ax + By + Cz - b).$$

问题(6.1)的变分不等式具体形式就成为

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^TF(w^*) \geq 0, \quad \forall\, w \in \Omega, \tag{6.2a}$$

其中

$$w = \begin{pmatrix} x \\ y \\ z \\ \lambda \end{pmatrix}, \quad u = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A^T\lambda \\ -B^T\lambda \\ -C^T\lambda \\ Ax+By+Cz-b \end{pmatrix}, \tag{6.2b}$$

和

$$\theta(u) = \theta_1(x)+\theta_2(y)+\theta_3(z), \qquad \Omega = \mathcal{X}\times\mathcal{Y}\times\mathcal{Z}\times\Re^m. \tag{6.2c}$$

相应的增广拉格朗日函数记为(有别与可分离两块的优化问题中的$\mathcal{L}_\beta^{[2]}(x,y,\lambda)$)记为

$$\mathcal{L}_\beta^{[3]}(x,y,z,\lambda) = \theta_1(x)+\theta_2(y)+\theta_3(z)-\lambda^T(Ax+By+Cz-b)+\frac{\beta}{2}\|Ax+By+Cz-b\|^2.$$

对可分离三块的凸优化问题, 采用直接推广的乘子交替方向法, 核心变量是 $v=(y,z,\lambda)$. 第 k-步迭代是从给定的 $v^k=(y^k,z^k,\lambda^k)$ 出发, 通过

$$\left\{\begin{array}{ll} x^{k+1} = \arg\min\big\{\mathcal{L}_\beta^{[3]}(x,y^k,z^k,\lambda^k)\ \big|\ x\in\mathcal{X}\big\}, & (6.3a)\\ y^{k+1} = \arg\min\big\{\mathcal{L}_\beta^{[3]}(x^{k+1},y,z^k,\lambda^k)\ \big|\ y\in\mathcal{Y}\big\}, & (6.3b)\\ z^{k+1} = \arg\min\big\{\mathcal{L}_\beta^{[3]}(x^{k+1},y^{k+1},z,\lambda^k)\ \big|\ z\in\mathcal{Z}\big\}, & (6.3c)\\ \lambda^{k+1} = \lambda^k-\beta(Ax^{k+1}+By^{k+1}+Cz^{k+1}-b) & (6.3d) \end{array}\right.$$

求得核心变量的新的迭代点 $v^{k+1}=(y^{k+1},z^{k+1},\lambda^{k+1})$. 直接推广的 ADMM (6.3)对三块的可分离问题(6.1)当矩阵A,B,C 中有两个的列空间正交的时候才保证收敛, 这时三块的可分离问题实际上相当于两块的问题. 在一般情形下, 方法(6.3)对三块的问题(6.1)是不能保证收敛的 [7]. 在明确 [7]中的结论前后, 我们发表了一些修正的 ADMM 类算法 [22,23,29]求解三个和三个以上可分离块的凸优化问题.

后面我们介绍如何利用第3节中的统一框架直接构造求解变分不等式(6.2)的收敛方法. 我们总假设子问题(6.3a)是不难求解的. 在子问题(6.3b)和(6.3c)的目标函数中, 二次项分别是 $\|B(y-y^k)\|^2$ 和 $\|C(z-z^k)\|^2$.

6.1 利用统一框架设计的 PPA 类算法

求解变分不等式(6.2)的 PPA 型算法要求(3.2)中的预测矩阵 Q 本身就是一个对称正定矩阵. 这时, 我们就把预测中相应的 Q 矩阵记为 H. 这类方法中, 我们用平凡松弛的校正(3.3)得到 v^{k+1}. 实际运算中, 一般取 $\alpha\in[1.2,1.8]$.

6.1.1 预测中子问题难度和直接推广的 ADMM 中相当的算法

这一小节介绍的利用统一框架构造求解变分不等式(6.2)的方法, y 和 z-子问题的目标函数中分别含有二次项 $\|B(y-y^k)\|^2$ 和 $\|C(z-z^k)\|^2$. 如果我们为求解(6.2)构造的预测公式提供的 $\tilde{w}^k$ 满足

$$\tilde{w}^k\in\Omega, \quad \theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^TF(\tilde{w}^k)\ge(v-\tilde{v}^k)^TH(v^k-\tilde{v}^k), \quad \forall w\in\Omega, \tag{6.4a}$$

其中预测矩阵

$$H=\begin{pmatrix} 2(1+\delta)\beta B^TB & 0 & -B^T \\ 0 & 2(1+\delta)\beta C^TC & -C^T \\ -B & -C & (1/\beta)I_m \end{pmatrix}, \quad (\delta>0). \tag{6.4b}$$

当 B 和 C 列满秩时矩阵 H 正定. 用平凡松弛的校正(3.5)给出新的迭代点 v^{k+1}. 根据统一框架, 算法就是收敛的. 因此, 问题归结为如何实现满足(6.4)的预测. 利用(6.2)中 $F(w)$ 的表达式把(6.4)的具体形式写出来就是 $\tilde{w}^k=(\tilde{x}^k,\tilde{y}^k,\tilde{z}^k,\tilde{\lambda}^k)\in\Omega$, 满足

$$\begin{cases} \theta_1(x)-\theta_1(\tilde{x}^k)+(x-\tilde{x}^k)^T\{\underline{-A^T\tilde{\lambda}^k}\}\ge 0, \quad \forall x\in\mathcal{X}, & (6.5\text{a})\\ \theta_2(y)-\theta_2(\tilde{y}^k)+(y-\tilde{y}^k)^T\{\underline{-B^T\tilde{\lambda}^k} & \\ \qquad +2(1+\delta)\beta B^TB(\tilde{y}^k-y^k)-B^T(\tilde{\lambda}^k-\lambda^k)\}\ge 0, \ \ \forall y\in\mathcal{Y}, & (6.5\text{b})\\ \theta_3(z)-\theta_3(\tilde{z}^k)+(z-\tilde{z}^k)^T\{\underline{-C^T\tilde{\lambda}^k} & \\ \qquad +2(1+\delta)\beta C^TC(\tilde{z}^k-z^k)-C^T(\tilde{\lambda}^k-\lambda^k)\}\ge 0, \ \ \forall z\in\mathcal{Z}, & (6.5\text{c})\\ (\underline{A\tilde{x}^k+B\tilde{y}^k+C\tilde{z}^k-b})-B(\tilde{y}^k-y^k)-C(\tilde{z}^k-z^k)+(1/\beta)(\tilde{\lambda}^k-\lambda^k)=0. & (6.5\text{d}) \end{cases}$$

上式中, 有下划线的凑在一起, 就是(6.4a)中的 $F(\tilde{w}^k)$. 按照下面 x,λ,y,z 的次序计算:

$$\begin{cases} \tilde{x}^k=\operatorname{argmin}\{\theta_1(x)-x^TA^T\lambda^k+\frac{1}{2}\beta\|Ax+By^k+Cz^k-b\|^2\,|\,x\in\mathcal{X}\}, & (6.6\text{a})\\ \tilde{\lambda}^k=\lambda^k-\beta(A\tilde{x}^k+By^k+Cz^k-b), & (6.6\text{b})\\ \tilde{y}^k=\operatorname{argmin}\big\{\theta_2(y)-y^TB^T[2\tilde{\lambda}^k-\lambda^k]+(1+\delta)\beta\|B(y-y^k)\|^2\,\big|\,y\in\mathcal{Y}\big\}, & (6.6\text{c})\\ \tilde{z}^k=\operatorname{argmin}\big\{\theta_3(z)-z^TC^T[2\tilde{\lambda}^k-\lambda^k]+(1+\delta)\beta\|C(z-z^k)\|^2\,\big|\,z\in\mathcal{Z}\big\}, & (6.6\text{d}) \end{cases}$$

就得到满足条件(6.5)的预测点. 事实上, 根据引理2.1, x-子问题(6.6a)的最优性条件是

$$\tilde{x}^k\in\mathcal{X},\ \ \theta_1(x)-\theta_1(\tilde{x}^k)+(x-\tilde{x}^k)^T\{-A^T\lambda^k+\beta A^T(A\tilde{x}^k+By^k+Cz^k-b)\}\ge 0,\ \ \forall x\in\mathcal{X}.$$

利用 $\tilde{\lambda}^k$ 的预测式(6.6b), 上式就可以写成(6.5a). 注意到(6.6b)本身就是(6.5d). 剩下的(6.5b)和(6.5c), 根据引理2.1, 可以分别通过求解y-子问题(6.6c)和z-子问题(6.6d)去实现. 所以, 按照(6.6)中以 x,λ,y,z 的顺序生成的预测点 $\tilde{w}^k=(\tilde{x}^k,\tilde{y}^k,\tilde{z}^k,\tilde{\lambda}^k)$ 满足预测条件(6.4).

6.1.2 预测中子问题简化的算法

简化预测中的子问题, 就是只改动(6.4)中的预测矩阵, 使得

$$\tilde{w}^k\in\Omega,\quad \theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^TF(\tilde{w}^k)\ge(v-\tilde{v}^k)^TH(v^k-\tilde{v}^k),\quad \forall w\in\Omega, \tag{6.7a}$$

其中预测矩阵

$$H=\begin{pmatrix} 2rI_{n_2} & 0 & -B^T \\ 0 & 2sI_{n_3} & -C^T \\ -B & -C & (1\beta)I_m \end{pmatrix}. \tag{6.7b}$$

当 $r > \beta\|B^TB\|$ 和 $s > \beta\|C^TC\|$ 时, 上面的矩阵 H 是正定的. 同样用平凡松弛的校正(3.5)给出新的迭代点 v^{k+1}. 根据统一框架, 算法就是收敛的. 因此, 问题归结为如何实现满足(6.7)的预测. 如同第5.1.1节中的分析, 按照下面 x, λ, y, z 的次序计算:

$$
\begin{cases}
\tilde{x}^k = \mathrm{argmin}\{\theta_1(x) - x^TA^T\lambda^k + \frac{1}{2}\beta\|Ax + By^k + Cz^k - b\|^2 \,|\, x \in \mathcal{X}\}, & (6.8a)\\
\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k + Cz^k - b), & (6.8b)\\
\tilde{y}^k = \mathrm{argmin}\big\{\theta_2(y) - y^TB^T[2\tilde{\lambda}^k - \lambda^k] + r\|y - y^k\|^2 \,\big|\, y \in \mathcal{Y}\big\}, & (6.8c)\\
\tilde{z}^k = \mathrm{argmin}\big\{\theta_3(z) - z^TB^T[2\tilde{\lambda}^k - \lambda^k] + s\|z - z^k\|^2 \,\big|\, z \in \mathcal{Z}\big\}, & (6.8d)
\end{cases}
$$

就可以得到满足(6.7)的预测点. 从(6.4)到(6.7), 同时简化了(6.6)中的 y-子问题和 z-子问题. 假如只需要简化其中一个子问题的时候, 读者应该明白相应的预测该如何处理.

6.2 利用统一框架设计非平凡校正的方法

这一节介绍的方法, 都在预测的基础上采用步长 $\alpha = 1$ 的非平凡的校正(3.3)去生成新的迭代点 v^{k+1}, 其中校正矩阵

$$
M = \begin{pmatrix} I_{n_2} & 0 & 0 \\ 0 & I_{n_3} & 0 \\ -\beta B & -\beta C & I_m \end{pmatrix}. \tag{6.9}
$$

6.2.1 预测中子问题难度和直接推广的ADMM 相当的算法

我们为求解(6.2)构造的预测公式提供的 $\tilde{w}^k$ 满足

$$
\tilde{w}^k \in \Omega, \quad \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^TF(\tilde{w}^k) \geq (v - \tilde{v}^k)^TQ(v^k - \tilde{v}^k), \quad \forall w \in \Omega, \tag{6.10a}
$$

其中预测矩阵

$$
Q = \begin{pmatrix} 2(1+\delta)\beta B^TB & 0 & 0 \\ 0 & 2(1+\delta)\beta C^TC & 0 \\ -B & -C & (1/\beta)I_m \end{pmatrix}, \quad (\delta > 0) \tag{6.10b}
$$

当 B 和 C 列满秩时矩阵 $Q^T + Q$ 正定. 然后用 $\alpha = 1$ 的校正公式(3.3)进行校正, 其中校正矩阵 M 由(6.9)给出. 用统一框架中的条件(3.4)去验证收敛性. 由于校正矩阵 M 由(6.9)给

出, H 和 G 分别是

$$
\begin{aligned}
H &= QM^{-1} = \begin{pmatrix} 2(1+\delta)\beta B^TB & 0 & 0 \\ 0 & 2(1+\delta)\beta C^TC & 0 \\ -B & -C & (1/\beta)I_m \end{pmatrix}\begin{pmatrix} I_{n_2} & 0 & 0 \\ 0 & I_{n_3} & 0 \\ \beta B & \beta C & I_m \end{pmatrix} \\
&= \begin{pmatrix} 2(1+\delta)\beta B^TB & 0 & 0 \\ 0 & 2(1+\delta)\beta C^TC & 0 \\ 0 & 0 & (1/\beta)I_m \end{pmatrix},
\end{aligned}
$$

和

$$
\begin{aligned}
G &= Q^T + Q - Q^TM = \begin{pmatrix} (1+2\delta)\beta B^TB & -\beta B^TC & 0 \\ -\beta C^TB & (1+2\delta)\beta C^TC & 0 \\ 0 & 0 & (1/\beta)I_m \end{pmatrix} \\
&= \begin{pmatrix} 2\delta\beta B^TB & 0 & 0 \\ 0 & 2\delta\beta C^TC & 0 \\ 0 & 0 & (1/\beta)I_m \end{pmatrix} + \beta\begin{pmatrix} B^TB & -B^TC & 0 \\ -C^TB & C^TC & 0 \\ 0 & 0 & 0 \end{pmatrix}.
\end{aligned}
$$

当 B 和 C 列满秩时, 矩阵 H 和 G 就是正定的. 根据统一框架, 算法就是收敛的. 因此, 问题归结为如何实现满足(6.10)的预测. 把(6.10)的具体形式写出来就是 $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k) \in \Omega$, 使得

$$
\begin{cases}
\theta_1(x) - \theta_1(\tilde{x}^k) + (x-\tilde{x}^k)^T\{\underline{-A^T\tilde{\lambda}^k}\} \ge 0, \quad \forall x \in \mathcal{X}, & (6.11\text{a}) \\
\theta_2(y) - \theta_2(\tilde{y}^k) + (y-\tilde{y}^k)^T\{\underline{-B^T\tilde{\lambda}^k} + 2(1+\delta)\beta B^TB(\tilde{y}^k - y^k)\} \ge 0, \ \forall y \in \mathcal{Y}, & (6.11\text{b}) \\
\theta_3(z) - \theta_3(\tilde{z}^k) + (z-\tilde{z}^k)^T\{\underline{-C^T\tilde{\lambda}^k} + 2(1+\delta)\beta C^TC(\tilde{z}^k - z^k)\} \ge 0, \ \forall z \in \mathcal{Z}, & (6.11\text{c}) \\
(\underline{A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b}) - B(\tilde{y}^k - y^k) - C(\tilde{z}^k - z^k) + (1/\beta)(\tilde{\lambda}^k - \lambda^k) = 0. & (6.11\text{d})
\end{cases}
$$

上式中, 有下划线的凑在一起, 就是 $F(\tilde{w}^k)$. 就像第5.1.1节中解释的, 按照下面 x, λ, y, z 的次序计算:

$$
\begin{cases}
\tilde{x}^k = \operatorname{argmin}\{\theta_1(x) - x^TA^T\lambda^k + \frac{1}{2}\beta\|Ax + By^k + Cz^k - b\|^2 \,|\, x \in \mathcal{X}\}, & (6.12\text{a}) \\
\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k + Cz^k - b), & (6.12\text{b}) \\
\tilde{y}^k = \operatorname{argmin}\big\{\theta_2(y) - y^TB^T\tilde{\lambda}^k + (1+\delta)\beta\|B(y-y^k)\|^2 \,\big|\, y \in \mathcal{Y}\big\}, & (6.12\text{c}) \\
\tilde{z}^k = \operatorname{argmin}\big\{\theta_3(z) - z^TC^T\tilde{\lambda}^k + (1+\delta)\beta\|C(z-z^k)\|^2 \,\big|\, z \in \mathcal{Z}\big\}, & (6.12\text{d})
\end{cases}
$$

就得到满足条件(6.11)的预测点. 把预测(6.10)和步长 $\alpha = 1$, 校正矩阵 M 由(6.9)给出的校

正(3.3)放在一起, 略去预测点记号, 这一节的迭代格式也可以直接写成

$$\left\{\begin{array}{lcl} x^{k+1} & = & \mathrm{argmin}\{\theta_1(x) - x^TA^T\lambda^k + \frac{1}{2}\beta\|Ax+By^k+Cz^k-b\|^2 \,|\, x\in\mathcal{X}\}, \\ \lambda^{k+\frac{1}{2}} & = & \lambda^k - \beta(Ax^{k+1}+By^k+Cz^k-b), \\ y^{k+1} & = & \mathrm{argmin}\big\{\theta_2(y) - y^TB^T\lambda^{k+\frac{1}{2}} + (1+\delta)\beta\|B(y-y^k)\|^2 \,\big|\, y\in\mathcal{Y}\big\}, \\ z^{k+1} & = & \mathrm{argmin}\big\{\theta_3(z) - z^TC^T\lambda^{k+\frac{1}{2}} + (1+\delta)\beta\|C(z-z^k)\|^2 \,\big|\, z\in\mathcal{Z}\big\} \\ \lambda^{k+1} & = & \lambda^k - \beta(Ax^{k+1}+By^{k+1}+Cz^{k+1}-b). \end{array}\right. \tag{6.13}$$

这完全等同于 [23] 中给出的方法. 利用统一框架, 收敛性分析和证明就非常简单. 该方法被S. Osher 课题组采用, 花很大篇幅阐述怎样用来解决他们的问题, 给了不错的评价 [9].

6.2.2 预测中简化子问题的算法

如果我们为求解(6.2)构造的预测公式满足

$$\tilde{w}^k\in\Omega,\quad \theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^TF(\tilde{w}^k)\ge(v-\tilde{v}^k)^TQ(v^k-\tilde{v}^k),\quad \forall w\in\Omega, \tag{6.14a}$$

其中预测矩阵

$$Q=\begin{pmatrix} 2rI_{n_2} & 0 & 0 \\ 0 & 2sI_{n_3} & 0 \\ -B & -C & (1/\beta)I_m \end{pmatrix}. \tag{6.14b}$$

我们还是取 $r>\beta\|B^TB\|$ 和 $s>\beta\|C^TC\|$, 这时 Q^T+Q 正定. 然后用 $\alpha=1$ 的公式(3.3)进行校正, 其中校正矩阵 M 由(6.9)给出. 用统一框架中的条件(3.4)去验证收敛性. 矩阵 H 和 G 分别是

$$H=QM^{-1}=\begin{pmatrix} 2rI_{n_2} & 0 & 0 \\ 0 & 2sI_{n_3} & 0 \\ -B & -C & (1/\beta)I_m \end{pmatrix}\begin{pmatrix} I_{n_2} & 0 & 0 \\ 0 & I_{n_3} & 0 \\ \beta B & \beta C & I_m \end{pmatrix}=\begin{pmatrix} 2rI_{n_2} & 0 & 0 \\ 0 & 2sI_{n_3} & 0 \\ 0 & 0 & (1/\beta)I_m \end{pmatrix},$$

和

$$\begin{aligned} G &= Q^T+Q-M^THM=Q^T+Q-Q^TM \\ &= \begin{pmatrix} 4rI_{n_2} & 0 & -B^T \\ 0 & 4sI_{n_3} & -C^T \\ -B & -C & (2/\beta)I_m \end{pmatrix} - \begin{pmatrix} 2rI_{n_2}+\beta B^TB & \beta B^TC & -B^T \\ \beta C^TB & 2sI_{n_3}+\beta C^TC & -C^T \\ -B & -C & (1/\beta)I_m \end{pmatrix} \\ &= \begin{pmatrix} 2(rI_{n_2}-\beta B^TB) & 0 & 0 \\ & 2(sI_{n_3}-\beta C^TC) & 0 \\ 0 & 0 & (1/\beta)I_m \end{pmatrix} + \beta\begin{pmatrix} B^TB & -B^TC & 0 \\ -C^TB & C^TC & 0 \\ 0 & 0 & 0 \end{pmatrix}. \end{aligned}$$

当 $r > \beta\|B^TB\|$ 和 $s > \beta\|C^TC\|$ 时, 矩阵 H 和 G 就是正定的. 根据统一框架, 方法就是收敛的. 因此, 问题归结为如何实现满足(6.14)的预测. 如同第5.2.1节中的分析, 按照下面 x, λ, y, z 的次序计算:

$$\begin{cases} \tilde{x}^k = \text{argmin}\{\theta_1(x) - x^TA^T\lambda^k + \frac{1}{2}\beta\|Ax + By^k + Cz^k - b\|^2 \,|\, x \in \mathcal{X}\}, & (6.15\text{a}) \\ \tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k + Cz^k - b), & (6.15\text{b}) \\ \tilde{y}^k = \text{argmin}\big\{\theta_2(y) - y^TB^T\tilde{\lambda}^k + r\|B(y - y^k)\|^2 \,\big|\, y \in \mathcal{Y}\big\}, & (6.15\text{c}) \\ \tilde{z}^k = \text{argmin}\big\{\theta_3(z) - z^TC^T\tilde{\lambda}^k + s\|C(z - z^k)\|^2 \,\big|\, z \in \mathcal{Z}\big\}, & (6.15\text{d}) \end{cases}$$

就得到满足条件(6.14)的预测点. 把预测(6.14)和步长 $\alpha = 1$, 校正矩阵 M 由(6.9)给出的校正(3.3)放在一起, 略去预测点记号, 这一节的迭代格式也可以直接写成

$$\begin{cases} x^{k+1} & = & \text{argmin}\{\theta_1(x) - x^TA^T\lambda^k + \frac{1}{2}\beta\|Ax + By^k + Cz^k - b\|^2 \,|\, x \in \mathcal{X}\}, \\ \lambda^{k+\frac{1}{2}} & = & \lambda^k - \beta(Ax^{k+1} + By^k + Cz^k - b), \\ y^{k+1} & = & \text{argmin}\big\{\theta_2(y) - y^TB^T\lambda^{k+\frac{1}{2}} + r\|y - y^k\|^2 \,\big|\, y \in \mathcal{Y}\big\}, \\ z^{k+1} & = & \text{argmin}\big\{\theta_3(z) - z^TC^T\lambda^{k+\frac{1}{2}} + s\|z - z^k\|^2 \,\big|\, z \in \mathcal{Z}\big\}, \\ \lambda^{k+1} & = & \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b). \end{cases} \quad (6.16)$$

这实际上是算法(6.13) 的同时简化了 y-子问题和 z-子问题的情形. 假如只需要简化其中一个子问题的时候, 相应的算法应该如何处理, 读者应该自己能够推导.

方法(6.16)还能进一步的改进. 关于ADMM 类算法最优正则化因子 [30], 实际上证明了对(6.16)的 y-子问题二次项 $\|y - y^k\|^2$ 的系数 r, z-子问题二次项 $\|z - z^k\|^2$ 的系数 s, 前面还能乘上一个 $(0.5, 1)$ 的因子 τ. 方法变成

$$\begin{cases} x^{k+1} & = & \text{argmin}\{\theta_1(x) - x^TA^T\lambda^k + \frac{1}{2}\beta\|Ax + By^k + Cz^k - b\|^2 \,|\, x \in \mathcal{X}\}, \\ \lambda^{k+\frac{1}{2}} & = & \lambda^k - \beta(Ax^{k+1} + By^k + Cz^k - b), \\ y^{k+1} & = & \text{argmin}\big\{\theta_2(y) - y^TB^T\lambda^{k+\frac{1}{2}} + \tau r\|y - y^k\|^2 \,\big|\, y \in \mathcal{Y}\big\}, \\ z^{k+1} & = & \text{argmin}\big\{\theta_3(z) - z^TC^T\lambda^{k+\frac{1}{2}} + \tau s\|z - z^k\|^2 \,\big|\, z \in \mathcal{Z}\big\}, \\ \lambda^{k+1} & = & \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b). \end{cases} \quad (6.17)$$

文章中我们用例子指出, 当上式中 $\tau r < 0.5\beta\|B^TB\|$ 和 $\tau s < 0.5\beta\|C^TC\|$ 的时候, 方法会不收敛. 论文 [30] 中的证明要在第3节统一框架的基础上做比较精细的延伸分析, 限于篇幅这里也不做介绍.

7 结论和思考

在线性约束的凸优化问题有解的假设下, 对问题的 Lagrange 函数(2.5) 的鞍点和变分不等式(2.7)的解点建立了等价关系. 对于任意的 $(u,\lambda)\in\Omega$ 和 $(u^*,\lambda^*)\in\Omega^*$, 根据(2.5), 有

$$L(u,\lambda^*)-L(u^*,\lambda)\geq 0.$$

这个量称为点 $(u,\lambda)\in\Omega$ 的对偶间隙. 利用第2.1节中的推导, 对任意给定的 $(u,\lambda)\in\Omega$,

$$\begin{aligned}L(u,\lambda^*)-L(u^*,\lambda) &= [L(u,\lambda^*)-L(u^*,\lambda^*)]+[L(u^*,\lambda^*)-L(u^*,\lambda)]\\ &= [\theta(u)-\theta(u^*)+(u-u^*)^T(-\mathcal{A}^T\lambda^*)]+[(\lambda-\lambda^*)^T(\mathcal{A}u^*-b)]\\ &= \theta(u)-\theta(u^*)+(w-w^*)^TF(w^*)\geq 0, \qquad \forall w^*\in\Omega^*.\end{aligned}$$

上式建立了对偶间隙和变分不等式之间的等价关系. 对于给定的 $\varepsilon>0$, 如果一对 $(u,\lambda)\in\Omega$ 满足

$$\theta(u)-\theta(u^*)+(w-w^*)^TF(w^*)\leq\varepsilon, \quad \forall w^*\in\Omega^*,$$

由于上式左端非负, 这对 $(u,\lambda)\in\Omega$ 就是对偶间隙不大于 ε 的近似解. 因此, 变分不等式也是证明收敛速率的工具. 就像 [26]中证明 ADMM 遍历意义下的 $O(1/t)$ 收敛速率一样, 本文提到的算法都具有这个性质, 证明的程式也一样 [15,38].

凸优化分裂收缩算法的统一框架是以单调变分不等式和邻近点算法为基础. 统一框架算法的收敛性证明用到的知识非常简单, 篇幅也很短. 本文利用统一框架, 对典型的线性约束凸优化问题给出了适合各种需要的具体算法. 遵循统一框架的规则, 基本可以做到随‘需’而‘定’设计算法. 线性代数中的分块矩阵技术, 在算法设计中起了非常重要的作用. 因为工程界比较喜欢用简单一些的方法, 本文主要讲述采用固定步长的校正. 我们的计算实践表明, 如果采用计算步长的校正, 方法会比取固定步长的收敛得快一些.

对分块更多的一些问题, 我们建议采用一种回代型的 ADMM 类算法 [22], 这个方法的修正版本在 [38]中可以查到. 如果要处理不等式约束问题, 我们建议采用 [25]中应用范围更广和便于推广的方法. 最后需要指出的是, 根据“邻近点策略”设计的算法, 有“步步为营”的稳健(收敛性可以得到保证), 有时也会有收敛速度方面的困扰. 因此, 这些算法在编程实现时, 参数的合理选择尤为重要. 我们在 [24] 中提出了以平衡残量为原则的参数调比法则, S. Boyd 在他们的 ADMM 综述文章 [3] 中也介绍了我们的这个法则.

怎样进一步提高统一框架下分裂收缩算法的效率, 作者认为应该分别在预测和校正方面作努力. 在预测方面, [31] 中采用的均困平衡的思想可以借鉴. 确定了预测, 在校正公式 (3.3)(也包括(3.8)) 中如何选取满足收敛性条件的校正矩阵 M. 预测(3.2)只是提供了一个下降方向, 是不是需要跟变尺度拟牛顿方法 [8] 一样, 迭代过程中不断修正矩阵 H 和 M, 也是我们有兴趣进一步考虑的问题.

参考文献

[1] Beck A. First-Order Methods in Optimization. MOS-SIAM Series on Optimization, 2017.

[2] Becker S. The Chen-Teboulle algorithm is the proximal point argorithm, 2019. arXiv:1908.03633 [math.OC].

[3] Boyd S, Parikh N, Chu E, Peleato B and Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends. Mach. Learn., 2010, 3: 1–122.

[4] Cai X J, Gu G Y, He B S and Yuan X M. A proximal point algorithms revisit on the alternating direction method of multipliers. Science China Mathematics, 2013, 56: 2179–2186.

[5] Chambolle A and Pock T. A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision, 2011, 40: 120–145.

[6] Chambolle A and Pock T. On the ergodic convergence rates of a first-order primal-dual algorithm. Math. Program., 2016, 159: 253–287.

[7] Chen C H, He B S, Ye Y Y and Yuan X M. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. Mathematical Programming, 2016, 155: 57–79.

[8] Davidon W C. Variable metric method for minimization, SIAM Journal on Optimization, 1991, 1: 1–17.

[9] Esser E, Möller M, Osher S, Sapiro G and Xin J. A convex model for non-negative matrix factorization and dimensionality reduction on physical space. IEEE Trans. Image Process., 2012, 21: 3239–3252.

[10] Gabay D. Applications of the method of multipliers to variational inequalities. in Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems, Fortin M and Glowinski R, eds., North Holland, Amsterdam, The Netherlands, 1983, 299–331.

[11] Glowinski R. Numerical Methods for Nonlinear Variational Problems. Springer-Verlag, 1984.

[12] Glowinski R and Marrocco A. Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problémes non linéaires. RAIRO Anal. Numer. R2, 41–76 (1975)

[13] Gu G Y, He B S and Yuan X M. Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a unified approach. Comput. Optim. Appl., 2014, 59: 135–161.

[14] He B S. From the projection and contraction methods for variational inequalities to the splitting contraction methods for convex optimization (in Chinese). Numerical Mathematics - A journal of Chinese Universities, 2016, 38: 74–96.

[15] 何炳生. 我和乘子交替方向法20 年. 运筹学学报, 2018, 22: 1–31.

[16] 何炳生. 凸优化和单调变分不等式收缩算法的统一框架. 中国科学：数学, 2018, 48: 255–272.

[17] 何炳生, 何旭初. 一类求解凸规划的鞍点法. 计算数学, 1991, 13: 51–57.

[18] He B S, Liao L Z and Wang X. Proximal-like contraction methods for monotone variational inequalitiesin a unified framework I: Effective quadruplet and primary methods. Comput. Optim. Appl., 2012, 51: 649–679.

[19] He B S, Liao L Z and Wang X. Proximal-like contraction methods for monotone variational inequalities in a unified framework II: General methods and numerical experiments. Comput. Optim. Appl., 2012, 51: 681–708.

[20] He B S, Ma F and Yuan X M. Optimally linearizing the alternating direction method of multipliers for convex programming. Comput. Optim. Appl., 2020, 75: 361–388.

[21] He B S, Liu H, Wang Z R and Yuan X M. A strictly contractive Peaceman-Rachford splitting method for convex programming. SIAM J. Optim., 2014, 24: 1011–1040.

[22] He B S, Tao M and Yuan X M. Alternating direction method with Gaussian back substitution for separable convex programming. SIAM J. Optim., 2012, 22: 313–340.

[23] He B S, Tao M and Yuan X M. A splitting method for separable convex programming. IMA J. Numer. Anal., 2015, 31: 394–426.

[24] He B S, Yang H and Wang S L, Alternating directions method with self-adaptive penalty parameters for monotone variational inequalities, JOTA, 2020, 23: 349–368.

[25] He B S, Xu S J and Yuan X M. Extensions of ADMM for separable convex optimization problems with linear equality or inequality constraints, manusript, 2021. arXiv:2107.01897 [math.OC].

[26] He B S and Yuan X M. On the $O(1/n)$ convergence rate of the alternating direction method. SIAM J. Numer. Anal., 2012, 50: 700–709.

[27] He B S and Yuan X M. On non-ergodic convergence rate of Douglas-Rachford alternating directions method of multipliers. Numer. Math., 2015, 130: 567–577.

[28] He B S and Yuan X M. Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective. SIAM J. Imaging Science, 2012, 5: 119–149.

[29] He B S and Yuan X M. A class of ADMM-based algorithms for three-block separable convex programming. Comput. Optim. Appl., 2018, 70: 791–826.

[30] He B S and Yuan X M. On the optimal proximal parameter of an ADMM-like splitting method for separable convex programming http://www.optimization-online.org/DB_HTML/2017/10/6235.html

[31] He B S and Yuan X M. Balanced augmented Lagrangian method for convex optimization. manusript, 2021. arXiv:2108.08554 [math.OC].

[32] Hestenes M R. Multiplier and gradient methods. J. Optim. Theo. Appl., 1969, 4: 302–320.

[33] Martinet B. Regularisation, d'inéquations variationelles par approximations succesives. Rev. Francaise d'Inform. Recherche Oper., 1970, 4: 154–159.

[34] Nocedal J and Wright S. Numerical Optimization. New York: Springer-Verlag, 1999.

[35] Powell M. A method for nonlinear constraints in minimization problems, in Optimization, Fletcher R, ed., New York: Academic Press, 1969, 283–298.

[36] Rockafellar R T. Monotone operators and the proximal point algorithm. SIAM J. Cont. Optim., 1976, 14: 877–898.

[37] Zhu M and Chan T F. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. CAM Report 08-34, UCLA, 2008.

[38] 何炳生. 凸优化的一阶分裂算法−−变分不等式为工具的统一框架. 见 http://maths.nju.edu. cn/˜hebma 中的《My Talk》的第三个报告.

[39] 何炳生. 凸优化和单调变分不等式的收缩算法. 见http://maths.nju.edu.cn/˜hebma 中的系列讲义.