# 我和乘子交替方向法20年

**何　炳　生**

南方科技大学数学系　　南京大学数学系

E-mail: hebma@nju.edu.cn　Homepage: maths.nju.edu.cn/~hebma

国家自然科学基金(批准号: 11471156)资助项目

**摘要.** 1997 年, 交通网络分析方面的问题把我引进乘子交替方向法 (ADMM) 的研究领域. 近 10 年来, 原本用来求解变分不等式的 ADMM 在优化计算中被广泛采用, 影响越来越大. 本文总结我们 20 年来在 ADMM 方面的工作, 特别是近 10 年 ADMM 在凸优化分裂收缩算法方面的进展. 梳理主要结果, 说清来龙去脉. 文章利用变分不等式的形式研究凸优化的 ADMM 类算法, 论及的所有方法都能纳入一个简单的预测-校正统一框架. 在统一框架下证明算法的收缩性质特别简单. 通读, 有利于了解 ADMM 类算法的概貌. 仔细阅读, 也许就掌握了根据实际问题需要构造分裂算法的基本技巧. 也要清醒地看到, ADMM 类算法源自增广拉格朗日乘子法(ALM) 和邻近点(PPA)算法, 它只是便于利用问题的可分离结构, 并没有消除ALM和PPA等一阶算法固有的缺点.

**关键词.** 凸优化, 单调变分不等式, 乘子交替方向法, 收缩性质, $O(1/t)$ 收敛速率, 统一框架

**MSC (2010) 主题分类.** 90C25, 90C30, 90C33.

## 1　引言

说起我做乘子交替方向法 (ADMM) 研究, 跟人生经历还有一定关系. 首先, 我对优化感兴趣, 源于文化大革命期间"推广优选法小分队"在我家乡县城的一个普及报告. 文革结束恢复高考我上了大学, 华罗庚先生 1980 年在南京大学礼堂关于优选法和数学应用的报告, 以及南京大学何旭初先生在国内优化界的学术地位, 决定了我选择最优化作为专业方向.

听优选法的报告使我懂得, 现实生活中的数学问题里, 函数一般没有显式表达式. 华先生推广的优选法就是求一元单峰函数极值点的只用函数值且少用函数值的方法. 在优化领域, 最初我热心于变分不等式的研究, 是因为它是描述平衡问题的核心工具, 可以用来解释管理科学中的一些问题. 由于受当年华先生推广优选法的影响, 在变分不等式的求解方法中, 我主要对只用函数值的方法感兴趣, 致力于研究少用函数值的方法. 在研究 ADMM 之前, 我的主要工作是求解单调变分不等式的投影收缩算法 [11, 12, 13, 16]. 线性变分不等式的投影收缩算法 [11, 12] 被成功应用到机器人的运动规划和实时控制中 [5, 9, 35, 38]. 由于 [11] 发表于 1994 年, [12] 中描述方法的公式标号是(4) -(7), 这些方法在中山大学的张雨浓课题组2011年以来的20多篇论文以及专著 [39] 中分别称为 94 LVI 和 E47. 利用 [13, 16] 中求解非线性变分不等式投影收缩算法, 中科院武汉岩土力学研究所的学者成功解决一些岩土工程问题 [40, 41]. 工程界能捡到一个能解决他们问题的方法, 都会特别高兴.

交通网络分析 [28] 中的数学问题往往归结为变分不等式. 香港科技大学的杨海教授从我发表的论文 [42] 知道我搞变分不等式算法研究, 邀请我 1997 年去香港科大访问. 阅读了一些交通研究的文献 [28, 29] 以后, 我们开始了ADMM求解变分不等式的研究. 近 10 年来, ADMM 被广泛应用于求解带可分离结构的凸优化问题, 成了热门课题, 也激发了我们进一步的研究兴趣.

本文§2 介绍一般的单调变分不等式和凸优化对应的变分不等式. §3 回顾我们从1997 年开始的第一个 10 年的 ADMM 工作. §4 阐明用 ADMM 求解分离结构凸优化的理由. §5 和 §6 分别介绍最近 10 年 ADMM 类算法在求解两个和多个可分离算子问题上的主要进展. §7 在统一框架

下给出 ADMM 类算法的主要收敛性质的简单证明. §8 验证经典的 ADMM 及其 (§5 和 §6 提到的) 主要进展都可以纳入统一框架. 最后, 我们综述一下这些研究的结论与体会.

# 2 单调变分不等式

优化是最接地气的应用数学, 门槛低. 本文仅要的预备知识是常识性的, 这里不加证明列为引理.

**引理 2.1.** 设 $\mathcal{X} \subset \Re^n$ 是闭凸集, $\theta(x)$ 和 $\varphi(x)$ 都是 $\Re^n \to \Re$ 的凸函数. 如果 $\varphi(x)$ 可微并且 $\min\{\theta(x) + \varphi(x) \,|\, x \in \mathcal{X}\}$ 有解, 那么

$$\tilde{x} \in \arg\min\{\theta(x) + \varphi(x) \,|\, x \in \mathcal{X}\} \tag{2.1a}$$

的充分必要条件是

$$\tilde{x} \in \mathcal{X}, \quad \theta(x) - \theta(\tilde{x}) + (x - \tilde{x})^T \nabla\varphi(\tilde{x}) \geq 0, \quad \forall\, x \in \mathcal{X}. \tag{2.1b}$$

对证明感兴趣的读者也可以参考我主页上第三个报告 [44] 的第一部分.

## 2.1 经典的结构型单调变分不等式

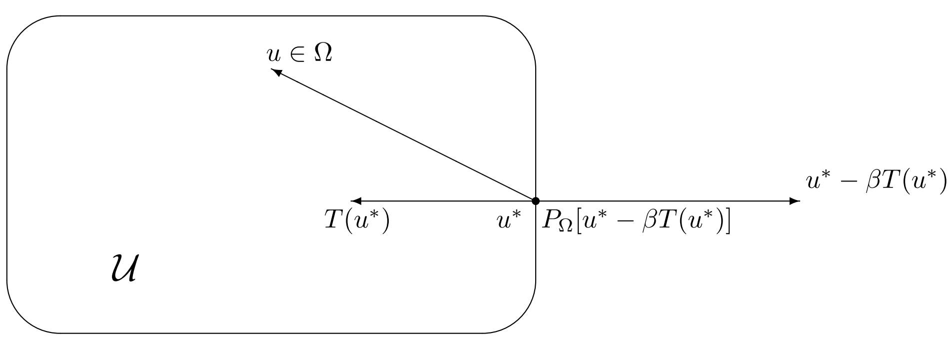
设 $\mathcal{U} \subset \Re^n$ 是一个闭凸集, $T(u)$ 是从 $\Re^n$ 到自身的一个算子, 变分不等式问题是求

$$u^* \in \mathcal{U}, \quad (u - u^*)^T T(u^*) \geq 0, \quad \forall\, u \in \mathcal{U}. \tag{2.2}$$

根据变分不等式和投影方程的关系 (见[14]中的定理1), $u^*$ 是投影方程

$$u = P_{\mathcal{U}}[u - \beta T(u)] \tag{2.3}$$

的解, 其中 $\beta > 0$ 是任意常数. (2.2) 和 (2.3) 的等价性可见下图.




如果对属于 $\mathcal{U}$ (或者 $\Re^n$) 的 $u$ 和 $\tilde{u}$ 都有

$$(u - \tilde{u})^T (T(u) - T(\tilde{u})) \geq 0,$$

就说 $T$ 是 $\mathcal{U}$ (或者 $\Re^n$) 上的单调算子. 在变分不等式 (2.2) 中, 即使 $T(u)$ 可微, 我们也不要求其雅可比(Jacobian)矩阵 $\nabla T(\cdot)$ 对称. 假如 $\nabla T(\cdot)$ 是对称矩阵, 那么 $T(u) \in \Re^n$ 可以看做是一个可微函数 $\varphi(u) : \Re^n \to \Re$ 的梯度, $\nabla T(\cdot) \in \Re^{n \times n}$ 是 $\varphi(\cdot)$ 的海色(Hessian)矩阵. 换句话说, 变分不等式 (2.2) 包含了可微优化问题, 反之则不是.

交通网络分析中遇到的变分不等式

$$u^* \in \mathcal{U}, \quad (u-u^*)^T T(u^*) \ge 0, \quad \forall\, u \in \mathcal{U} \tag{2.4a}$$

中, $u, T(u), \mathcal{U}$ 往往有如下的可分离结构, 其中,

$$u = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad T(u) = \begin{pmatrix} f(x) \\ g(y) \end{pmatrix}, \tag{2.4b}$$

$f(x)$, $g(y)$ 分别是从 $\Re^{n_1}$ 和 $\Re^{n_2}$ 到自身的单调算子;

$$\mathcal{U} = \{(x,y)\,|\, Ax + By = b,\, x \in \mathcal{X},\, y \in \mathcal{Y}\}, \tag{2.4c}$$

其中 $A \in \Re^{m\times n_1}, B \in \Re^{m\times n_2}, b \in \Re^m$, $\mathcal{X} \subset \Re^{n_1}, \mathcal{Y} \subset \Re^{n_2}$ 是简单闭凸集. 对 (2.4c) 中的线性约束 $Ax + By = b$ 引进拉格朗日乘子 $\lambda$, 问题 (2.4) 就转换成

$$w^* \in \Omega, \quad (w-w^*)^T F(w^*) \ge 0, \quad \forall\, w \in \Omega. \tag{2.5a}$$

其中

$$w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \quad u = \begin{pmatrix} x \\ y \end{pmatrix}, \quad F(w) = \begin{pmatrix} f(x) - A^T\lambda \\ g(y) - B^T\lambda \\ Ax + By - b \end{pmatrix}, \tag{2.5b}$$

和 $\Omega = \mathcal{X} \times \mathcal{Y} \times \Re^m$. 由于 $f(x)$, $g(y)$ 是单调的, 这里的 $F(w)$ 也是单调的.

## 2.2 与线性约束凸优化问题等价的变分不等式

设 $\mathcal{U} \subset \Re^n$ 是闭凸集, $\theta(\cdot): \Re^n \to \Re$ 是凸函数, $\mathcal{A} \in \Re^{m\times n}$, $b \in \Re^m$. 我们考虑线性约束的凸优化问题

$$\min\{\theta(u) \mid \mathcal{A}u = b,\ u \in \mathcal{U}\}. \tag{2.6}$$

它的拉格朗日(Lagrange)函数是定义在 $\mathcal{X} \times \Re^m$ 上的

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

如果一对 $(u^*,\lambda^*) \in \mathcal{U} \times \Re^m$ 满足

$$L_{\lambda\in\Re^m}(u^*,\lambda) \le L(u^*,\lambda^*) \le L_{u\in\mathcal{U}}(u,\lambda^*),$$

就称它为 Lagrange 函数的鞍点. 上式写开来就是

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

根据引理 2.1, 鞍点的等价表达式是下面的单调变分不等式:

$$\begin{cases} u^* \in \mathcal{U}, & \theta(u) - \theta(u^*) + (u-u^*)^T(-\mathcal{A}^T\lambda^*) \ge 0, \quad \forall\, u \in \mathcal{U}, \\ \lambda^* \in \Re^m, & (\lambda-\lambda^*)^T(\mathcal{A}u^* - b) \ge 0, \quad \forall\, \lambda \in \Re^m. \end{cases} \tag{2.7}$$

写成紧凑的形式就是

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w-w^*)^T F(w^*) \ge 0, \quad \forall\, w \in \Omega, \tag{2.8a}$$

其中

$$w=\begin{pmatrix} u \\ \lambda \end{pmatrix}, \quad F(w)=\begin{pmatrix} -\mathcal{A}^T\lambda \\ \mathcal{A}u-b \end{pmatrix}, \quad \Omega=\mathcal{U}\times\Re^m. \tag{2.8b}$$

从 (2.8) 到 (2.7), 只要对其中任意的 $w\in\Omega$ 分别取 $w=(u,\lambda^*)$ 和 $w=(u^*,\lambda)$. 变分不等式 (2.5) 和 (2.8) 的解集, 都用 $\Omega^*$ 表示. 在变分不等式框架下研究凸优化的算法, 就像用导函数的方法求二次凸函数的极值, 会带来很大的方便. 这种观点, 正得到越来越多的认可 [3].

# 3 1997- 2007 年间交替方向法方面的主要工作

对交通网络分析中的变分不等式 (2.4) 中的等式约束 $Ax+By-b=0$ 引入 Lagrange 乘子后的变分不等式形式为 (2.5). 显然, (2.5) 又与下面的问题等价:

$$w^*\in\Omega, \quad (w-w^*)^T\mathcal{F}(w^*)\ge 0, \quad \forall\, w\in\Omega. \tag{3.1a}$$

其中

$$w=\begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \quad u=\begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathcal{F}(w)=\begin{pmatrix} f(x)-A^T\lambda+\beta A^T(Ax+By-b) \\ g(y)-B^T\lambda+\beta B^T(Ax+By-b) \\ Ax+By-b \end{pmatrix}, \tag{3.1b}$$

$\beta>0$ 是给定的罚参数.

求解变分不等式 (2.5) 的**乘子交替方向法** (ADMM) [8] 着眼于用松弛分裂的方式求解 (3.1). 它的 $k$ 步迭代从给定的 $v^k=(y^k,z^k)$ 开始, 先求得 $x$-子变分不等式

$$x\in\mathcal{X},\ \ (x'-x)^T\big\{f(x)-A^T\lambda^k+\beta A^T(Ax+By^k-b)\big\}\ge 0,\ \ \forall x'\in\mathcal{X}, \tag{3.2a}$$

的解 $x^{k+1}$. 然后利用已知的 $x^{k+1}$ 和 $\lambda^k$, 求得 $y$-子变分不等式

$$y\in\mathcal{Y},\ \ (y'-y)^T\big\{g(y)-B^T\lambda^k+\beta B^T(Ax^{k+1}+By-b)\big\}\ge 0,\ \ \forall y'\in\mathcal{Y}, \tag{3.2b}$$

的解 $y^{k+1}$. 最后通过

$$\lambda^{k+1}=\lambda^k-\beta(Ax^{k+1}+By^{k+1}-b) \tag{3.2c}$$

更新 Lagrange 乘子. 因为 $k$-步迭代有了 $(y^k,\lambda^k)$ 就可以开始, 我们把 $w=(x,y,\lambda)$ 中的 $(y,\lambda)$ 称为**核心变量**, 而 $x$ 称为**中间变量**. 有的文献 [8] 更新 Lagrange 乘子用

$$\lambda^{k+1}=\lambda^k-\gamma\beta(Ax^{k+1}+By^{k+1}-b),$$

其中 $\gamma\in(0,\frac{1+\sqrt{5}}{2})$. 一般来说, 我们建议取 $\gamma\in[1,1.5)$.

## 3.1 从固定的线性约束罚因子到单调的罚因子序列

乘子交替方向法 (3.2) 的收敛速度对参数 $\beta$ 的选取比较敏感, 过大和过小都会严重影响收敛速度. 或许是受罚函数法的影响, Nagurney 在 [29] 中建议, 用一个单调上升的序列 $\{\beta_k\}$ 代替固定的 $\beta>0$. 我们利用变分不等式残量分析, 对 $\{\beta_k\}$ 单调上升和单调下降(有界)的两种不同情况, 证明了算法的收敛性 [22], 这是我们 1997 年投稿, 1998 年发表的关于交替方向法的第一篇文章.

根据变分不等式和投影方程的关系 (见(2.2) 和 (2.3)), 交替方向法 (3.2) 的 $k$ 步迭代的子问题 (3.2a) 和 (3.2b) 相当于求 $x^{k+1}$ 和 $y^{k+1}$, 它们分别是投影方程

$$x = P_{\mathcal{X}}\{x - \frac{1}{r}[f(x) - A^T\lambda^k + \beta A^T(Ax + By^k - b)]\} \tag{3.3a}$$

和

$$y = P_{\mathcal{Y}}\{y - \frac{1}{s}[g(y) - B^T\lambda^k + \beta B^T(Ax^{k+1} + By - b)]\} \tag{3.3b}$$

的解, 其中 $r > 0, s > 0$ 是任意的常数. 由于等式两边都有未知变量, 我们称之为隐式投影方程.

### 3.2 从固定的线性约束罚因子到自调比的罚因子序列

终究, 我们并不知道初始的 $\beta_0$ 取的过大还是过小. 采用 (3.2) 求解不知 $\beta$ 要往上调还是往下调. 通过对原始和对偶两部分残量的比较, 根据平衡的原理, 我们于2000年发表在 JOTA 的关于交替方向法的文章 [23], 首先给出了迭代过程中自动调整 $\{\beta_k\}$ 的自调比准则. 这个法则被广为采用, 斯坦福大学 Boyd 教授在他们的ADMM综述文章 [1] 中给予介绍, 并在他们的求解器 [10] 中采用了这个法则. 最近10 年, 随着交替方向法的应用越来越广, 这个法则的知名度也越来越高. 从 Google 数据看, 论文 [23] 发表后前 10 年和后 8 年的被引用分别是 20 和 180. 重要原因是交替方向法在大规模稀疏优化中有了越来越受人们关注的应用.

### 3.3 不精确求解子问题的交替方向法

2002 年, 我们在 MP 发了一篇关于交替方向法的文章[17]. 对子问题 (3.2a) 和 (3.2b) 分别加上邻近点项

$$R(x - x^k) \qquad 和 \qquad S(y - y^k),$$

其中 $R$, $S$ 是正定矩阵. 这样, $k$ 步迭代就从给定的 $w^k = (x^k, y^k, \lambda^k)$ 开始, 加上邻近点项的 (3.2a) 和 (3.2b), 求得的 $x^{k+1}$ 和 $y^{k+1}$ 分别是

$$x \in \mathcal{X}, \;\; (x' - x)^T\big\{f(x) - A^T\lambda^k + \beta A^T(Ax + By^k - b) + R(x - x^k)\big\} \ge 0, \;\; \forall\, x' \in \mathcal{X}, \tag{3.4a}$$

和

$$y \in \mathcal{Y}, \;\; (y' - y)^T\big(g(y) - B^T\lambda^k + \beta B^T(Ax^{k+1} + By - b\big) + S(y - y^k) \ge 0, \;\; \forall\, y' \in \mathcal{Y} \tag{3.4b}$$

的解. 由于 $R$, $S$ 正定, 子问题 (3.4a) 和 (3.4b) 就成了强单调变分不等式而可以近似求解. 我们原来的意思是要做成预测-校正方法. 只有采取校正, 子问题的求解精度才可以降低. 然而, 审稿人要求我们不用校正. 因此 (3.4a) 和 (3.4b) 的不精确解 $x^{k+1}$ 和 $y^{k+1}$ 需要满足

$$\|x^{k+1} - \tilde{x}^{k+1}\| \le \nu_k \qquad 和 \qquad \|y^{k+1} - \tilde{y}^{k+1}\| \le \nu_k, \tag{3.5}$$

其中 $\tilde{x}^{k+1}$, $\tilde{y}^{k+1}$ 分别是 (3.4a) 和 (3.4b) 的真解, 非负数列 $\{\nu_k\}$ 要可加 ($\sum_{k=0}^{\infty}\nu_k < +\infty$).

由于 (3.4a) 和 (3.4b) 的真解 $\tilde{x}^{k+1}$, $\tilde{y}^{k+1}$ 是不可能知道的, 又需要根据

$$f(x) + (\beta A^TA + R)x \qquad 和 \qquad g(y) + (\beta B^TB + S)y$$

的强单调因子, 给出 $\|x^{k+1} - \tilde{x}^{k+1}\|$ 的一个上界函数. 具体说来, 假如 $f(x) + (\beta A^TA + R)x$ 的强单调因子是 $r$, 那么对 $\alpha \ge r^{-1}$, 会有

$$\|x^{k+1} - \tilde{x}^{k+1}\|^2 \le 2\alpha\big(E_{[\mathcal{X},\alpha f_k]}(x^{k+1})\big)^T f_k(x^{k+1}) - \|E_{[\mathcal{X},\alpha f_k]}(x^{k+1})\|^2, \tag{3.6}$$

其中

$$E_{[\mathcal{X},\alpha f_k]}(x)=x-P_{\mathcal{X}}[x-\alpha f_k(x)],$$

和

$$f_k(x)=f(x)-A^T\lambda^k+\beta A^T(Ax+By^k-b)+R(x-x^k).$$

换句话说, 在近似求解的过程中, 对得到的 $x^{k+1}$, 要验证

$$2\alpha\big(E_{[\mathcal{X},\alpha f_k]}(x^{k+1})\big)^T f_k(x^{k+1})-\|E_{[\mathcal{X},\alpha f_k]}(x^{k+1})\|^2\le\nu_k^2$$

是否满足, 这显然是相当麻烦的. 此外, $\sum_{k=0}^{\infty}\nu_k<+\infty$ 在计算中也缺乏指导意义. 管理科学中的问题, $f(x)$ 和 $g(y)$ 也没有解析表达式, 文章中也没有按照这个理论上可行的办法给出算例.

### 3.4 容易实行的交替方向法

预测-校正是数值计算的一种基本方法. 求解变分不等式 (2.2) 的投影收缩算法 [11, 12, 13] 是一些典型的预测校正方法. 在 [17] 中没有实现的预测-校正意图, 在 [18] 中得到了实现.

假如 (3.4) 中的 $R=rI$, $S=sI$, $x^{k+1}$ 和$y^{k+1}$ 就变成问题

$$x\in\mathcal{X},\ (x'-x)^{\rm T}\big\{f(x)-A^{\rm T}\lambda^k+\beta A^{\rm T}(Ax+By^k-b)+r(x-x^k)\big\}\ge 0,\ \forall\, x'\in\mathcal{X},$$

和

$$y\in\mathcal{Y},\ (y'-y)^{\rm T}\big\{g(y)-B^{\rm T}\lambda^k+\beta B^{\rm T}(Ax^{k+1}+By-b\big)+s(y-y^k)\}\ge 0,\ \forall\, y'\in\mathcal{Y}$$

的解. 再根据变分不等式和投影方程的关系(见前面的(2.2)和(2.3)), 就得到

$$x^{k+1}=P_{\mathcal{X}}\big\{x^k-\frac{1}{r}[f(x^{k+1})-A^T\lambda^k+\beta A^T(Ax^{k+1}+By^k-b)]\big\},\tag{3.7a}$$

和

$$y^{k+1}=P_{\mathcal{Y}}\big\{y^k-\frac{1}{s}[g(y^{k+1})-B^T\lambda^k+\beta B^T(Ax^{k+1}+By^{k+1}-b\big)]\}\tag{3.7b}$$

由于 (3.7a) 的左右两端都有要求的 $x^{k+1}$ ((3.7b) 的左右两端都有要求的 $y^{k+1}$), 求解这些子问题本身是有难度的. 随随便便把 (3.7) 右端尚未知晓的 $x^{k+1}$ 和 $y^{k+1}$ 分别用 $x^k$ 和 $y^k$ 去代替, 实现起来就非常容易, 但是保证收敛就有问题. 我们以这种想法来建立预测-校正的乘子交替方向法. 对给定的 $w^k=(x^k,y^k,\lambda^k)$, 把通过

$$\tilde{x}^k=P_{\mathcal{X}}\big\{x^k-\frac{1}{r}[f(x^k)-A^T\lambda^k+\beta A^T(Ax^k+By^k-b)]\big\},\tag{3.8a}$$

$$\tilde{y}^k=P_{\mathcal{Y}}\big\{y^k-\frac{1}{s}[g(y^k)-B^T\lambda^k+\beta B^T(A\tilde{x}^k+By^k-b\big)]\}\tag{3.8b}$$

和

$$\tilde{\lambda}^k=\lambda^k-\beta(A\tilde{x}^k+B\tilde{y}^k-b).\tag{3.8c}$$

得到的点 $\tilde{w}^k=(\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)$ 当成预测点, 其中的 $r,s>0$ 是适当选取常数, 使得

$$\|\xi_x^k\|\le\nu r\|x^k-\tilde{x}^k\|,\qquad \xi_x^k=f(x^k)-f(\tilde{x}^k)+\beta A^TA(x^k-\tilde{x}^k),\tag{3.9a}$$

和

$$\|\xi_y^k\|\le\nu s\|y^k-\tilde{y}^k\|,\qquad \xi_y^k=g(y^k)-g(\tilde{y}^k)+\beta B^TB(y^k-\tilde{y}^k),\tag{3.9b}$$

能够成立, 这里 $\nu = 0.9 \in (0,1)$. 注意到 (3.8a) 和 (3.8b) 是已知向量在简单集合上的投影, (3.9) 中用的是相对误差, 当 $f(x)$ 和 $g(y)$ Lipschitz 连续时, 这是可以办到的.

得到预测点以后, 再用校正公式

$$\textbf{(校正公式-I)} \qquad w^{k+1} = w^k - \alpha_k d(w^k, \tilde{w}^k, \xi^k) \tag{3.10}$$

产生新的迭代点. 这里的方向由

$$d(w^k, \tilde{w}^k, \xi^k) = (w^k - \tilde{w}^k) - G_k^{-1}\xi^k,$$

给出, 其中

$$G_k = \begin{pmatrix} rI & 0 & 0 \\ 0 & sI + \beta B^T B & 0 \\ 0 & 0 & \frac{1}{\beta}I \end{pmatrix}, \qquad \xi^k = \begin{pmatrix} \xi_x^k \\ \xi_y^k \\ 0 \end{pmatrix}.$$

步长则通过

$$\alpha_k = \gamma \cdot \frac{(\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k) + (w^k - \tilde{w}^k)^T G_k d(w^k, \tilde{w}^k, \xi^k)}{\|d(w^k, \tilde{w}^k, \xi^k)\|_{G_k}^2}, \quad \gamma \in (0,2),$$

给出. 特别地, 当 $B$ 也是单位矩阵时, 这时 $G$ 是分块数量矩阵, 也可以用校正公式

$$\textbf{(校正公式-II)} \qquad w^{k+1} = P_\Omega\{w^k - \alpha_k G_k^{-1} q(w^k, \tilde{w}^k)\} \tag{3.11}$$

产生新的迭代点 $w^{k+1}$. 校正公式-II 的步长和校正公式-I 相同, 方向

$$q(w^k, \tilde{w}^k) = F(\tilde{w}^k) + \beta(A, B, 0)^T B(y^k - \tilde{y}^k).$$

可以这样做的原因是当 $G$ 是分块数量矩阵, $\Omega = \mathcal{X} \times \mathcal{Y} \times \Re^m$ 时,

$$\begin{aligned} & \text{Argmin}\{\frac{1}{2}\|w - [w^k - \alpha_k G_k^{-1} q(w^k, \tilde{w}^k)]\|_{G_k}^2 \mid w \in \Omega\} \\ = \;& \text{Argmin}\{\frac{1}{2}\|w - [w^k - \alpha_k G_k^{-1} q(w^k, \tilde{w}^k)]\|^2 \mid w \in \Omega\} \\ = \;& P_\Omega\{w^k - \alpha_k G_k^{-1} q(w^k, \tilde{w}^k)\}. \end{aligned}$$

这个方法不要求 $f(x)$ 和 $g(y)$ 有解析表达式, 只要求对给定的 $x^k$ 和 $y^k$, 得到相应的 $f(x^k)$ 和 $g(y^k)$. [18] 中以一些经典的交通网络分析的问题为例, 用上面的方法做了数值计算. 为完成这篇论文, 我特地邀请当年我们计算数学考研第一名的大学同学钱迈健合作.

变分不等式 (2.4) 用来解释管理科学中的问题, 文献中总是人为地构造一些算例. 虽然凸优化问题也可以转换成一个结构型变分不等式(2.5), 但很少会想用求解变分不等式 (一大类难问题) 的方法去求解(其中一类相对容易的) 优化问题. 因此, 我们前 10 年的 ADMM 研究, 相对后 10 年而言, 开了个头, 但动力不足, 有些低迷. 2000 年 JOTA 论文 [23] 中的自调比准则, 和 [18] 中只用函数值的求解方法, 成了我们这阶段 ADMM 研究的主要贡献.

## 4 用乘子交替方向法求解凸优化问题

激起我们进一步研究 ADMM 的是信息科学中的一些结构型优化问题[36, 37]. 先从一般问题 (2.6) 谈起, 求解这个问题的两类经典方法是二次罚函数方法(Quadratic Penalty Method) 和增广拉格朗日乘子法 (Augmented Lagrangian Method) [26, 31].

- 求解问题 (2.6) 的**二次罚函数方法(QPM)**的 $k$-步迭代是求解子问题

$$u^{k+1} = \mathrm{Argmin}\big\{\theta(u) + \tfrac{\beta_k}{2}\|\mathcal{A}u - b\|^2 \big| u \in \mathcal{U}\big\}, \tag{4.1}$$

  其中 $\{\beta_k\}$ 是给定的单调上升趋向 $+\infty$ 的数列. 求解子问题 (4.1) 一般以 $u^k$ 为初始点.

- 求解问题 (2.6) 的**增广拉格朗日乘子法(ALM)**的 $k$-步迭代是从给定的 $\lambda^k$ 开始, 通过

$$\left\{ \begin{array}{rcl} u^{k+1} &=& \mathrm{Argmin}\left\{\theta(u) - (\lambda^k)^T(\mathcal{A}u - b) + \frac{\beta}{2}\|\mathcal{A}u - b\|^2 | u \in \mathcal{U}\right\}, \\ \lambda^{k+1} &=& \lambda^k - \beta(\mathcal{A}u^{k+1} - b). \end{array} \right. \tag{4.2}$$

  求得新的迭代点 $w^{k+1} = (u^{k+1}, \lambda^{k+1})$, 其中 $\beta > 0$ 是给定的常数.

注意到, 忽略 (4.2) 的 $u$-子问题目标函数中的常数项, 有

$$\begin{array}{rl} & \mathrm{Argmin}\big\{\theta(u) - (\lambda^k)^T(\mathcal{A}u - b) + \frac{\beta}{2}\|\mathcal{A}u - b\|^2 \big| u \in \mathcal{U}\big\} \\ = & \mathrm{Argmin}\big\{\theta(u) + \frac{\beta}{2}\|\mathcal{A}u - (b + \dfrac{1}{\beta}\lambda^k)\|^2 \big| u \in \mathcal{U}\big\}. \end{array}$$

求解问题 (2.6), 增广拉格朗日乘子法 (4.2) 和二次罚函数法 (4.1) 需要求解的子问题难度完全一样. Nocedal 和 Wright 的专著 [30] 的第十七章说的很清楚, 增广拉格朗日乘子法远优于二次罚函数法. 通常, 我们把自变量 $u$ 和对偶变量 $\lambda$ 看做对弈的双方, 二次罚函数方法只考虑了自变量一方, 增广拉格朗日乘子法则同时顾及了对偶方的感受.

图像重构中的一些问题 [34] 可以归结为一个有两个可分离算子的结构型凸优化问题

$$\min\{\theta_1(x) + \theta_2(y) \mid Ax + By = b,\ x \in \mathcal{X}, y \in \mathcal{Y}\}. \tag{4.3}$$

这相当于在 (2.6) 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$A \in \Re^{m\times n_1}$, $B \in \Re^{m\times n_2}$.

§2.2 的分析告诉我们, 求解 (4.3) 相当于求解变分不等式

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \ge 0, \quad \forall\, w \in \Omega, \tag{4.4a}$$

其中

$$w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \quad u = \begin{pmatrix} x \\ y \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A^T\lambda \\ -B^T\lambda \\ Ax + By - b \end{pmatrix}, \tag{4.4b}$$

和

$$\theta(u) = \theta_1(x) + \theta_2(y), \qquad\qquad \Omega = \mathcal{X} \times \mathcal{Y} \times \Re^m. \tag{4.4c}$$

问题 (4.3) 的增广 Lagrange 函数是

$$\mathcal{L}_\beta(x, y, \lambda) = \theta_1(x) + \theta_2(y) - \lambda^T(Ax + By - b) + \frac{\beta}{2}\|Ax + By - b\|^2. \tag{4.5}$$

对问题 (4.3), 人们首先考虑到一些经典的方法. **二次罚函数方法和增广拉格朗日乘子法**.

- 根据 (4.1), 求解问题 (4.3) 的二次罚函数方法的 $k$-步迭代是

$$(x^{k+1}, y^{k+1}) = \mathrm{Argmin}\big\{\theta_1(x) + \theta_2(y) + \tfrac{\beta_k}{2}\|Ax + By - b\|^2 \big| x \in \mathcal{X},\ y \in \mathcal{Y}\big\}. \tag{4.6}$$

- 根据 (4.2), 求解问题 (4.3) 的增广 Lagrange 乘子法的 $k$-步迭代是从给定的 $\lambda^k$ 开始, 求得

$$
\left\{\begin{array}{rcl}
(x^{k+1}, y^{k+1}) & = & \mathrm{Argmin}\left\{\begin{array}{c} \theta_1(x)+\theta_2(y)-(\lambda^k)^T(Ax+By-b) \\ +\frac{\beta}{2}\|Ax+By-b\|^2 \end{array}\middle| \begin{array}{c} x\in\mathcal{X} \\ y\in\mathcal{Y}\end{array}\right\}, \\
\lambda^{k+1} & = & \lambda^k-\beta(Ax^{k+1}+By^{k+1}-b).
\end{array}\right. \tag{4.7}
$$

利用增广 Lagrange 函数的表达式 (4.5), (4.7) 中的 $(x,y)$-子问题可以写成

$$
(x^{k+1}, y^{k+1}) = \mathrm{Argmin}\big\{\mathcal{L}_\beta(x,y,\lambda^k)\,|\,x\in\mathcal{X}, y\in\mathcal{Y}\big\}.
$$

这两种处理方式, (4.7) 优于 (4.6). 它们共同的缺点是没有利用问题的可分离结构, 求解这些子问题有时会无从下手.

**松弛了的二次罚函数方法和增广拉格朗日乘子法**. 针对 (4.6) 和 (4.7) 中 $(x,y)$-子问题难解的情况, 考虑将 $(x,y)$ 子问题通过松弛分开来做, 分别得到下面的**交替极小化方法和乘子交替方向法**.

- 把求解问题 (4.3) 的二次罚函数法 (4.6) 中的 $\beta_k\equiv\beta>0$ 固定，并将一起求解的 $(x,y)$ 子问题分开求解, 就得到所谓的**交替极小化方法 (AMA)**. 它的 $k$-步迭代是从给定的 $y^k$ 开始, 通过

$$
\left\{\begin{array}{rcl}
x^{k+1} & = & \mathrm{Argmin}\big\{\theta_1(x)+\frac{\beta}{2}\|Ax+By^k-b\|^2\big|x\in\mathcal{X}\big\}, \\
y^{k+1} & = & \mathrm{Argmin}\big\{\theta_2(y)+\frac{\beta}{2}\|Ax^{k+1}+By-b\|^2\big|y\in\mathcal{Y}\big\},
\end{array}\right. \tag{4.8}
$$

  得到 $(x^{k+1}, y^{k+1})$. 换句话说, 交替极小化方法实际上是处理可分离结构型优化问题 (4.3) 的松弛了的二次罚函数方法.

- 把求解问题 (4.3) 的 ALM (4.7) 中一起求解的 $(x,y)$ 子问题分开求解, 就得到所谓的**乘子交替方向法 (ADMM)**. 它的 $k$-步迭代是从给定的 $(y^k,\lambda^k)$ 开始, 通过

$$
\left\{\begin{array}{rcl}
x^{k+1} & = & \mathrm{Argmin}\big\{\theta_1(x)-(\lambda^k)^TAx+\frac{\beta}{2}\|Ax+By^k-b\|^2\big|x\in\mathcal{X}\big\}, \\
y^{k+1} & = & \mathrm{Argmin}\big\{\theta_2(y)-(\lambda^k)^TBy+\frac{\beta}{2}\|Ax^{k+1}+By-b\|^2\big|y\in\mathcal{Y}\big\}, \\
\lambda^{k+1} & = & \lambda^k-\beta(Ax^{k+1}+By^{k+1}-b).
\end{array}\right. \tag{4.9}
$$

  得到 $(x^{k+1}, y^{k+1}, \lambda^{k+1})$, 完成一次迭代. 换句话说, 乘子交替方向法实际上是处理可分离结构型优化问题 (4.3) 的松弛了的增广拉格朗日乘子法.

增广拉格朗日乘子法远优于罚函数方法. 对它们分别松弛以后, 乘子交替方向法 (ADMM) 也应该优于交替极小化方法 (AMA). 基于这种考虑, 2009 年 10 月上海大学主办的华东地区运筹学与控制论博士论坛安排我做大会报告的时候, 我就报告“信息技术中的凸优化问题和交替方向法求解”, 呼吁青年学者注意交替方向法. 第二年, 网上就有了 Boyd 他们受到热捧的关于乘子交替方向法的文章 [1].

# 5 近 10 年在乘子交替方向法方面的主要工作

科学计算中有大量的结构型优化问题 (4.3), ADMM 是求解这些问题的好方法[1, 36, 37]. 凭借前期的研究基础, 近 10 年来, 我们主要致力于ADMM类方法研究. 对求解两个算子的 ADMM, 我们在方法改进和收敛速率方面做了些工作. 这方面的主要结果, 证明都可以用本文 §7 的统一框架, 在 §8.1 中很容易验证. 为了行文方便, 当 $\mathcal{A}=[A,B]$ 或者$\mathcal{A}=[A,B,C]$ 的时候, 我们都假设 $B$ 和 $C$ 列满秩.

## 5.1 ADMM 方法上的改进

两个算子问题的 ADMM 方法改进, 主要是提出了"PPA 意义下可以延拓的乘子交替方向法"和"对称更新乘子的交替方向法".

### 5.1.1 PPA 意义下延拓的乘子交替方向法

结构型可分离算子的凸优化都是问题 (2.6) 的特例. 要说求解 (4.3) 的延拓的乘子交替方向法就必须从凸优化问题 (2.6) 的增广拉格朗日乘子法说起. 凸优化问题 (2.6) 的增广拉格朗日函数是

$$\mathcal{L}_\beta(u,\lambda)=\theta(u)-\lambda^T(\mathcal{A}u-b)+\frac{\beta}{2}\|\mathcal{A}u-b\|^2.$$

求解 (2.6) 的增广拉格朗日乘子法的 $k$-步迭代, 是从给定的 $\lambda^k$ 开始, 通过

$$\left\{\begin{array}{lcl} u^{k+1} & = & \mathrm{Argmin}\{\mathcal{L}_\beta(u,\lambda^k)\mid u\in\mathcal{U}\} \\ \lambda^{k+1} & = & \lambda^k-\beta(\mathcal{A}u^{k+1}-b). \end{array}\right. \tag{5.1}$$

增广拉格朗日乘子法收敛的主要性质是

$$\|\lambda^{k+1}-\lambda^*\|^2\le\|\lambda^k-\lambda^*\|^2-\|\lambda^k-\lambda^{k+1}\|^2. \tag{5.2}$$

事实上, 增广拉格朗日乘子法 (5.1) 中求解 $u$-子问题也是要花代价的. 如果将 $\lambda$ 进一步延拓, 取

$$\lambda^{k+1}:=\lambda^k-\gamma(\lambda^k-\lambda^{k+1}),\quad \gamma=1.5\in(0,2)$$

往往会提高收敛速度, 上式右端的 $\lambda^{k+1}$ 是由 (5.1) 提供的.

既然乘子交替方向法(ADMM)是从增广拉格朗日乘子法(ALM)松弛而来的, ALM 可以延拓的性质ADMM能不能也有? 答案是经过适当改造也可以有 [2]. 具体做法是 $k$-步迭代从给定的 $(y^k,\lambda^k)$ 开始, 将经典的 ADMM (4.9) 中求解 $y$-子问题和校正 $\lambda$ 的顺序交换, 通过

$$\left\{\begin{array}{lcl} x^{k+1} & = & \mathrm{Argmin}\{\mathcal{L}_\beta(x,y^k,\lambda^k)\,|\,x\in\mathcal{X}\}, \\ \lambda^{k+1} & = & \lambda^k-\beta(Ax^{k+1}+By^k-b), \\ y^{k+1} & = & \mathrm{Argmin}\{\mathcal{L}_\beta(x^{k+1},y,\lambda^{k+1})\,|\,y\in\mathcal{Y}\}, \end{array}\right. \tag{5.3a}$$

得到的 $w^{k+1}=(x^{k+1},\,y^{k+1},\,\lambda^{k+1})$ 作为预测点, 然后

$$\left\{\begin{array}{lcl} y^{k+1} & := & y^k-\gamma(y^k-y^{k+1}), \\ \lambda^{k+1} & := & \lambda^k-\gamma(\lambda^k-\lambda^{k+1}). \end{array}\right. \quad \text{（松弛延拓）} \tag{5.3b}$$

这里 $\gamma\in(0,2)$. 赋值号 ":=" 表示 (5.3b) 右端的 $(y^{k+1},\lambda^{k+1})$ 是由算法的前半部分 (5.3a) 产生的. (5.3b) 左端才是下一步迭代开始所需要的 $(y^{k+1},\lambda^{k+1})$. 对多数问题, 这样往往能加快收敛.

### 5.1.2 对称更新乘子的交替方向法

求解 (4.3) 的经典的乘子交替方向法是 (4.9). 从问题 (4.3) 本身看, 原始变量 $x$ 和 $y$ 是平等的, 在算法设计上平等对待子 $x$ 和 $y$ 子问题, 也是最自然不过的考虑. 因此我们采用对称的交替方向法 [19]. 它的 $k$-步迭代从给定的 $(y^k,\lambda^k)$ 开始, 通过

$$\left\{\begin{array}{l} x^{k+1}=\mathrm{Argmin}\{\mathcal{L}_\beta(x,y^k,\lambda^k)\,|\,x\in\mathcal{X}\}, \\ \lambda^{k+\frac{1}{2}}=\lambda^k-\mu\beta(Ax^{k+1}+By^k-b), \\ y^{k+1}=\mathrm{Argmin}\{\mathcal{L}_\beta(x^{k+1},y,\lambda^{k+\frac{1}{2}})\,|\,y\in\mathcal{Y}\}, \\ \lambda^{k+1}=\lambda^{k+\frac{1}{2}}-\mu\beta(Ax^{k+1}+By^{k+1}-b). \end{array}\right. \tag{5.4}$$

得到新的迭代点 $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1})$, 其中 $\mu \in (0,1)$ (我们通常取$\mu = 0.9$). 当 $\mu = 1$ 时, 可以举出不收敛的反例.

## 5.2 经典乘子交替方向法的收敛速率

理论方面, 对经典的乘子交替方向法, 在遍历意义下和点列意义下证明了 $O(1/t)$ 的收敛速率.

### 5.2.1 遍历意义下的 $O(1/t)$ 的迭代复杂性

为了证明算法的迭代复杂性, 我们需要对变分不等式 (2.8) 的解集做新的刻画. 由于对 (2.8) 中的仿射算子 $F$ 恰有

$$(w - w^*)^T F(w^*) = (w - w^*)^T F(w),$$

变分不等式问题

$$w^* \in \Omega, \;\; \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \ge 0, \;\; \forall\, w \in \Omega,$$

和

$$w^* \in \Omega, \;\; \theta(u) - \theta(u^*) + (w - w^*)^T F(w) \ge 0, \;\; \forall\, w \in \Omega,$$

是等价的. 我们用后者定义变分不等式 (2.8) 的近似解. 对给定的 $\epsilon > 0$, 如果 $\tilde{w} \in \Omega$ 满足

$$\tilde{w} \in \Omega, \;\; \theta(u) - \theta(\tilde{u}) + (w - \tilde{w})^T F(w) \ge -\epsilon, \;\; \forall\; w \in \mathcal{D}_{(\tilde{w})}, \tag{5.5a}$$

其中

$$\mathcal{D}_{(\tilde{w})} = \{w \in \Omega \,|\, \|w - \tilde{w}\| \le 1\}, \tag{5.5b}$$

就叫做变分不等式 (2.8) 的 $\epsilon$ 近似解. 它可以等价地表示成

$$\tilde{w} \in \Omega, \quad \sup_{w \in \mathcal{D}_{(\tilde{w})}} \big\{\theta(\tilde{u}) - \theta(u) + (\tilde{w} - w)^T F(w)\big\} \le \epsilon. \tag{5.6}$$

人们感兴趣的是：对给定的 $\epsilon > 0$, 经过多少次迭代, 能够得到一个 $\tilde{w} \in \Omega$, 使得 (5.6) 成立.

对求解 (4.3) 的经典的乘子交替方向法 (4.9), 假设迭代从给定的 $v^0 = (y^0, \lambda^0)$ 开始, 生成迭代序列 $\{w^k\} = \{(x^k, y^k, \lambda^k)\}$. 我们通过

$$\tilde{x}^k = x^{k+1}, \qquad \tilde{y}^k = y^{k+1} \qquad \text{和} \qquad \tilde{\lambda}^k = \lambda^k - \beta(Ax^{k+1} + By^k - b),$$

定义了辅助序列 $\{\tilde{w}^k\}$, 在 [24] 中证明了 $O(1/t)$ 收敛速率需要的关键不等式

$$\theta(\tilde{u}_t) - \theta(u) + (\tilde{w}_t - w)^T F(w) \le \frac{1}{2(t+1)} \|v - v^0\|_H^2, \;\; \forall w \in \Omega,$$

其中 $t$ 为任意正整数, $\tilde{w}_t$ 是 $\tilde{w}^0, \tilde{w}^1, \ldots, \tilde{w}^t$ 的算术平均, 也即

$$\tilde{w}_t = \frac{1}{t+1} \sum_{k=0}^{t} \tilde{w}^k, \quad \text{矩阵} \quad H = \begin{pmatrix} \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I \end{pmatrix}.$$

2012 年初发表的这篇证明收敛速率的文章[24], 篇幅不长. 由于 ADMM 广受关注, 该文这几年常在 SIAM 数值分析的热点论文中排名前五. 中国科学院文献情报中心科学计量团队在《科学观察》2017年第12卷第6期上公布的统计数据, 该文在中国学者于2012–2016 发表的数学论文中被引频次最高, 在热点论文中排名第一.

### 5.2.2 点列意义下收敛速率方面的结果

求解问题 (4.3) 的经典的 ADMM (4.9) 是从增广拉格朗日乘子法松弛来的. 增广拉格朗日乘子法 (5.1) 的 $k$ 步迭代从 $\lambda^k$ 开始, 产生的迭代序列 $\{\lambda^k\}$ 具有收缩性质 (5.2). ADMM (4.9) 的 $k$-步迭代从 $v^k=(y^k,\lambda^k)$ 开始, 也有相应的性质

$$\|v^{k+1}-v^*\|_H^2 \le \|v^k-v^*\|_H^2 - \|v^k-v^{k+1}\|_H^2, \quad \text{其中} \quad H=\begin{pmatrix} \beta B^TB & 0 \\ 0 & \frac{1}{\beta}I \end{pmatrix}. \tag{5.7}$$

这个性质是 [22] 中定理 1 的特例. 我的主页上系列讲义 [45] 的第 11 讲给出了更简单的证明.

求解 (2.6) 的增广拉格朗日乘子法 (5.1), 我们可以把 $\|\lambda^k-\lambda^{k+1}\|$ 的大小看做误差的度量. 增广拉格朗日乘子除了有收缩性质 (5.2) 以外, 还有残量序列 $\{\|\lambda^k-\lambda^{k+1}\|^2\}$ 单调不增的性质, 也就是

$$\|\lambda^k-\lambda^{k+1}\|^2 \le \|\lambda^{k-1}-\lambda^k\|^2.$$

ADMM 能否有类似的性质? 我们在 [25] 中证明了

$$\|v^k-v^{k+1}\|_H^2 \le \|v^{k-1}-v^k\|_H^2. \tag{5.8}$$

这是 ADMM 有点列意义下的收敛速率的关键不等式. 由 (5.7) 和 (5.8)

$$\|v^t-v^{t+1}\|_H^2 \le \frac{1}{t+1}\Big(\sum_{k=0}^{t}\|v^k-v^{k+1}\|_H^2\Big) \le \frac{1}{t+1}\Big(\sum_{k=0}^{\infty}\|v^k-v^{k+1}\|_H^2\Big) \le \frac{1}{t+1}\|v^0-v^*\|_H^2.$$

这些性质, 使用本文 §7 的统一框架, 证明更加简单.

## 6 多个可分离算子的凸优化问题

我们以三个可分离算子的凸优化问题

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

为例. 这相当于在 (2.6) 中, 置 $n=n_1+n_2+n_3$, $\mathcal{X}\subset\Re^{n_1}$, $\mathcal{Y}\subset\Re^{n_2}$ $\mathcal{Z}\subset\Re^{n_3}$, $\mathcal{U}=\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}$. $\theta(u)=\theta_1(x)+\theta_2(y)+\theta_3(z)$, $\theta_1(x):\Re^{n_1}\to\Re$, $\theta_2(y):\Re^{n_2}\to\Re$, $\theta_3(z):\Re^{n_3}\to\Re$. 矩阵 $\mathcal{A}=(A,B,C)$, 其中$A\in\Re^{m\times n_1}$, $B\in\Re^{m\times n_2}$ $C\in\Re^{m\times n_3}$. 这个问题的拉格朗日函数是

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

问题 (6.1) 同样可以归结为变分不等式问题

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

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(与两个算子的符号有区别)

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

对三个可分离算子的凸优化问题, 采用直接推广的乘子交替方向法, $k$ 步迭代是从给定的 $v^k=(y^k,z^k,\lambda^k)$ 出发, 通过

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

求得新的迭代点 $w^{k+1}=(x^{k+1},y^{k+1},z^{k+1},\lambda^{k+1})$. 当矩阵 $A$, $B$, $C$ 中有两个是互相正交的时候, 方法 (6.1) 是收敛的 [4], 因为这时的三个是假的, 实际上相当于两个算子的问题.

## 6.1 三个算子不收敛的例子和值得研究的重要问题

直接推广的乘子交替方向法对三个以上算子的问题, 计算效果也相当不错, 但是至今也没有给出收敛性证明. 如果要举个不收敛的例子, 那肯定从最简单的线性方程组着手. 对 (6.1), 我们取 $\theta_1(x)=\theta_2(y)=\theta_3(z)=0$, $\mathcal{X}=\mathcal{Y}=\mathcal{Z}=\Re$,

$$\mathcal{A}=[A,B,C]\in\Re^{3\times3}\quad \text{是个非奇异矩阵,}\quad b=0\in\Re^3.$$

问用直接推广的 ADMM (6.3) 迭代求解线性方程组 $\mathcal{A}u=0$ 是否一定收敛. 我们用线性方程组

$$\begin{pmatrix}1&1&1\\1&1&2\\1&2&2\end{pmatrix}\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}0\\0\\0\end{pmatrix},$$

还有一些据此延伸的例子, 证明了直接推广的 ADMM 并不收敛. 然而, 这些例子更多的是在理论方面有意义, 因为实际问题中 $\mathcal{A}=[A,B,C]$ 并不是这种形式.

**值得继续研究的问题和猜想.** 三个算子的实际问题中, 线性约束矩阵

$$\mathcal{A}=[A,B,C]\quad \text{中, 往往至少有一个是单位矩阵. 即,}\quad \mathcal{A}=[A,B,I].$$

直接推广的 ADMM 处理这种更贴近实际的三个算子的问题, 既没有证明收敛, 也没有举出反例, 至今我们于心不甘! 举个简单的例子来说吧:

- 乘子交替方向法 (4.9) 处理问题

$$\min\{\theta_1(x)+\theta_2(y)|Ax+By=b,\ x\in\mathcal{X},y\in\mathcal{Y}\}\quad \text{是收敛的.}$$

- 将等式约束换成不等式约束, 问题就变成

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

- 再化成三个算子的等式约束问题就是

$$\min\{\theta_1(x)+\theta_2(y)+0\,|\,Ax+By+z=b,\ x\in\mathcal{X},y\in\mathcal{Y},z\ge0\}.$$

- 直接推广的乘子交替方向法 (6.3) 处理上面这种问题, 我们猜想是收敛的, 但是至今没有证明收敛性. 仍然是一个遗留个我们的挑战性问题 !

### 6.2 处理三个算子问题的 ADMM 类方法

在对直接推广的 ADMM (6.3) 证明不了收敛性的时候, 我们就着手对三个算子的问题提出一些修正算法. 修正方法的原则是尽量少做改动, 保持 ADMM 好品性. 特别是对问题不加任何额外条件, 对经典 ADMM 中需要调比选取的 $\beta$, 不做任何限制. 一句话, 只对方法本身动手术！

#### 6.2.1 部分平行分裂 ALM 的预测校正方法

为了更接近 (6.3), 我们还是把 $x$ 当成中间变量, 迭代从 $v^k=(y^k,z^k,\lambda^k)$ 到 $v^{k+1}=(y^{k+1},z^{k+1},\lambda^{k+1})$. 平行处理 $y$, $z$-子问题再更新 $\lambda$. 采用

$$\left\{\begin{array}{rcl} x^{k+1} & = & \arg\min\left\{\mathcal{L}_\beta^3(x,y^k,z^k,\lambda^k)\;\middle|\;x\in\mathcal{X}\right\},\\ y^{k+1} & = & \arg\min\left\{\mathcal{L}_\beta^3(x^{k+1},y,z^k,\lambda^k)\;\middle|\;y\in\mathcal{Y}\right\},\\ z^{k+1} & = & \arg\min\left\{\mathcal{L}_\beta^3(x^{k+1},y^k,z,\lambda^k)\;\middle|\;z\in\mathcal{Z}\right\},\\ \lambda^{k+1} & = & \lambda^k-\beta(Ax^{k+1}+By^{k+1}+Cz^{k+1}-b) \end{array}\right. \tag{6.4}$$

生成的点 $(x^{k+1},y^{k+1},z^{k+1},\lambda^{k+1})$ 当成预测点. $y$, $z$ 子问题平行了, 太自由, 包括据此更新的 $\lambda$, 都需要校正. 校正公式是

$$v^{k+1}:=v^k-\alpha(v^k-v^{k+1}),\quad \alpha\in(0,2-\sqrt{2}). \tag{6.5}$$

譬如说, 我们可以取 $\alpha=0.55$. 注意到 (6.5) 右端 $v^{k+1}=(y^{k+1},z^{k+1},\lambda^{k+1})$ 是由(6.4) 提供的. 换句话说, 这里的校正就是把走的太“远”的 $v^{k+1}$ 往回拉一点. 预测-校正, 是我们从投影收缩算法开始的算法框架. 预测只是提供收缩方向, 校正时计算步长, 可以确保 $\|v^k-v^*\|_H$ 下降收缩. 文章 [15] 2009 年发表在 Computational Optimization and Applications. 是 2007 年投的稿. 因此, 也可以看做我和 ADMM 20 年中前后衔接的一个工作.

#### 6.2.2 带高斯回代的 ADMM 方法

直接推广的乘子交替方向法 (6.3) 对三个算子的问题不能保证收敛, 是因为它们处理有关核心变量的 $y$ 和 $z$-子问题不公平. 采取补救的办法是将 (6.3) 提供的 $(y^{k+1},z^{k+1},\lambda^{k+1})$ 当成预测点, , 校正公式为

$$\begin{pmatrix} y^{k+1}\\ z^{k+1}\\ \lambda^{k+1}\end{pmatrix}:=\begin{pmatrix} y^{k}\\ z^{k}\\ \lambda^{k}\end{pmatrix}-\nu\begin{pmatrix} I & -(B^TB)^{-1}B^TC & 0\\ 0 & I & 0\\ 0 & 0 & I\end{pmatrix}\begin{pmatrix} y^k-y^{k+1}\\ z^k-z^{k+1}\\ \lambda^k-\lambda^{k+1}\end{pmatrix}. \tag{6.6}$$

其中 $\nu\in(0,1)$, 右端的 $(y^{k+1},z^{k+1},\lambda^{k+1})$ 是由 (6.3) 提供的. 这个方法发表在 [20]. 想法是不公平, 就要做找补, 调整. 事实上, 也可以就用 (6.3) 提供的 $\lambda^{k+1}$, 只通过

$$\begin{pmatrix} y^{k+1}\\ z^{k+1}\end{pmatrix}:=\begin{pmatrix} y^{k}\\ z^{k}\end{pmatrix}-\nu\begin{pmatrix} I & -(B^TB)^{-1}B^TC\\ 0 & I\end{pmatrix}\begin{pmatrix} y^k-y^{k+1}\\ z^k-z^{k+1}\end{pmatrix}. \tag{6.7}$$

校正 $y$ 和 $z$ (无需校正$\lambda$). 由于为下一步迭代只需要准备 $(By^{k+1},Cz^{k+1},\lambda^{k+1})$, 我们只要做比(6.7)更简单的

$$\begin{pmatrix} By^{k+1}\\ Cz^{k+1}\end{pmatrix}:=\begin{pmatrix} By^{k}\\ Cz^{k}\end{pmatrix}-\nu\begin{pmatrix} I & -I\\ 0 & I\end{pmatrix}\begin{pmatrix} By^k-By^{k+1}\\ Cz^k-Cz^{k+1}\end{pmatrix}. \tag{6.8}$$

有关统一框架下收敛性的简单证明可以从我主页的报告 [44] 的 §4 中找到.

### 6.2.3 部分平行并加正则项的 ADMM 方法

下面的方法与 §6.2.1 中方法相同的是平行求解 $y$, $z$-子问题, 不同的是不做后处理, 而是给这两个子问题预先都加个正则项. 方法写起来就是

$$\begin{cases} x^{k+1} = \arg\min\big\{\mathcal{L}^3_\beta(x, y^k, z^k, \lambda^k) \;\big|\; x\in\mathcal{X}\big\},\\ y^{k+1} = \arg\min\big\{\mathcal{L}^3_\beta(x^{k+1}, y, z^k, \lambda^k) + \frac{\tau}{2}\beta\|B(y-y^k)\|^2 \big| y\in\mathcal{Y}\big\},\\ z^{k+1} = \arg\min\big\{\mathcal{L}^3_\beta(x^{k+1}, y^k, z, \lambda^k) + \frac{\tau}{2}\beta\|C(z-z^k)\|^2 \big| z\in\mathcal{Z}\big\},\\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases}$$

其中 $\tau > 1$. 上述做法相当于

$$\begin{cases} x^{k+1} = \mathrm{Argmin}\{\theta_1(x) + \frac{\beta}{2}\|Ax + By^k + Cz^k - b - \frac{1}{\beta}\lambda^k\|^2 \,|\, x\in\mathcal{X}\},\\ \qquad \lambda^{k+\frac12} = \lambda^k - \beta(Ax^{k+1} + By^k + Cz^k - b)\\ y^{k+1} = \mathrm{Argmin}\{\theta_2(y) - (\lambda^{k+\frac12})^T By + \frac{\mu\beta}{2}\|B(y-y^k)\|^2 \,|\, y\in\mathcal{Y}\},\\ z^{k+1} = \mathrm{Argmin}\{\theta_3(z) - (\lambda^{k+\frac12})^T Cz + \frac{\mu\beta}{2}\|C(z-z^k)\|^2 \,|\, z\in\mathcal{Z}\},\\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{6.9}$$

其中 $\mu = \tau + 1$. 例如, 可以取 $\mu = 2.01$. 这类发表在 [21, 33] 的算法思想是: 让 $y$ 和 $z$ 太自由, 以后又不准备校正, 那就用加正则项让它们不会走得太远. [21] 中的方法被 UCLA Osher 教授的课题组成功用来求解图像降维问题[6].

上面提到的方法都与邻近点算法 PPA [27, 32] 有关. 所有的 ADMM 类分裂算法都源于增广 Lagrange 乘子法. ADMM 类方法只是对应用中出现一些实际问题有较好的计算效果. 但说到底, 还只是一阶方法.

## 7 凸优化分裂收缩算法的统一框架

建立凸优化分裂收缩算法的统一框架, 是受求解经典的单调变分不等式 (2.2) 的投影收缩算法框架 [16, 43] 的启发. 求解单调变分不等式 (2.2), 对给定的当前点 $u^k$ 和 $\beta_k > 0$, 我们利用投影

$$\tilde{u}^k = P_\Omega[u^k - \beta_k T(u^k)] = \arg\min\big\{\tfrac12\|u - [u^k - \beta_k T(u^k)]\|^2 \,|\, u\in\Omega\big\} \tag{7.1}$$

生成一个预测点 $\tilde{u}^k$. 在投影 (7.1) 中, 我们假设选取的 $\beta_k$ 满足

$$\beta_k\|T(u^k) - T(\tilde{u}^k)\| \le \nu\|u^k - \tilde{u}^k\|, \quad \nu\in(0,1). \tag{7.2}$$

由于 $\tilde{u}^k$ 是极小化问题 $\min\big\{\frac12\|u - [u^k - \beta_k T(u^k)]\|^2 \,|\, u\in\Omega\big\}$ 的解, 根据引理 2.1, 有

$$\tilde{u}^k\in\Omega, \quad (u - \tilde{u}^k)^T\{\tilde{u}^k - [u^k - \beta_k T(u^k)]\} \ge 0, \quad \forall u\in\Omega.$$

上式两边都加上 $(u - \tilde{u}^k)^T d(u^k, \tilde{u}^k)$, 其中

$$d(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k) - \beta_k[T(u^k) - T(\tilde{u}^k)], \tag{7.3}$$

就得到了我们需要的以 $\tilde{u}^k$ 为预测点的预测公式

$$[\text{预测}] \qquad \tilde{u}^k\in\Omega, \;\; (u - \tilde{u}^k)^T\beta_k T(\tilde{u}^k) \ge (u - \tilde{u}^k)^T d(u^k, \tilde{u}^k), \;\; \forall u\in\Omega. \tag{7.4}$$

将 (7.4) 中任意的 $u \in \Omega$ 选成 $u^*$, 就得到

$$(\tilde{u}^k - u^*)^T d(u^k, \tilde{u}^k) \geq \beta_k (\tilde{u}^k - u^*)^T T(\tilde{u}^k). \tag{7.5}$$

由单调性, $(\tilde{u}^k - u^*)^T T(\tilde{u}^k) \geq (\tilde{u}^k - u^*)^T T(u^*)$. 因为 $\tilde{u}^k \in \Omega$, 根据变分不等式的定义 (2.2), $(\tilde{u}^k - u^*)^T T(u^*) \geq 0$, 因而 (7.5) 的右端非负. 据此, 随后马上得到

$$(u^k - u^*)^T d(u^k, \tilde{u}^k) \geq (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k). \tag{7.6}$$

由 $d(u^k, \tilde{u}^k)$ 的表达式 (7.3) 和假设 (7.2), 利用 Cauchy-Schwarz 不等式便可得到

$$(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) \geq (1-\nu)\|u^k - \tilde{u}^k\|^2.$$

因此, 不等式 (7.6) 的右端为正. 上式可以看成

$$\nabla\Big(\frac{1}{2}\|u - u^*\|_H^2\Big)\Big|_{u=u^k} H^{-1} d(u^k, \tilde{u}^k) \geq (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k).$$

这表示, 对任何相容的正定矩阵 $H$, $H^{-1}d(u^k, \tilde{u}^k)$ 是未知函数 $\frac{1}{2}\|u - u^*\|_H^2$ 在 $u^k$ 处 $H$-模意义下的一个上升方向. 我们用

## 7.1 变分不等式形式下的统一框架

我们要求解的是单调变分不等式 (2.8). 为了收敛性证明的方便, 我们把算法的每步迭代理解成 (有时是分拆成) 预测和校正.

> **[预测]** 算法的 $k$-步迭代从 $v^k$ 开始, 通过求解一些子问题, 产生的预测点 $\tilde{w}^k \in \Omega$, 使得
>
> $$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (v - \tilde{v}^k)^T Q(v^k - \tilde{v}^k), \ \forall\, w \in \Omega, \tag{7.7}$$
>
> 成立. 其中 $Q^T + Q$ 正定.

因此, (7.7) 中的 $Q(v^k - \tilde{v}^k)$ 相当于 (7.4) 中的 $d(u^k, \tilde{u}^k)$. 由 $F$ 的单调性, 从 (7.7) 得到,

$$\begin{array}{rcl}(\tilde{v}^k - v^*)^T Q(v^k - \tilde{v}^k) & \geq & \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k) \\ & \geq & \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(w^*) \ \geq \ 0.\end{array}$$

进而得到,

$$(v^k - v^*)^T Q(v^k - \tilde{v}^k) \geq (v^k - \tilde{v}^k)^T Q(v^k - \tilde{v}^k) = \frac{1}{2}\|v^k - \tilde{v}^k\|_{(Q^T+Q)}.$$

由于上式右端为正, 取正定矩阵 $H$, $-H^{-1}Q(v^k - \tilde{v}^k)$ 就是未知距离函数 $\frac{1}{2}\|v - v^*\|_H^2$ 在 $v^k$ 处的下降方向. 记 $M = H^{-1}Q$.

> **[校正]** 我们用校正公式
>
> $$v^{k+1} = v^k - \alpha M(v^k - \tilde{v}^k), \tag{7.8}$$
>
> 产生新的迭代点.

通常, 算法中不是预先去确定好 $H$, 而是对已经有的(或者是分拆成的)预测-校正方法 (7.7) 和 (7.8), 检查下面的条件是否满足.

**收敛性条件** 对预测-校正公式中的矩阵 $Q$ 和$M$, 以及步长 $\alpha$, 有

$$H = QM^{-1} \succ 0, \tag{7.9a}$$

和

$$G = Q^T + Q - \alpha M^T HM \succ 0, \quad (\text{至少} \succeq 0). \tag{7.9b}$$

## 7.2 统一框架下的收缩性质

我们对求解变分不等式 (2.8) 的预测-校正方法 (7.7)-(7.8) 在条件 (7.9) 成立的情况下证明有关收敛性质.

**定理 7.1.** 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式 (2.8) 的预测-校正方法 (7.7)-(7.8) 生成的序列. 如果条件 (7.9) 成立, 那么有

$$\begin{aligned} &\alpha\big(\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k)\big) \\ &\quad \geq \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big) + \frac{\alpha}{2}\|v^k - \tilde{v}^k\|_G^2, \ \ \forall w \in \Omega. \end{aligned} \tag{7.10}$$

**证明**. 利用预测校正公式 (7.7)-(7.8), 我们有

$$\alpha\{\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k)\} \geq (v - \tilde{v}^k)^T H(v^k - v^{k+1}), \ \forall w \in \Omega. \tag{7.11}$$

对上式的右端利用恒等式

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

并令 $a = v$, $b = \tilde{v}^k$, $c = v^k$ 和 $d = v^{k+1}$, 就得到

$$(v - \tilde{v}^k)^T H(v^k - v^{k+1}) = \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big) + \frac{1}{2}(\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2). \tag{7.12}$$

对 (7.12) 式右端的最后一项, 利用校正公式, 就有

$$\begin{aligned} &\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2 \\ &\overset{(7.9a)}{=} \ \|v^k - \tilde{v}^k\|_H^2 - \|(v^k - \tilde{v}^k) - \alpha M(v^k - \tilde{v}^k)\|_H^2 \\ &= \ 2\alpha(v^k - \tilde{v}^k)^T HM(v^k - \tilde{v}^k) - \alpha^2(v^k - \tilde{v}^k)^T M^T HM(v^k - \tilde{v}^k) \\ &= \ \alpha(v^k - \tilde{v}^k)^T(Q^T + Q - \alpha M^T HM)(v^k - \tilde{v}^k) \overset{(7.9b)}{=} \alpha\|v^k - \tilde{v}^k\|_G^2. \end{aligned} \tag{7.13}$$

将 (7.12)和(7.13) 代入 (7.11), 定理的结论就得到证明. □

关于迭代序列 $\{v^k\}$, 我们有如下的收缩性定理.

**定理 7.2.** 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式 (2.8) 的预测-校正方法 (7.7)-(7.8) 生成的序列. 如果条件 (7.9) 成立, 那么有

$$\|v^{k+1} - v^*\|_H^2 \leq \|v^k - v^*\|_H^2 - \alpha\|v^k - \tilde{v}^k\|_G^2, \quad \forall v^* \in \mathcal{V}^*. \tag{7.14}$$

**证明**. 将 (7.10) 中任意的 $w \in \Omega$ 设为 $w^*$, 我们得到

$$\begin{aligned} & \|v^k - v^*\|_H^2 - \|v^{k+1} - v^*\|_H^2 \\ \geq \ & \alpha\|v^k - \tilde{v}^k\|_G^2 + 2\alpha\{\theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k)\}. \end{aligned} \tag{7.15}$$

利用$w^*$的最优性和$F(w)$的单调性, 就有

$$\theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k) \geq \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(w^*) \geq 0$$

因此

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

定理得证. □

## 7.3 迭代复杂性

迭代复杂性包括遍历意义下的迭代复杂性和点列意义下的迭代复杂性, 我们分别加以证明.

### 7.3.1 遍历意义下迭代复杂性

定理 7.1 也是证明遍历意义(Ergodic)迭代复杂性的基础. 利用 $F$ 的单调性, 有

$$(w - \tilde{w}^k)^T F(w) \geq (w - \tilde{w}^k)^T F(\tilde{w}^k).$$

将此代入 (7.10), (此时只要求 $G$ 半正定) 就能得到

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(w) + \frac{1}{2\alpha}\|v - v^k\|_H^2 \geq \frac{1}{2\alpha}\|v - v^{k+1}\|_H^2, \ \forall w \in \Omega. \tag{7.16}$$

**定理 7.3.** 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式 (2.8) 的预测-校正方法 (7.7)-(7.8) 生成的序列, 条件 (7.9) (此时$G$只要求半正定) 成立. 记

$$\tilde{w}_t = \frac{1}{t+1}\sum_{k=0}^{t} \tilde{w}^k. \tag{7.17}$$

那么, 对任意的正整数 $t > 0$, $\tilde{w}_t \in \Omega$ 并

$$\theta(\tilde{u}_t) - \theta(u) + (\tilde{w}_t - w)^T F(w) \leq \frac{1}{2\alpha(t+1)}\|v - v^0\|_H^2, \quad \forall w \in \Omega. \tag{7.18}$$

**证明**. $\tilde{w}_t$ 是 $\tilde{w}^0, \tilde{w}^1, \cdots, \tilde{w}^t$ 的凸组合, 因此 $\tilde{w}_t \in \Omega$ 是显然的. 对不等式 (7.16) 按 $k = 0, 1, \ldots, t$ 累加, 得到

$$(t+1)\theta(u) - \sum_{k=0}^{t}\theta(\tilde{u}^k) + \Big((t+1)w - \sum_{k=0}^{t}\tilde{w}^k\Big)^T F(w) + \frac{1}{2\alpha}\|v - v^0\|_H^2 \geq 0, \ \forall w \in \Omega.$$

利用 $\tilde{w}_t$ 的表达式, 这就可以写成

$$\frac{1}{t+1}\sum_{k=0}^{t}\theta(\tilde{u}^k) - \theta(u) + (\tilde{w}_t - w)^T F(w) \leq \frac{1}{2\alpha(t+1)}\|v - v^0\|_H^2, \quad \forall w \in \Omega. \tag{7.19}$$

因为 $\theta(u)$ 是凸函数, 根据

$$\tilde{u}_t = \frac{1}{t+1}\sum_{k=0}^{t}\tilde{u}^k \qquad \text{就有} \qquad \theta(\tilde{u}_t) \leq \frac{1}{t+1}\sum_{k=0}^{t}\theta(\tilde{u}^k).$$

将它代入 (7.19), 定理的结论就马上得到. □

### 7.3.2 点列意义下迭代复杂性

为证明点列意义下的迭代复杂性, 我们要先证明一个引理.

**引理 7.1.** 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式 (2.8) 的预测-校正方法 (7.7)-(7.8) 生成的序列, 且 (7.9a) 成立. 那么, 我们有

$$\begin{aligned}&(v^k-\tilde{v}^k)^T M^T H M[(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})]\\ \geq\;& \frac{1}{2\alpha}\|(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})\|^2_{(Q^T+Q)}.\end{aligned} \tag{7.20}$$

**证明**. 首先, 在 (7.7) 中设 $w=\tilde{w}^{k+1}$, 我们有

$$\theta(\tilde{u}^{k+1})-\theta(\tilde{u}^k)+(\tilde{w}^{k+1}-\tilde{w}^k)^T F(\tilde{w}^k)\geq(\tilde{v}^{k+1}-\tilde{v}^k)^T Q(v^k-\tilde{v}^k). \tag{7.21}$$

在 (7.7) 中之 $k$ 为 $k+1$, 并令 $w=\tilde{w}^k$, 我们有

$$\theta(\tilde{u}^k)-\theta(\tilde{u}^{k+1})+(\tilde{w}^k-\tilde{w}^{k+1})^T F(\tilde{w}^{k+1})\geq(\tilde{v}^k-\tilde{v}^{k+1})^T Q(v^{k+1}-\tilde{v}^{k+1}). \tag{7.22}$$

将 (7.21) 和 (7.22) 相加并利用 $F$ 的单调性就得到

$$(\tilde{v}^k-\tilde{v}^{k+1})^T Q[(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})]\geq 0.$$

给上式的两端都加上

$$[(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})]^T Q[(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})]$$

并利用 $v^TQv=\frac{1}{2}v^T(Q^T+Q)v$, 就有

$$(v^k-v^{k+1})^T Q[(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})]\geq\frac{1}{2}\|(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})\|^2_{(Q^T+Q)}.$$

将左端的 $(v^k-v^{k+1})$ 改写成 $\alpha M(v^k-\tilde{v}^k)$ 并利用 $Q=HM$, 我们得到(7.20), 引理得证. □

我们做好了证明本节关键定理的所有准备.

**定理 7.4.** 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式 (2.8) 的预测-校正方法 (7.7)-(7.8) 生成的序列, 且 (7.9a) 成立. 那么, 我们有

$$\|M(v^{k+1}-\tilde{v}^{k+1})\|_H\leq\|M(v^k-\tilde{v}^k)\|_H,\quad\forall\, k>0. \tag{7.23}$$

**证明**. 以 $a=M(v^k-\tilde{v}^k)$ 和 $b=M(v^{k+1}-\tilde{v}^{k+1})$ 代入恒等式

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

我们得到

$$\begin{aligned}&\|M(v^k-\tilde{v}^k)\|_H^2-\|M(v^{k+1}-\tilde{v}^{k+1})\|_H^2\\ =\;&2(v^k-\tilde{v}^k)^TM^THM[(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})]-\|M[(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})]\|_H^2.\end{aligned}$$

对以上恒等式右端的第一项使用 (7.20) 并利用 $G$ 的定义, 最后得到

$$\begin{aligned}&\|M(v^k-\tilde{v}^k)\|_H^2-\|M(v^{k+1}-\tilde{v}^{k+1})\|_H^2\\ \geq\;&\frac{1}{\alpha}\|(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})\|^2_{(Q^T+Q)}-\|M[(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})]\|_H^2\\ =\;&\frac{1}{\alpha}\|(v^k-\tilde{v}^k)-(v^{k+1}-\tilde{v}^{k+1})\|^2_G\geq 0.\end{aligned}$$

由于 $G$ 至少是半正定的, 定理得证. □

**定理 7.5.** 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式 (2.8) 的预测-校正方法 (7.7)-(7.8) 生成的序列, 且 (7.9b) 中 $G \succ 0$. 那么, 对任何的整数 $t > 0$, 我们有

$$\|M(v^t - \tilde{v}^t)\|_H^2 \le \frac{1}{(t+1)c_0}\|v^0 - v^*\|_H^2. \tag{7.24}$$

**证明**. 当矩阵 $G \succ 0$ 时, 根据 (7.14), 存在常数 $c_0$, 使得

$$\|v^{k+1} - v^*\|_H^2 \le \|v^k - v^*\|_H^2 - c_0\|M(v^k - \tilde{v}^k)\|_H^2, \quad \forall v^* \in \mathcal{V}^*.$$

接着就有

$$\sum_{k=0}^{\infty} c_0\|M(v^k - \tilde{v}^k)\|_H^2 \le \|v^0 - v^*\|_H^2, \quad \forall\, v^* \in \mathcal{V}^*. \tag{7.25}$$

根据定理 7.4, 序列 $\{\|M(v^k - \tilde{v}^k)\|_H^2\}$ 是单调不增的. 所以, 我们有

$$(t+1)\|M(v^t - \tilde{v}^t)\|_H^2 \le \sum_{k=0}^{t}\|M(v^k - \tilde{v}^k)\|_H^2. \tag{7.26}$$

从 (7.25) 和 (7.26) 可以直接得到 (7.24), 定理得证. □

定理 7.5 说明 $\|M(v^t - \tilde{v}^t)\|$ 或者$\|v^t - \tilde{v}^t\|$ 有 $O(1/\sqrt{t})$ 的收敛速率. 注意到对可微凸函数 $f(x)$, 如果 $x^*$ 是极小点, 由于 $\nabla f(x^)=0$, 因此有

$$f(x) - f(x^*) = \nabla f(x^*)^T(x - x^*) + O(\|x - x^*\|^2) = O(\|x - x^*\|^2).$$

因此也可以看做目标函数值意义下的 $O(1/t)$ 收敛速率.

有关统一框架下收敛性的简单证明可以从我主页的第三个报告 [44] 的 §4 中找到. 变分不等式是分析凸优化分裂收缩算法的有力工具. 在变分不等式框架下收敛性证明非常简单.

# 8 用统一框架验证收敛性

这一节验证经典的 ADMM 及其 §5 提到的主要进展, 以及 §6 中的关于三个算子的所有 ADMM 类算法, 都可以纳入预测-校正的统一框架 (7.7)-(7.8). 验证了条件 (7.9), 算法就有了 §7.2 中的收缩性质和 §7.3 中的迭代复杂性.

## 8.1 两个算子的乘子交替方向法及其改进

两个算子的经典的 ADMM 是指算法 (4.9). 此外, 我们验证 §5.1.1 和 §5.1.2 中介绍的方法.

### 8.1.1 经典的交替方向法

对经典的交替方向法, 利用迭代开始给定的 $(y^k, \lambda^k)$ 和 (4.9) 产生的 $x^{k+1}$ 和 $y^{k+1}$, 通过定义

$$\tilde{x}^k = x^{k+1}, \quad \tilde{y}^k = y^{k+1}, \quad \text{和} \quad \tilde{\lambda}^k = \lambda^k - \beta(Ax^{k+1} + By^k - b),$$

可以把算法“分拆”成预测-校正方法. 预测由

$$\left\{\begin{array}{lcl} \tilde{x}^k &=& \mathrm{Argmin}\big\{\theta_1(x) - (\lambda^k)^T Ax + \frac{\beta}{2}\|Ax + By^k - b\|^2 \,\big|\, x \in \mathcal{X}\big\}, \\ \tilde{y}^k &=& \mathrm{Argmin}\big\{\theta_2(y) - (\lambda^k)^T By + \frac{\beta}{2}\|A\tilde{x}^k + By - b\|^2 \,\big|\, y \in \mathcal{Y}\big\}, \\ \tilde{\lambda}^k &=& \lambda^k - \beta(A\tilde{x}^k + By^k - b). \end{array}\right.$$

实现, 校正由

$$x^{k+1}=\tilde{x}^k,\quad y^{k+1}=\tilde{y}^k,\quad \text{和}\quad \lambda^{k+1}=\tilde{\lambda}^k+\beta B(y^k-\tilde{y}^k)$$

完成. 根据引理 2.1, 预测可以写成 (7.7) 的形式, 其中

$$Q=\begin{pmatrix}\beta B^TB & 0\\ -B & \frac{1}{\beta}I\end{pmatrix}.$$

由于$y^{k+1}=\tilde{y}^k$ 和 $\lambda^{k+1}=\tilde{\lambda}^k+\beta B(y^k-\tilde{y}^k)=\lambda^k-[-\beta B(y^k-\tilde{y}^k)+(\lambda^k-\tilde{\lambda}^k)]$, 校正公式可以写成 (7.8) 的形式, 其中

$$M=\begin{pmatrix}I & 0\\ -\beta B & I\end{pmatrix}\quad \text{和}\quad \alpha=1.$$

由此得到

$$H=QM^{-1}=\begin{pmatrix}\beta B^TB & 0\\ 0 & \frac{1}{\beta}I\end{pmatrix}\quad \text{和}\quad G=Q^T+Q-M^THM=\begin{pmatrix}0 & 0\\ 0 & \frac{1}{\beta}I\end{pmatrix},$$

条件 (7.9) 满足. 虽然这里 $G$ 是半正定的, 此时定理 7.2 中的结论为

$$\|v^{k+1}-v^*\|_H^2\le\|v^k-v^*\|_H^2-\frac{1}{\beta}\|\lambda^k-\tilde{\lambda}^k\|^2,\quad \forall v^*\in\mathcal{V}^*.$$

然而, 对于经典的交替方向法, 我们有 (见 [44])

$$\frac{1}{\beta}\|\lambda^k-\tilde{\lambda}^k\|^2\ge\frac{1}{\beta}\|\lambda^k-\lambda^{k+1}\|^2+\beta\|B(y^k-y^{k+1})\|^2.$$

因此, 利用矩阵 $H$ 的结构, 就有

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

利用定理 7.4 和校正公式 $v^{k+1}=v^k-M(v^k-\tilde{v}^k)$, 还有

$$\|v^k-v^{k+1}\|_H^2\le\|v^{k-1}-v^k\|_H^2$$

这个在 [25] 中的主要结果, 利用 §7 的统一框架就非常容易得到.

### 8.1.2 PPA 意义下延拓的乘子交替方向法

对 §5.1.1 中介绍的 PPA 意义下延拓的乘子交替方向法, 我们把 (5.3a) 和 (5.3b) 分别看做预测和校正. 将预测点记为 $\tilde{w}^k=(\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)$, 预测-校正公式就分别是

$$\left\{\begin{array}{rcl}\tilde{x}^k & = & \text{Argmin}\{\mathcal{L}_\beta(x,y^k,\lambda^k)\,|\,x\in\mathcal{X}\},\\ \tilde{\lambda}^k & = & \lambda^k-\beta(A\tilde{x}^k+By^k-b),\\ \tilde{y}^k & = & \text{Argmin}\{\mathcal{L}_\beta(\tilde{x}^k,y,\tilde{\lambda}^k)\,|\,y\in\mathcal{Y}\},\end{array}\right.$$

和

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

根据引理 2.1, 预测可以写成 (7.7) 的形式, 其中

$$Q=\begin{pmatrix}\beta B^TB & -B^T\\ -B & \frac{1}{\beta}I\end{pmatrix}.$$

校正公式 (7.8) 中, $M$ 为单位矩阵, $\alpha \in (0,2)$. 由此得到

$$H = Q \qquad \text{和} \qquad G = (2-\alpha)Q.$$

由于 $Q$ 是半正定的, 条件 (7.9) 满足. 所以, §5.1.1 中的方法符合 §7.1 中的统一的预测-校正框架 (7.7)-(7.8) 及其收敛性条件 (7.9). 虽然这里 $H = Q$ 是半正定的, 实际计算并不影响收敛.

### 8.1.3 对称更新乘子的交替方向法

对于 §5.1.2 中对称更新乘子的交替方向法, 利用迭代开始给定的 $(y^k, \lambda^k)$ 和 (5.4) 产生的 $x^{k+1}$ 和 $y^{k+1}$, 通过定义

$$\tilde{x}^k = x^{k+1}, \quad \tilde{y}^k = y^{k+1}, \quad \text{和} \quad \tilde{\lambda}^k = \lambda^k - \beta(Ax^{k+1} + By^k - b),$$

预测就是通过

$$\begin{cases} \tilde{x}^k &=& \operatorname{Argmin}\{\theta_1(x) - (\lambda^k)^T Ax + \frac{\beta}{2}\|Ax + By^k - b\|^2 \big| x \in \mathcal{X}\}, \\ \tilde{y}^k &=& \operatorname{Argmin}\{\theta_2(y) - [\tilde{\lambda}^k + \mu(\tilde{\lambda}^k - \lambda^k)]^T By + \frac{\beta}{2}\|A\tilde{x}^k + By - b\|^2 \big| y \in \mathcal{Y}\}, \\ \tilde{\lambda}^k &=& \lambda^k - \beta(A\tilde{x}^k + By^k - b), \end{cases}$$

实现的. 根据引理 2.1, 预测可以写成 (7.7) 的形式, 其中

$$Q = \begin{pmatrix} \beta B^T B & -\mu B^T \\ -B & \frac{1}{\beta} I \end{pmatrix}.$$

利用 $v^{k+1}$ 与 $v^k$ 和 $\tilde{v}^k$ 的关系, 有

$$y^{k+1} = \tilde{y}^k, \quad \text{和} \quad \lambda^{k+1} = \lambda^k - [-\mu\beta B(y^k - \tilde{y}^k) + 2\mu(\lambda^k - \tilde{\lambda}^k)].$$

这样就相当于校正公式 (7.8) 中取 $\alpha = 1$ 和

$$M = \begin{pmatrix} I & 0 \\ -\mu\beta B & 2\mu I \end{pmatrix}.$$

由此得到

$$H = QM^{-1} = \begin{pmatrix} (1-\frac{1}{2})\beta B^T B & -\frac{1}{2} B^T \\ -\frac{1}{2} B & \frac{1}{2\mu\beta} I \end{pmatrix} \quad \text{和} \quad G = (1-\mu) \begin{pmatrix} \beta B^T B & -B^T \\ -B & \frac{2}{\beta} I \end{pmatrix}.$$

所以, §5.1.2 中的方法符合 §7.1 中的统一的预测-校正框架 (7.7)-(7.8), 并满足收敛性条件 (7.9).

## 8.2 三个算子的 ADMM 类方法

现在用 §7 中的统一框架验证 §6.2 中介绍的求解三个算子问题的三种 ADMM 类方法.

### 8.2.1 部分平行分裂 ALM 的预测校正方法

求解三个可分离算子凸优化 (6.1), §6.2.1 中介绍了部分平行分裂 ALM 的预测校正方法. 把由 (6.4) 生成的 $(x^{k+1}, y^{k+1}, z^{k+1})$ 视为 $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k)$, 并定义

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k + Cz^k - b).$$

这样, 预测点 $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$ 就可以看成有下式生成：

$$\left\{ \begin{array}{lcl} \tilde{x}^k &=& \arg\min\big\{\mathcal{L}^3_\beta(x, y^k, z^k, \lambda^k) \bigm| x \in \mathcal{X}\big\}, \\ \tilde{y}^k &=& \arg\min\big\{\mathcal{L}^3_\beta(\tilde{x}^k, y, z^k, \lambda^k) \bigm| y \in \mathcal{Y}\big\}, \\ \tilde{z}^k &=& \arg\min\big\{\mathcal{L}^3_\beta(\tilde{x}^k, y^k, z, \lambda^k) \bigm| z \in \mathcal{Z}\big\}, \\ \tilde{\lambda}^k &=& \lambda^k - \beta(A\tilde{x}^k + By^k + Cz^k - b). \end{array} \right.$$

利用引理 2.1, 预测可以写成统一框架中的 (7.7), 其中

$$Q = \begin{pmatrix} \beta B^T B & 0 & 0 \\ 0 & \beta C^T C & 0 \\ -B & -C & \frac{1}{\beta} I \end{pmatrix}. \tag{8.1}$$

注意到这时 (6.5) 右端的 $v^{k+1}$ 中,

$$y^{k+1} = \tilde{y}^k, \quad z^{k+1} = \tilde{z}^k, \quad \text{和} \quad \lambda^{k+1} = \tilde{\lambda}^k + \beta B(y^k - \tilde{y}^k) + \beta C(y^k - \tilde{y}^k).$$

因此, 利用预测点, 校正公式(6.5) 就可以写成

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ z^k \\ \lambda^k \end{pmatrix} - \alpha \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\beta B & -\beta C & I \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

也就是说, 在统一框架的校正公式 (7.8) 中

$$M = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\beta B & -\beta C & I \end{pmatrix}. \tag{8.2}$$

对这样的 $Q$ 和 $M$, 设

$$H = \begin{pmatrix} \beta B^T B & 0 & 0 \\ 0 & \beta C^T C & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix},$$

就有 $HM = Q$, 说明收敛性条件 (7.9a) 满足. 简单的矩阵运算就得到

$$\begin{array}{rcl} G &=& (Q^T + Q) - \alpha M^T H M = (Q^T + Q) - \alpha M^T Q \\ &=& \begin{pmatrix} \sqrt{\beta} B^T & 0 & 0 \\ 0 & \sqrt{\beta} C^T & 0 \\ 0 & 0 & \frac{1}{\sqrt{\beta}} I \end{pmatrix} \begin{pmatrix} 2(1-\alpha) I & -\alpha I & -(1-\alpha) I \\ -\alpha I & 2(1-\alpha) I & -(1-\alpha) I \\ -(1-\alpha) I & -(1-\alpha) I & (2-\alpha) I \end{pmatrix} \begin{pmatrix} \sqrt{\beta} B & 0 & 0 \\ 0 & \sqrt{\beta} C & 0 \\ 0 & 0 & \frac{1}{\sqrt{\beta}} I \end{pmatrix}. \end{array}$$

容易验证, 对所有的 $\alpha \in (0, 2-\sqrt{2})$, 矩阵

$$\begin{pmatrix} 2(1-\alpha) & -\alpha & -(1-\alpha) \\ -\alpha & 2(1-\alpha) & -(1-\alpha) \\ -(1-\alpha) & -(1-\alpha) & (2-\alpha) \end{pmatrix} \succ 0.$$

收敛性条件 (7.9b) 满足. 所以, §6.2.1 中的方法符合 §7.1 中的统一的预测-校正框架 (7.7)-(7.8), 并满足收敛性条件 (7.9). 具备 §7 中的所有收敛性质.

### 8.2.2 带高斯回代的 ADMM 方法

求解三个可分离算子凸优化 (6.1), §6.2.2 中介绍了带高斯回代的 ADMM 方法. 我们把由直接推广的 (6.3) 生成的 $x^{k+1}, y^{k+1}, z^{k+1}$ 分别视为 $\tilde{x}^k$, $\tilde{y}^k$, $\tilde{z}^k$, 并定义

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k + Cz^k - b).$$

把 $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$ 看做预测点, 它由下面的公式

$$\left\{ \begin{array}{lcl} \tilde{x}^k & = & \operatorname{Argmin}\big\{\theta_1(x) - (\lambda^k)^T Ax + \frac{\beta}{2}\|Ax + By^k + Cz^k - b\|^2 \big| x \in \mathcal{X}\big\}, \\ \tilde{y}^k & = & \operatorname{Argmin}\big\{\theta_2(y) - (\lambda^k)^T By + \frac{\beta}{2}\|A\tilde{x}^k + By + Cz^k - b\|^2 \big| y \in \mathcal{Y}\big\}, \\ \tilde{z}^k & = & \operatorname{Argmin}\big\{\theta_3(z) - (\lambda^k)^T Cz + \frac{\beta}{2}\|A\tilde{x}^k + B\tilde{y}^k + Cz - b\|^2 \big| z \in \mathcal{Z}\big\}, \\ \tilde{\lambda}^k & = & \lambda^k - \beta(A\tilde{x}^k + By^k + Cz^k - b). \end{array} \right.$$

这样, 利用引理 2.1, 预测就可以写成统一框架中的 (7.7) 式, 其中

$$Q = \begin{pmatrix} \beta B^T B & 0 & 0 \\ \beta C^T B & \beta C^T C & 0 \\ -B & -C & \frac{1}{\beta} I \end{pmatrix}.$$

利用这样的预测点, 只校正 $y$ 和 $z$ 的公式 (6.7) (注意 $\lambda^{k+1}$ 和 $\tilde{\lambda}^k$ 的关系) 就可以写成

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ z^k \\ \lambda^k \end{pmatrix} - \begin{pmatrix} \nu I & -\nu(B^T B)^{-1} B^T C & 0 \\ 0 & \nu I & 0 \\ -\beta B & -\beta C & I \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

也就是说, 在统一框架的校正公式 (7.8) 中

$$M = \begin{pmatrix} \nu I & -\nu(B^T B)^{-1} B^T C & 0 \\ 0 & \nu I & 0 \\ -\beta B & -\beta C & I \end{pmatrix}. \tag{8.3}$$

对于矩阵

$$H = \begin{pmatrix} \frac{1}{\nu}\beta B^T B & \frac{1}{\nu}\beta B^T C & 0 \\ \frac{1}{\nu}\beta C^T B & \frac{1}{\nu}\beta[C^T C + C^T B(B^T B)^{-1} B^T C] & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix} \tag{8.4}$$

可以验证 $H$ 正定并有 $HM = Q$, 这说明收敛性条件 (7.9a) 满足. 此外,

$$\begin{aligned} G &= (Q^T + Q) - M^T H M = (Q^T + Q) - M^T Q \\ &= \begin{pmatrix} 2\beta B^T B & \beta B^T C & -B^T \\ \beta C^T B & 2\beta C^T C & -C^T \\ -B & -C & \frac{2}{\beta} I \end{pmatrix} - \begin{pmatrix} (1+\nu)\beta B^T B & \beta B^T C & -B^T \\ \beta C^T B & (1+\nu)\beta C^T C & -C^T \\ -B & -C & \frac{1}{\beta} I \end{pmatrix} \\ &= \begin{pmatrix} (1-\nu)\beta B^T B & 0 & 0 \\ 0 & (1-\nu)\beta C^T C & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix}. \end{aligned}$$

由于 $\nu \in (0,1)$, 矩阵 $G$ 正定, 收敛性条件 (7.9b) 满足. 所以, §6.2.2 中的方法符合 §7.1 中的统一的预测-校正框架 (7.7)-(7.8), 并满足收敛性条件 (7.9).

### 8.2.3 部分平行并加正则项的 ADMM 方法

对 §6.2.3 中介绍的部分平行并加正则项的 ADMM 方法, 我们把 (6.9) 生成的 $(x^{k+1}, y^{k+1}, z^{k+1}, \lambda^{k+\frac{1}{2}})$ 视为预测点 $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$. 这个预测公式就成为

$$\begin{cases} \tilde{x}^k = \mathrm{Argmin}\{\theta_1(x) - (\lambda^k)^T Ax + \frac{\beta}{2}\|Ax + By^k + Cz^k - b\|^2 \mid x \in \mathcal{X}\}, \\ \tilde{y}^k = \mathrm{Argmin}\{\theta_2(y) - (\tilde{\lambda}^k)^T By + \frac{\mu\beta}{2}\|B(y - y^k)\|^2 \mid y \in \mathcal{Y}\}, \\ \tilde{z}^k = \mathrm{Argmin}\{\theta_3(z) - (\tilde{\lambda}^k)^T Cz + \frac{\mu\beta}{2}\|C(z - z^k)\|^2 \mid z \in \mathcal{Z}\}, \\ \tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k + Cz^k - b). \end{cases} \tag{8.5}$$

这样, 利用引理 2.1, 预测就可以写成统一框架中的 (7.7) 式, 其中

$$Q = \begin{pmatrix} \mu\beta B^T B & 0 & 0 \\ 0 & \mu\beta C^T C & 0 \\ -B & -C & \frac{1}{\beta} I \end{pmatrix}.$$

利用这样的预测点, 校正 $y$ 和 $z$ 的公式 (注意 $\lambda^{k+1}$ 和 $\tilde{\lambda}^k$ 的关系) 就可以写成

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ z^k \\ \lambda^k \end{pmatrix} - \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\beta B & -\beta C & I \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

也就是说, 在统一框架的校正公式 (7.8) 中

$$M = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\beta B & -\beta C & I \end{pmatrix}. \tag{8.6}$$

对于矩阵

$$H = \begin{pmatrix} \mu\beta B^T B & 0 & 0 \\ 0 & \mu\beta C^T C & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix},$$

可以验证 $H$ 正定并有 $HM = Q$. 这说明收敛性条件 (7.9a) 满足. 此外,

$$
\begin{aligned}
G &= (Q^T + Q) - M^T H M = (Q^T + Q) - M^T Q \\
&= \begin{pmatrix} 2\mu\beta B^T B & 0 & -B^T \\ 0 & 2\mu\beta C^T C & -C^T \\ -B & -C & \frac{2}{\beta} I \end{pmatrix} - \begin{pmatrix} (1+\mu)\beta B^T B & \beta B^T C & -B^T \\ \beta C^T B & (1+\mu)\beta C^T C & -C^T \\ -B & -C & \frac{1}{\beta} I \end{pmatrix} \\
&= \begin{pmatrix} (\mu-1)\beta B^T B & -\beta B^T C & 0 \\ -\beta C^T B & (\mu-1)\beta C^T C & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix}.
\end{aligned}
$$

由于 $\mu > 2$, 矩阵 $G$ 正定, 收敛性条件 (7.9b) 满足. 所以, §6.2.3 中的方法符合 §7.1 中的统一的预测-校正框架 (7.7)-(7.8), 并满足收敛性条件 (7.9).

# 9 结论和体会

本文回顾了我们 20 年来在单调变分不等式和凸优化方面的 ADMM 类方法的工作. 利用一个框架, 统一了近 10 年在凸优化的 ADMM 类方法的进展, 自成系统. 特点是简单与统一, 一个模式, 一条主线. 我们深信, 只有简单, 他人才会看懂使用; 因为统一, 自己才有美的享受. 追求简单与统一, 是我们研究工作欲罢不能的原因.

求解线性约束凸优化, 增广拉格朗日乘子法 (ALM) 优于罚函数方法, 我们把它解释为在“对原始变量求极小的时候顾及对偶变量的感受”. 对两个可分离算子的线性约束凸优化问题, 增广拉格朗日乘子法 (ALM) 和罚函数方法, 松弛后分别成了乘子交替方向法 (ADMM) 和交替极小化方法 (AMA). ADMM 优于 AMA 也毫无疑问.

ADMM 不是我们提出来的. 因为交通网络分析的需求和已有的10 多年变分不等式投影收缩算法的基础, 我们从 1997 年开始对 ADMM 研究. 最近 10 年发现 ADMM 在结构型优化上有广泛应用, 带领学生对ADMM 方法做一些有价值的改进和证明一些重要的理论结果, 就顺理成章.

方法上, 交换了原始变量 $y$ 和对偶变量 $\lambda$ 的次序, 进而得到因需定制的 PPA 意义下的 ADMM [2]. 平等对待原始变量 $x$ 和 $y$, 两次校正对偶变量 $\lambda$, 就得到对称型的 ADMM [19]. 这些方法, 道理上能站住脚, 计算表现也不俗.

理论上, 我们对 ADMM 在遍历意义下 [24] 和点列意义下 [25] 的收敛速率给出了证明. 利用变分不等式框架, 这些证明都不复杂.

ADMM 的广泛应用, 人们自然想到向三个算子和多个算子的问题推广. 我们在不能证明“直接推广的方法”收敛的时候, 提出了一些处理多个算子问题的 ADMM 类方法:

- “不公平就找补校正”的方法 [20],
- “各自为政处理子问题,就必须加正则项加强自我节制”[21],

采取这些策略, 手段上是必须的, 机制上也是合理的. 由于经典的 ADMM (4.9) 收敛速度严重依赖于罚参数 $\beta$ 的选择, 我们提出的用于求解三个算子问题的 ADMM 类方法, 保持了自由选择 $\beta$. 或者可以说, 这些修正的方法, 最大限度地保持了经典 ADMM 的优良品性. 这 20 年的 ADMM 类算法研究历程, 生活理念的帮助, 使得我们一步一步走的大致不错.

最后需要指出的, ADMM 类算法不是解决一切问题的方法. ADMM 类算法是松弛了的拉格朗日乘子法 (ALM), 增广拉格朗日乘子法又是线性约束凸优化对偶问题的邻近点算法 (PPA). ADMM 类算法只是利用了问题的可分离结构. PPA 和 ALM 具有的基因短处, ADMM 类方法仍然会有. 如果设计ADMM类算法对问题本身还要外加一些额外的条件, 那就等于在本不强壮的 PPA 和 ALM 身上再另加负担. 那样做, 显然不“厚道”, 也不值得称道. 通篇以“我”为主, 是为了便于说明研究进展的来龙去脉. 我做这些研究, 得益于中学的数理基础和独特的人生经历, 用到的只是普通的大学数学和一般的优化常识. ADMM类算法这些比较系统的成果, 说明让人走自己的路, 找一个与自己能力适合的课题特别重要.

## References


[1] Boyd S, Parikh N, Chu E, Peleato B and Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends in Machine Learning*, 2011, 3: 1–122.

[2] 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.

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

[4] 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.

[5] D. Chen and Y. Zhang, A Hybrid Multi-Objective Scheme Applied to Redundant Robot Manipulators, IEEE Transactions on Automation Science and Engineering, 2017, 14: 1337–1350.

[6] 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, *Transactions on Image Processing*, 2012, 21: 3239–3252.

[7] Gabay D. Applications of the method of multipliers to variational inequalities, in *Augmented Lagrange methods: applications to the solution of boundary-valued problems*, M. Fortin and R. Glowinski, eds., North Holland, Amsterdam, The Netherlands, 1983, 299–331.

[8] Glowinski R. *Numerical methods for nonlinear variational problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[9] Guo D S and Zhang Y N. Simulation and experimental verification of weighted velocity and acceleration minimization for robotic redundancy resolution, *IEEE Transactions on Automation Science and Engineering*, 2014, 11: 1203–1217.

[10] Hallac D, Wong C, Diamond S,Sharang A, Sosič R, Boyd S and Leskovec J. SnapVX: A network-based convex optimization solver, *Journal of Machine Learning Research*, 2017, 18: 1–5.

[11] He B S. A new method for a class of linear variational inequalities, *Mathematical Programming*, 1994, 66: 137–144.

[12] He B S. Solving a class of linear projection equations, *Numerische Mathematik*, 1994, 68: 71-80.

[13] He B S. A class of projection and contraction methods for monotone variational inequalities, *Applied Mathematics and Optimization*, 1997, 35: 69–76.

[14] He B S. Inexact implicit methods for monotone general variational inequalities, *Mathematical Programming*, 1999, 86: 199–217.

[15] He B S. Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities, *Computational Optimization and Applications*, 2009, 42: 195–212.

[16] He B S and Liao L Z. Improvements of some projection methods for monotone nonlinear variational inequalities, *Journal of Optimization Theory and Applications*, 2002, 112: 111–128.

[17] He B S, Liao L Z, Han D R and Yang H. A new inexact alternating directions method for monotone variational inequalities, *Mathematical Programming*, 2002, 92: 103–118.

[18] He B S, Liao L Z and Qian M J. Alternating projection based prediction-correction methods for structured variational inequalities, *Journal of Computational Mathematics*, 2006, 24: 693–710.

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

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

[21] He B S, Tao M and Yuan X M. A splitting method for separable convex programming, *IMA Journal of Numerical Analysis*, 2015, 31: 394–426.

[22] He B S and Yang H. Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities, *Operations Research Letters*, 1998, 23: 151–161.

[23] He B S, Yang H and Wang S L. Alternating directions method with self-adaptive penalty parameters for monotone variational inequalities, *Journal of Optimization Theory and Applications*, 2003, 23: 349–368.

[24] He B S and Yuan X M. On the $O(1/n)$ convergence rate of the alternating direction method, *SIAM Journal of Numerical Analysis*, 2012, 50: 700–709.

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

[26] Hestenes M R. Multiplier and gradient methods, *Journal of Optimization Theory and Applications*, 1969, 4: 302–320.

[27] Martinet B. Regularisation, d'inéquations variationelles par approximations succesives, *RIRO*, 1970, 4: 154-159.

[28] Nagurney A. *Network economics: a variational inequality approach*, Kluwer Academic Publishers, Dordrecht, Boston, London, 1993.

[29] Nagurney A and Ramanujam P. Transportation network policy modeling with goal targets and generalized penalty functions, *Transportation Science* 1996, 30: 3–13.

[30] Nocedal J and Wright S. *Numerical optimization*, Springer Verlag, New York, 1999.

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

[32] Rockafellar R T. Monotone operators and the proximal point algorithm, *SIAM Journal on Control and Optimization*, 1976, 14: 877–898.

[33] Tao M and Yuan X M. Recovering low-rank and sparse components of matrices from incomplete and noisy observations *SIAM Journal on Optimization*, 2011, 21: 57-81.

[34] Wang Y L, Yang J F, Yin W T and Zhang Y. A new alternating minimization algorithm for total variation image reconstruction, *SIAM Journal on Imaging Sciences*, 2008, 1: 248–272.

[35] Xiao L and Zhang Y N. Acceleration-level repetitive motion planning and its experimental verification on six-link planar robot manipulator, *IEEE Transactions on Control System Technology*, 2013, 21: 906–914.

[36] Yang J F and Zhang Y. Alternating direction algorithms for $l_1$-problems in compressive sensing, *SIAM Journal on Scientific Computing*, 2011, 33: 250-278.

[37] Yang J F, Zhang Y, Yin W T. A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data, *IEEE Journal of Selected Topics in Signal Processing*, 2010, 4: 288–297.

[38] Yunong Zhang, Senbo Fu, Zhijun Zhang, and Lin Xiao, On the LVI-Based Numerical Method (E47 Algorithm) for Solving Quadratic Programming Problems, IEEE International Conference on Automation and Logistics, 2011.

[39] Yunong Zhang, Long Jin, Robot Manipulator Redundancy Resolution, Hoboken, Wiley, 2017

[40] Zheng H, Liu F and Du X L. Complementarity problem arising from static growth of multiple cracks and MLS-based numerical manifold method, *Computer Methods in Applied Mechanics and Engineering*, 2015, 295: 150–171.

[41] Zheng H, Zhang P and Du X L. Dual form of discontinuous deformation analysis, *Computer Methods in Applied Mechanics and Engineering*, 2016, 305: 196–216.

[42] 何炳生. 论求解变分不等式的一些投影收缩算法, 计算数学, 1996, 18: 54–60.

[43] 何炳生. 从变分不等式的统一收缩算法到凸优化的分裂收缩算法, 高等学校计算数学学报, 2016, 38: 74–96.

[44] 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》.

[45] 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.

# My 20 years research on alternating directions method of multipliers

He BingSheng

**Abstract** My research on ADMM dates back to 1997 when I considered the problems from traffic network analysis. Over the last 10 years, the ADMM based on variational inequalities is widely used in optimization. This paper summarizes our research on ADMM over the last 20 years, particularly, the developments in splitting and contraction methods based on ADMM for convex optimization over the last 10 years. We list the main results as well as the motivations. Our analysis is based on the variational inequalities. All methods mentioned fall in a simple unified prediction-correction framework, in which the convergence analysis is quite simple. A through reading will acquaint you with the ADMM, while a more carefully reading may make you familiar with the tricks on constructing splitting methods according to the problem you met. We should notice that the ADMM originates from ALM and PPA, which are good at utilizing the splitting structure. However, it also inherits the intrinsic shortcomings of these first order methods.

**Keywords** Convex optimization, monotone variational inequality, alternating directions method of multipliers, contractive properties, $O(1/t)$ convergence rate, unified framework.