

文章编号: 1004-4353(2021)03-0216-06

一种放松参数的部分并行 ADMM 算法

金青龙, 高前明

(南京财经大学 应用数学学院, 南京 210023)

摘要: 为了求解多块线性约束可分凸优化问题, 提出了一种放松参数的部分并行交替方向乘子法 (ADMM) 算法—PPADMMR 算法. 该算法在子问题中引入带参数的临近项, 放松了临近参数范围. 数值实验表明, PPADMMR 算法的收敛速度优于部分并行 ADMM (PPADMM) 算法, 因此提出的 PPADMMR 算法可为研究快速 ADMM 算法提供参考.

关键词: 可分凸优化; 交替方向乘子法; 部分并行分裂; 变分不等式; 多块

中图分类号: O159

文献标识码: A

A partially parallel ADMM algorithm with relaxed parameters

JIN Qinglong, GAO Qianming

(School of Applied Mathematics, Nanjing University of Finance and Economics, Nanjing 210023, China)

Abstract: To solve multi-block linearly constrained separable convex optimization problems, we propose a partially parallel alternating direction multiplier method with relaxed parameters (PPADMMR). The algorithm introduces the proximal point terms with parameters into the sub-problems and relaxes the range of parameters. Numerical experiments show that the convergence speed of the new algorithm is better than that of PPADMM. Therefore, the proposed PPADMMR algorithm in this paper can provide an excellent reference to investigate the faster ADMM algorithm.

Keywords: separable convex optimization; alternating direction multiplier method; partial parallel splitting; variational inequality; multiple blocks

0 引言

近年来随着对优化问题的深入研究, 一些学者研究了如下具有多块变量的线性可分凸优化模型:

$$\min \left\{ \sum_{i=1}^m \theta_i(x_i) \mid \sum_{i=1}^m A_i x_i = b, x_i \in X_i, i = 1, \dots, m \right\}, \quad (1)$$

其中, $\theta_i: \mathbf{R}^{n_i} \rightarrow \mathbf{R} (i = 1, \dots, m)$ 是闭的凸函数 (不一定光滑); $A \in \mathbf{R}^{l \times n_i}$ 、 $b \in \mathbf{R}^l$ 和 $X_i \subset \mathbf{R}^{n_i}$ 是给定的闭凸集. 假设模型 (1) 的解集是非空的, 矩阵 $A \in \mathbf{R}^{l \times n_i}$ 列满秩, 则模型 (1) 的拉格朗日函数为

$$L^m(x_1, x_2, \dots, x_m, \lambda) = \sum_{i=1}^m \theta_i(x_i) - \lambda^T \left(\sum_{i=1}^m A_i x_i - b \right), \quad (2)$$

增广拉格朗日函数为

$$L_A^m(x_1, x_2, \dots, x_m, \lambda) = L^m(x_1, x_2, \dots, x_m, \lambda) + \frac{\beta}{2} \left\| \sum_{i=1}^m A_i x_i - b \right\|^2, \quad (3)$$

收稿日期: 2021-07-06

基金项目: 江苏省研究生科研与实践创新计划 (KYCX20_1321)

作者简介: 金青龙 (1996—), 男, 硕士, 研究方向为最优化算法理论与设计.

$$\text{步骤 2(校正步)} \quad \begin{cases} x_1^{k+1} = \tilde{x}_1^k, \\ v^{k+1} = v^k - \mathbf{M}(v^k - \tilde{v}^k). \end{cases}$$

2.1 收敛性分析

为证明本文提出的算法具有收敛性,首先证明以下 2 个引理.

引理 1 给定 w^k, \tilde{w}^k 由上述步骤 1 生成,则有 $\tilde{w}^k \in \Omega$, 且

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (\tilde{v}^k - v)^T \mathbf{Q}(\tilde{v}^k - v^k). \quad (9)$$

证明 根据一阶最优性条件,步骤 1 中的 x_1 -子问题可以写为

$$\theta_1(x_1) - \theta_1(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^T \{-\mathbf{A}_1^T \tilde{\lambda}^k + s\beta \mathbf{A}_1^T (\mathbf{A}_1 \tilde{x}_1^k + \sum_{j=2}^m \mathbf{A}_j x_j^k - b)\} \geq 0. \quad (10)$$

x_i -子问题可以写为

$$\begin{aligned} \theta_i(x_i) - \theta_i(\tilde{x}_i^k) + (x_i - \tilde{x}_i^k)^T \{-\mathbf{A}_i^T \tilde{\lambda}^k + s\beta \mathbf{A}_i^T (\mathbf{A}_i \tilde{x}_i^k + \sum_{j=1, j \neq i}^m \mathbf{A}_j x_j^k - b) + \\ r\beta \mathbf{A}_i^T \mathbf{A}_i (\tilde{x}_i^k - x_i^k)\} \geq 0. \end{aligned} \quad (11)$$

由于 $\tilde{\lambda}^k = \lambda^k - s\beta(\mathbf{A}_1 \tilde{x}_1^k + \sum_{j=2}^m \mathbf{A}_j x_j^k - b)$, 因此式(10) 和式(11) 可以分别被写成如下形式:

$$\theta_1(x_1) - \theta_1(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^T \{-\mathbf{A}_1^T \tilde{\lambda}^k\} \geq 0, \quad (12)$$

$$\theta_i(x_i) - \theta_i(\tilde{x}_i^k) + (x_i - \tilde{x}_i^k)^T \{-\mathbf{A}_i^T \tilde{\lambda}^k + s\beta \mathbf{A}_i^T (\mathbf{A}_i (\tilde{x}_i^k - x_i^k)) + r\beta \mathbf{A}_i^T \mathbf{A}_i (\tilde{x}_i^k - x_i^k)\} \geq 0. \quad (13)$$

又因为 $\tilde{\lambda}^k = \lambda^k - s\beta(\mathbf{A}_1 \tilde{x}_1^k + \sum_{j=2}^m \mathbf{A}_j x_j^k - b)$, 所以式(12)、(13) 可以写成如下形式:

$$(\lambda - \tilde{\lambda}^k)^T \left\{ \sum_{j=1}^m \mathbf{A}_j \tilde{x}_j^k - \sum_{j=2}^m \mathbf{A}_j (\tilde{x}_j^k - x_j^k) + \frac{1}{s\beta} (\tilde{\lambda}^k - \lambda^k) \right\} \geq 0. \quad (14)$$

结合式(8)、(12)、(13) 和矩阵 $\hat{\mathbf{Q}}$ 的定义可得

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T \{F(\tilde{w}^k) + \hat{\mathbf{Q}}(\tilde{w}^k - w^k)\} \geq 0. \quad (15)$$

于是由 $\hat{\mathbf{Q}}$ 和 \mathbf{Q} 的定义可得

$$(\tilde{v}^k - v)^T \mathbf{Q}(\tilde{v}^k - v^k) = (\tilde{w}^k - w)^T \hat{\mathbf{Q}}(\tilde{w}^k - w^k). \quad (16)$$

将式(15) 代入到式(16) 中即可得证引理 1.

引理 2 令 $\tilde{w}^k (w^k = (\tilde{x}_1^k, \tilde{x}_2^k, \dots, \tilde{x}_m^k, \tilde{\lambda}^k))$ 由步骤 1 产生,并给定 $w^k = (x_1^k, x_2^k, \dots, x_m^k, \lambda^k)$, 则有

$$(\tilde{v}^k - v^*)^T \mathbf{Q}(v^k - \tilde{v}^k) \geq 0, \quad (17)$$

其中矩阵 \mathbf{Q} 按 1.2 中被定义.

证明 令 $v = v^*, w = w^*$, 将其代入到式(9) 中得

$$(\tilde{v}^k - v^*)^T \mathbf{Q}(\tilde{v}^k - v^k) \geq \theta(u^*) - \theta(\tilde{u}^k) + (w^* - \tilde{w}^k)^T F(\tilde{w}^k). \quad (18)$$

由 F 的单调性可得 $\theta(u^*) - \theta(\tilde{u}^k) + (w^* - \tilde{w}^k)^T F(\tilde{w}^k) \geq 0$. 再由式(17) 和式(18) 知引理 2 得证.

基于引理 1 和引理 2, 本文给出如下的定理 1 和定理 2.

定理 1 令 $w^{k+1} (w^{k+1} = (x_1^{k+1}, x_2^{k+1}, \dots, x_m^{k+1}, \lambda^{k+1}))$ 由新算法产生, $\tilde{w}^k (w^k = (\tilde{x}_1^k, \tilde{x}_2^k, \dots, \tilde{x}_m^k, \tilde{\lambda}^k))$ 由步骤 1 产生, 并给定 $w^k = (x_1^k, x_2^k, \dots, x_m^k, \lambda^k)$, 则有:

$$\|v^{k+1} - v^*\|_{\mathbf{H}}^2 \leq \|v^k - v^*\|_{\mathbf{H}}^2 - \|v^k - \tilde{v}^k\|_{\mathbf{G}}^2. \quad (19)$$

证明 由步骤 2 和引理 2 得:

$$\begin{aligned} \|v^k - v^*\|_{\mathbf{H}}^2 - \|v^{k+1} - v^*\|_{\mathbf{H}}^2 &= \|v^k - v^*\|_{\mathbf{H}}^2 - \|v^k - v^* - (v^k - v^{k+1})\|_{\mathbf{H}}^2 = \\ &= \|v^k - v^*\|_{\mathbf{H}}^2 - \|v^k - v^* - \mathbf{M}(v^k - \tilde{v}^k)\|_{\mathbf{H}}^2 = 2(v^k - v^*)^T \mathbf{H} \mathbf{M}(v^k - \tilde{v}^k) - \|\mathbf{M}(v^k - \tilde{v}^k)\|_{\mathbf{H}}^2 \geq \\ &= 2(v^k - \tilde{v}^k)^T \mathbf{Q}(v^k - \tilde{v}^k) - \|\mathbf{M}(v^k - \tilde{v}^k)\|_{\mathbf{H}}^2 = \|v^k - \tilde{v}^k\|_{\mathbf{Q} + \mathbf{Q}^T}^2 - \|v^k - \tilde{v}^k\|_{\mathbf{M}^T \mathbf{H} \mathbf{M}}^2 = \end{aligned}$$

$$\|v^k - \tilde{v}^k\|_{Q+Q^T-M^T H M}^2 = \|v^k - \tilde{v}^k\|_G^2,$$

因此定理 1 得证.

定理 2 由新算法生成的序列 $\{w^k\}$ 收敛于原问题的解点.

证明 由式(19)可知 $\{w^k\}$ 是有界的,因此 $\{w^k\}$ 至少有一个聚点. 设 w^∞ 是 $\{w^k\}$ 的某个聚点,则存在子列 $\{w^{k_j}\}$ 使 $\lim_{j \rightarrow \infty} w^{k_j} = w^\infty$. 由式(19)还可得 $\lim_{k \rightarrow \infty} \|\tilde{w}^k - w^k\|_G^2 = 0$. 将 $\{w^{k_j}\}$ 代入式(19)中,并对式两端关于 $\{w^k\}$ 取极限得 $\theta(u) - \theta(u^\infty) + (w - w^\infty)^T F(w^\infty) \geq 0$. 由此可知 w^∞ 是 $MVI(\Omega, F, f)$ 的一个解点,即序列 $\{w^k\}$ 能够收敛到原问题的解点.

3 数值实验

数值实验采用的 PC 机的配置为: 1.80 GHz CPU, 8 GB 内存, Windows10 家庭中文版操作系统. 编程软件为 MatlabR2018a. 测试模型为如下二次线性规划模型:

$$\min_x \sum_{i=1}^m g_i(x_i), \text{ s. t. } \sum_{i=1}^m A_i x_i = c, \quad (20)$$

其中 $g_i(x_i) = \frac{1}{2} x_i^T H_i x_i + x_i^T q_i \in \mathbf{R}^{m_i} \rightarrow \mathbf{R}$, $q \in \mathbf{R}^{m_i}$, $A_i \in \mathbf{R}^{n \times m_i}$, $H_i \in \mathbf{R}^{m_i \times m_i}$, $c \in \mathbf{R}^n$. 利用新算法求解上述模型的基本方法如下:

1) 模型构建: 矩阵 $A_i \in \mathbf{R}^{n \times m_i}$, $H_i \in \mathbf{R}^{m_i \times m_i}$ 的元素是独立同分布的高斯随机变量; $c_i = \sum_{i=1}^m A_i x_i^*$, $q_i = -H_i x_i^* + A_i^T \lambda^*$, $x_i^* = \text{randn}(m_i, 1)$, $\lambda^* = \text{randn}(n, 1)$.

2) 选取算法参数: $s = 1.2$, $r = 3s$, β 根据不同算法和问题设置进行人工调优.

3) 算法终止条件: $\text{relchg}(k) = \max \left\{ \frac{\|x_1^k - x_1^{k-1}\|}{x_1^{k-1}}, \dots, \frac{\|x_m^k - x_m^{k-1}\|}{x_m^{k-1}}, \frac{\|\lambda^k - \lambda^{k-1}\|}{\lambda^{k-1}} \right\} < 10^{-14}$.

4) 算法的 KKT 违反度: $KKT(k) := \max \{KKT_\lambda(k), KKT_1(k), KKT_2(k), \dots, KKT_m(k)\}$, 其中 $KKT_\lambda(k) = \|A_1 x_1^k + \dots + A_m x_m^k - c\|$, $KKT_i(k) = \|H_i x_i^k + q_i - A_i^T \lambda^k\|$.

为了验证新算法的收敛速度是否得到提高, 分别采用新算法(PPADMMR)和 PPADMM 算法^[7] 求解模型(式(20)). 测试分为 4 组 (n, m_i) , 每组运行 10 次, 实验结果见表 1. 由表 1 可知: 在 KKT 违反度方面, PPADMMR 算法和 PPADMM 算法相近; 在迭代次数方面, PPADMMR 算法的迭代次数比 PPADMM 算法的迭代次数减少 6% ~ 13%. 由此可知, PPADMMR 算法的收敛速度优于 PPADMM 算法, 该结果符合本文的理论分析.

表 1 PPADMMR 算法和 PPADMM 算法的测试结果

(n, m_i)	PPADMMR		PPADMM	
	KKT 违反度	迭代次数	KKT 违反度	迭代次数
(100, 100)	5.28×10^{-12}	193.2	8.16×10^{-12}	221.4
(100, 50)	1.01×10^{-11}	207.4	1.43×10^{-11}	238.5
(150, 50)	7.41×10^{-12}	197.3	1.34×10^{-11}	225.6
(200, 50)	7.55×10^{-12}	163.3	1.22×10^{-11}	174.0

为了更直观地观察 PPADMMR 算法与 PPADMM 算法在不同条件数 (n, m_i) 下, $\text{Cond}(H_i)$ 中矩阵 H_i 的收敛效果, 本文给出了两种算法的迭代次数随 $\text{Cond}(H_i)$ 的变化情况, 如图 1 所示. 由图 1 可以看出, 在 $\text{Cond}(H_i)$ 较高的情况下, PPADMMR 算法的迭代次数总是少于 PPADMM 算法的迭代次数, 且这种性能优势随着 $\text{Cond}(H_i)$ 的变化始终保持稳定. 由此可知, PPADMMR 算法的收敛效果优于 PPADMM 算法.

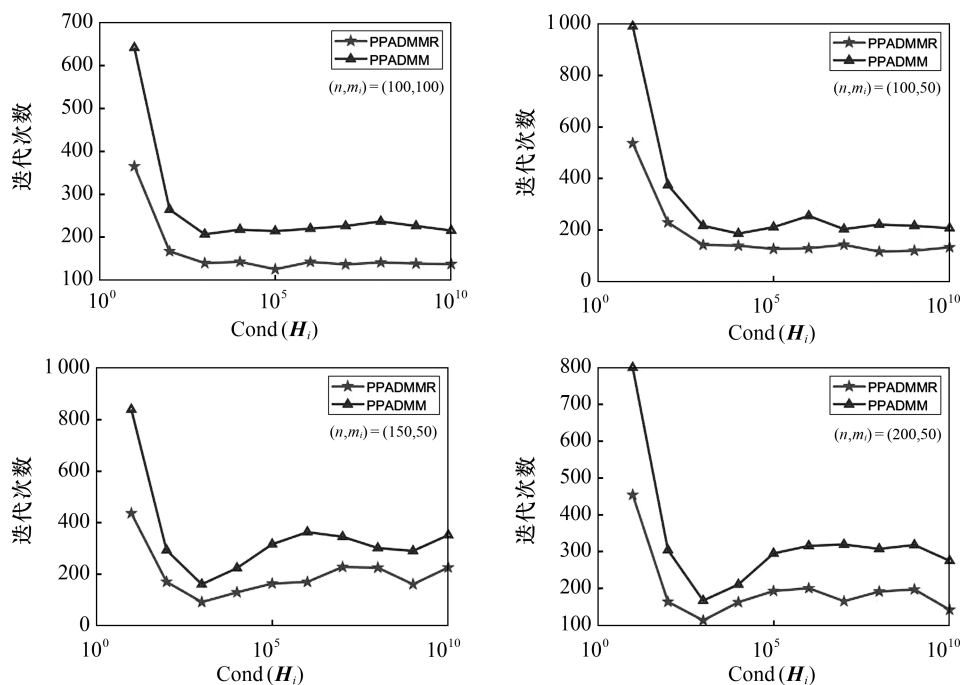


图 1 PPADMMR 算法和 PPADMM 算法在不同条件数下的迭代次数

4 结论

上述研究表明,本文提出的 PPADMMR 算法的收敛性能显著优于 PPADMM 算法,因此该算法可用于解决多块线性约束可分凸优化问题,同时可为研究快速 ADMM 算法提供参考.由于本文将 PPADMMR 算法的矫正步步长固定为 1,因此算法的收敛速度会受到固定步长的约束.在今后的研究中,我们将在该算法中引入自适应步长技术,以更进一步提高该算法的性能.

参考文献:

- [1] BOYD S, PARIKH N, CHU E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. *Found Trends Mach Learn*, 2010,3(1):1-122.
- [2] BOSE N K, BOO K J. High-resolution image reconstruction with multisensors[J]. *Int J Imag Syst Tech*, 1998,9(4):294-304.
- [3] RIBSHIRANI R, SAUNDERS M, ROSSET S, et al. Sparsity and smoothness via the fused lasso[J]. *J Roy Stat Soc*, 2010,67(1):91-108.
- [4] YE Y Y, CHEN C H, HE B S, et al. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent[J]. *Math Program*, 2016,155(1/2):57-79.
- [5] 陈彩华. 多块交替方向乘子法不收敛反例的几点注记[J]. *运筹学学报*, 2019,23(3):135-140.
- [6] HE B S, TAO M, YUAN X M. A splitting method for separable convex programming[J]. *Ima J Numer Anal*, 2013(1):394-426.
- [7] HOU L S, HE H J, YANG J F. A partially parallel splitting method for multiple-block separable convex programming with applications to robust PCA[J]. *Comput Optim Appl*, 2016,63(1):273-303.