

求解一类双线性规划问题的数值算法

答辩人: 胡雨宽

指导老师: 殷俊锋

同济大学数学科学学院

2019.6.4



提纲

- ① 引言
- ② 预备知识
- ③ 最优性条件
- ④ 算法设计
- ⑤ 收敛性分析
- ⑥ 数值实验
- ⑦ 总结与展望

提纲

- 1 引言
- 2 预备知识
- 3 最优性条件
- 4 算法设计
- 5 收敛性分析
- 6 数值实验
- 7 总结与展望

问题背景

最优运输问题:

- 建立有效比较概率分布的几何工具 (GAN).
- 极小化将一概率分布"运输"到另一概率分布的"花费".

Monge 问题

令 $P, Q \subset \mathbb{R}^d$. f 为 P 上的概率密度函数, g 为 Q 上的概率密度函数.
 $c: P \times Q \rightarrow [0, \infty)$ 为连续函数. Monge 问题

$$\inf_T M(T) := \int_P c(p, T(p)) f(p) \, dp, \quad \text{s.t. } T\#f = g.$$

其中 $T\#f = g$ 定义为

$$\int_B g(q) \, dq = \int_{T^{-1}(B)} f(p) \, dp, \quad \forall B \subset Q,$$

c 可以看做是成本泛函.

问题背景 (续)

当 f, g 为离散概率分布函数:

- $P = \{p_1, \dots, p_m\}, Q = \{q_1, \dots, q_n\}$.
- $\{f_1, \dots, f_m\}, \{g_1, \dots, g_n\}$.

$$\begin{aligned} \min_{a_{ij}} \quad & \sum_{i,j} c_{ij} a_{ij} \\ \text{s.t.} \quad & \sum_j a_{ij} = f_i, \quad i = 1, \dots, m, \\ & \sum_i a_{ij} = g_j, \quad j = 1, \dots, n, \\ & a_{ij} \geq 0, \quad i = 1, \dots, m, j = 1, \dots, n, \end{aligned} \quad (\text{LP})$$

这里 $c_{ij} := c(p_i, q_j)$.

问题陈述

$$\begin{aligned}
 \min_{X, Y} \quad & \sum_{i \neq j} \frac{x_{ij}}{|r_i - r_j|} + \sum_{i \neq k} \frac{y_{ik}}{|r_i - r_k|} + \sum_{i, j, k: j \neq k} \frac{x_{ij} y_{ik}}{|r_j - r_k|} \\
 \text{s.t.} \quad & \sum_j x_{ij} = \rho_i, \quad i = 1, \dots, n, \\
 & \sum_i x_{ij} = \rho_j, \quad j = 1, \dots, n, \\
 & \sum_k y_{ik} = \rho_i, \quad i = 1, \dots, n, \\
 & \sum_i y_{ik} = \rho_k, \quad i = 1, \dots, n, \\
 & x_{ij}, y_{ik} \geq 0, \quad i, j, k = 1, \dots, n, \\
 & x_{ii}, y_{ii} = 0, \quad i = 1, \dots, n,
 \end{aligned} \tag{1}$$

其中 $X, Y \in \mathbb{R}^{n \times n}$, $r = (r_1, \dots, r_n)^T$, $\rho = (\rho_1, \dots, \rho_n)^T \in \mathbb{R}_+^n$. 将 $\{|r_i - r_j|\}$ 储存于 $R = (r_{ij}) \in \mathbb{R}^{n \times n}$, 其中

$$r_{ij} = \begin{cases} 1/|r_i - r_j|, & i \neq j, \\ 0, & i = j. \end{cases}$$

问题陈述

$$\begin{aligned}
 \min_{X, Y} \quad & \sum_{i \neq j} \frac{x_{ij}}{|r_i - r_j|} + \sum_{i \neq k} \frac{y_{ik}}{|r_i - r_k|} + \sum_{i, j, k: j \neq k} \frac{x_{ij} y_{ik}}{|r_j - r_k|} \\
 \text{s.t.} \quad & \sum_j x_{ij} = \rho_i, \quad i = 1, \dots, n, \\
 & \sum_i x_{ij} = \rho_j, \quad j = 1, \dots, n, \\
 & \sum_k y_{ik} = \rho_i, \quad i = 1, \dots, n, \\
 & \sum_i y_{ik} = \rho_k, \quad i = 1, \dots, n, \\
 & x_{ij}, y_{ik} \geq 0, \quad i, j, k = 1, \dots, n, \\
 & x_{ii}, y_{ii} = 0, \quad i = 1, \dots, n.
 \end{aligned} \tag{1}$$

问题(1)有两个运输工具, 它的目标:

- 极小化两运输工具 X, Y 的运输费用 ((1)目标函数的前两项);
- 当 r_j, r_k 较近, X 应尽量不该将 i 往 j 运输, Y 应尽量不该将 i 往 k 运输 ((1)目标函数的最后一项).

研究现状

角度 1 可分离约束的双线性规划.

$$\begin{array}{ll} \min_{x,y} & x^T Q y + c^T x + d^T y \\ \text{s.t.} & x \in \mathcal{X}, y \in \mathcal{Y}. \end{array}$$

广泛的应用 (Konno '71等):

双矩阵博弈、动态马尔科夫分配问题、多商品网络流问题、某些动态生产问题、二次凹极小化问题、直线距离选址问题、三维分配问题以及一些互补规划问题.

已有方法:

- 割平面法 (Ritter '66等).
- 分支定界法 (Falk '73等).
- 割平面法速度慢; 分支定界法验证难.

研究现状 (续)

角度 2 非凸二次规划.

$$\begin{aligned} \min_{X, Y} \quad & (\text{vec}(X) + \text{vec}(Y))^T \text{vec}(R) + \text{vec}(X)^T (R \otimes I) \text{vec}(Y) \\ \text{s.t.} \quad & (\mathbf{1}^T \otimes I) \text{vec}(X) = (I \otimes \mathbf{1}^T) \text{vec}(X) = \rho, \\ & (\mathbf{1}^T \otimes I) \text{vec}(Y) = (I \otimes \mathbf{1}^T) \text{vec}(Y) = \rho, \\ & \text{vec}(I)^T \text{vec}(X) = 0, \text{vec}(X) \geq 0, \\ & \text{vec}(I)^T \text{vec}(Y) = 0, \text{vec}(Y) \geq 0. \end{aligned}$$

已有方法 (Nocedal/Wright '06):

- 逐步二次规划 (SQP).
- 积极集法 (Active-set Methods).
- 内点法 (Interior-point Methods).
- ...

研究现状 (续)

角度 2 非凸二次规划.

$$\begin{aligned} \min_{X, Y} \quad & (\text{vec}(X) + \text{vec}(Y))^T \text{vec}(R) + \text{vec}(X)^T (R \otimes I) \text{vec}(Y) \\ \text{s.t.} \quad & (\mathbf{1}^T \otimes I) \text{vec}(X) = (I \otimes \mathbf{1}^T) \text{vec}(X) = \rho, \\ & (\mathbf{1}^T \otimes I) \text{vec}(Y) = (I \otimes \mathbf{1}^T) \text{vec}(Y) = \rho, \\ & \text{vec}(I)^T \text{vec}(X) = 0, \text{vec}(X) \geq 0, \\ & \text{vec}(I)^T \text{vec}(Y) = 0, \text{vec}(Y) \geq 0. \end{aligned}$$

已有方法 (Nocedal/Wright '06):

- 逐步二次规划 (SQP).
- 积极集法 (Active-set Methods).
- 内点法 (Interior-point Methods).
- ...

以上方法均以列向量为求解对象 \Rightarrow 计算量问题.

提纲

- 1 引言
- 2 预备知识
- 3 最优性条件
- 4 算法设计
- 5 收敛性分析
- 6 数值实验
- 7 总结与展望

预备知识

标准矩阵内积

对 $\forall A, B \in \mathbb{R}^{n \times n}$, 定义

$$\langle A, B \rangle := \text{tr}(A^T B) = \sum_{i,j} a_{ij} b_{ij}.$$

求导规则:

$$\frac{d}{dA} \langle A, B \rangle = B, \quad \frac{d}{dB} \langle A, B \rangle = A.$$

预备知识

标准矩阵内积

对 $\forall A, B \in \mathbb{R}^{n \times n}$, 定义

$$\langle A, B \rangle := \text{tr}(A^T B) = \sum_{i,j} a_{ij} b_{ij}.$$

求导规则:

$$\frac{d}{dA} \langle A, B \rangle = B, \quad \frac{d}{dB} \langle A, B \rangle = A.$$

问题(1)即可化为矩阵形式:

$$\begin{aligned} \min_{X, Y} \quad & \langle R, X \rangle + \langle R, Y \rangle + \langle Y, XR \rangle \\ \text{s.t.} \quad & X\mathbf{1} = \rho, X^T\mathbf{1} = \rho, \text{tr}(X) = 0, X \geq 0, \\ & Y\mathbf{1} = \rho, Y^T\mathbf{1} = \rho, \text{tr}(Y) = 0, Y \geq 0, \end{aligned} \tag{2}$$

其中 $\mathbf{1}$ 为全 1 向量.

预备知识 (续)

实验表明 ...

假设

问题(2)的所有稳定点 (X, Y) 均满足 $X = Y$.

预备知识 (续)

实验表明 ...

假设

问题(2)的所有稳定点 (X, Y) 均满足 $X = Y$.

⇒ 简化问题:

$$\begin{array}{ll} \min_X & 2\langle X, R \rangle + \langle X, XR \rangle \\ \text{s.t.} & X\mathbf{1} = \rho, X^T\mathbf{1} = \rho, \text{tr}(X) = 0, X \geq 0. \end{array} \quad (3)$$

预备知识 (续)

实验表明 ...

假设

问题(2)的所有稳定点 (X, Y) 均满足 $X = Y$.

⇒ 简化问题:

$$\begin{aligned} \min_X \quad & 2\langle X, R \rangle + \langle X, XR \rangle \\ \text{s.t.} \quad & X\mathbf{1} = \rho, X^T\mathbf{1} = \rho, \text{tr}(X) = 0, X \geq 0. \end{aligned} \quad (3)$$

引入分裂变量 $Z \in \mathbb{R}^{n \times n}$, 进一步得到等价的

$$\begin{aligned} \min_{X, Z} \quad & f(X, Z) \triangleq 2\langle X, R \rangle + \langle Z, XR \rangle \\ \text{s.t.} \quad & X\mathbf{1} = \rho, \text{tr}(X) = 0, \\ & Z^T\mathbf{1} = \rho, Z \geq 0, \\ & X = Z. \end{aligned} \quad (4)$$

之后重点求解问题(4).

提纲

- 1 引言
- 2 预备知识
- 3 最优性条件
- 4 算法设计
- 5 收敛性分析
- 6 数值实验
- 7 总结与展望

最优性条件

$$\begin{aligned}\mathcal{L}_0(X, Z, \lambda_1, \lambda_2, \mu, \Phi, \Omega) = & f(X, Z) - \langle \lambda_1, X\mathbf{1} - \rho \rangle - \langle \lambda_2, Z^T\mathbf{1} - \rho \rangle \\ & - \mu \text{tr}(X) - \langle \Omega, Z \rangle - \langle \Phi, X - Z \rangle.\end{aligned}$$

最优性条件

$$\begin{aligned}\mathcal{L}_0(X, Z, \lambda_1, \lambda_2, \mu, \Phi, \Omega) = & f(X, Z) - \langle \lambda_1, X\mathbf{1} - \rho \rangle - \langle \lambda_2, Z^T\mathbf{1} - \rho \rangle \\ & - \mu \text{tr}(X) - \langle \Omega, Z \rangle - \langle \Phi, X - Z \rangle.\end{aligned}$$

问题(4)的 KKT 条件

若 (X^*, Z^*) 为问题(4)的解, 则存在拉格朗日乘子 $\mu^* \in \mathbb{R}, \lambda_1^*, \lambda_2^* \in \mathbb{R}^n, \Phi^*, 0 \leq \Omega^* \in \mathbb{R}^{n \times n}$, 使得

$$\left\{ \begin{array}{l} \nabla_X \mathcal{L}_0 = 2R + Z^* R - \lambda_1^* \mathbf{1}^T - \Phi^* - \mu^* I = 0, \\ \nabla_Z \mathcal{L}_0 = X^* R - \mathbf{1} (\lambda_2^*)^T + \Phi^* - \Omega^* = 0, \\ X^* \mathbf{1} = \rho, \text{tr}(X^*) = 0, \\ (Z^*)^T \mathbf{1} = \rho, Z^* \geq 0, \\ \Omega^* \geq 0, \\ \Omega^* \circ Z^* = 0. \end{array} \right\}$$

稳定性条件

或对偶可行性条件

原始可行性条件,

互补松弛条件.

(5)

这里“ \circ ”表示 Hadamard 积.

提纲

- 1 引言
- 2 预备知识
- 3 最优性条件
- 4 算法设计
- 5 收敛性分析
- 6 数值实验
- 7 总结与展望

ADMM 算法简介

ADMM 算法 (介绍可见 [Boyd, et al. '10](#)) 求解问题一般形式:

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \theta(x) := \sum_{i=1}^n \theta_i(x_i) + \ell(x_1, \dots, x_n) \\ \text{s.t.} \quad & \sum_{i=1}^n A_i x_i = b, \end{aligned} \tag{6}$$

这里 $\theta_i: \mathbb{R}^{d_i} \mapsto (-\infty, +\infty]$, $\ell: \mathbb{R}^d \mapsto (-\infty, +\infty]$; $x_i \in \mathbb{R}^{d_i}$; $A_i \in \mathbb{R}^{m \times d_i}$, $b \in \mathbb{R}^m$. (6) 的增广拉格朗日函数为

$$\begin{aligned} \mathcal{L}_A(x_1, \dots, x_n, \bar{\mu}) = & \sum_{i=1}^n \theta_i(x_i) + \ell(x_1, \dots, x_n) \\ & - \bar{\mu}^T \left(\sum_{i=1}^n A_i x_i - b \right) + \frac{\bar{\beta}}{2} \left\| \sum_{i=1}^n A_i x_i - b \right\|^2, \end{aligned}$$

其中 $\bar{\mu} \in \mathbb{R}^m$ 为拉格朗日乘子, $\bar{\beta} > 0$ 为惩罚因子.

ADMM 算法简介 (续)

算法 1 n 块 ADMM 算法

输入: $x_2^0, \dots, x_n^0, \bar{\mu}^0, k := 0$.

输出: $x_1^k, x_2^k, \dots, x_n^k, \bar{\mu}^k$.

1: **while** 收敛性测试未通过 **do**

2: $x_1^{k+1} := \arg \min_{x_1 \in \mathbb{R}^{d_1}} \mathcal{L}_A(x_1, \dots, x_n^k, \bar{\mu}^k);$

3: $\dots\dots\dots$

4: $x_n^{k+1} := \arg \min_{x_n \in \mathbb{R}^{d_n}} \mathcal{L}_A(x_1^{k+1}, \dots, x_n, \bar{\mu}^k);$

5: $\bar{\mu}^{k+1} := \bar{\mu}^k - \bar{\beta} \left(\sum_{i=1}^n A_i x_i^{k+1} - b \right);$

6: $k := k + 1;$

7: **end while**

ADMM 算法简介 (续)

现有工作简介:

- 两块凸可分问题 (Boyd, et al. '10, He/Yuan '12等): $n = 2, \ell = 0$, θ_1, θ_2 是凸函数.
- 多块凸可分问题 (He/Tao/Yuan '12, Cai/Han/Yuan '17等): $n \geq 3, \ell = 0$, $\theta_1, \dots, \theta_n$ 是凸函数. 可能发散 (Chen, et al. '16). \Rightarrow 无假设的困境.

ADMM 算法简介 (续)

现有工作简介:

- 两块凸可分问题 (Boyd, et al. '10, He/Yuan '12等): $n = 2, \ell = 0$, θ_1, θ_2 是凸函数.
- 多块凸可分问题 (He/Tao/Yuan '12, Cai/Han/Yuan '17等): $n \geq 3, \ell = 0$, $\theta_1, \dots, \theta_n$ 是凸函数. 可能发散 (Chen, et al. '16). \Rightarrow 无假设的困境.
- 凸不可分问题 (Hong, et al. '14, Chen, et al. '19). 即使 $n = 2, \theta(\cdot)$ 凸, 仍然开放 (Hong/Luo/Razaviyayn '16).

ADMM 算法简介 (续)

现有工作简介:

- 两块凸可分问题 (Boyd, et al. '10, He/Yuan '12等): $n = 2, \ell = 0$, θ_1, θ_2 是凸函数.
- 多块凸可分问题 (He/Tao/Yuan '12, Cai/Han/Yuan '17等): $n \geq 3, \ell = 0$, $\theta_1, \dots, \theta_n$ 是凸函数. 可能发散 (Chen, et al. '16). \Rightarrow 无假设的困境.
- 凸不可分问题 (Hong, et al. '14, Chen, et al. '19). 即使 $n = 2, \theta(\cdot)$ 凸, 仍然开放 (Hong/Luo/Razaviyayn '16).
- 非凸问题. 理论缺乏, 应用广泛.
 - Wen, et al. '13等: 需要强加无法验证的条件.
 - Hong/Luo/Razaviyayn '16: 只需要 θ_i 's 和 ℓ 满足合理的正则条件, 且 $\bar{\beta}$ 充分大.

ADMM 算法简介 (续)

$$\mathcal{L}_A(X, Z, \Phi) = f(X, Z) - \langle \Phi, X - Z \rangle + \frac{\beta}{2} \|X - Z\|_F^2.$$

框架 2 求解问题 (4) 的 ADMM 算法框架

输入: $X^0, Z^0, \Phi^0, \beta^0, k := 0$.

输出: X^k, Z^k, Φ^k .

1: **while** 收敛性测试未通过 **do**

2: $X^{k+1} = \arg \min_{X: X\mathbf{1}=\rho, \text{tr}(X)=0} \mathcal{L}_A(X, Z^k, \Phi^k);$

3: $Z^{k+1} = \arg \min_{Z: Z^T\mathbf{1}=\rho, Z \geq 0} \mathcal{L}_A(X^{k+1}, Z, \Phi^k);$

4: 更新 Φ^k 得到 $\Phi^{k+1};$

5: 如有需要, 更新 β^k 得到 $\beta^{k+1};$

6: $k := k + 1;$

7: **end while**

子问题的求解 - X 子问题

$$\begin{aligned} \min_X \quad & 2\langle X, R \rangle + \langle Z, XR \rangle - \langle \Phi, X - Z \rangle + \frac{\beta}{2} \|X - Z\|_F^2 \\ \text{s.t.} \quad & X\mathbf{1} = \rho, \quad \text{tr}(X) = 0. \end{aligned} \quad (7)$$

带等式约束的凸二次规划

子问题的求解 - X 子问题

$$\begin{aligned} \min_X \quad & 2\langle X, R \rangle + \langle Z, XR \rangle - \langle \Phi, X - Z \rangle + \frac{\beta}{2} \|X - Z\|_F^2 \\ \text{s.t.} \quad & X\mathbf{1} = \rho, \quad \text{tr}(X) = 0. \end{aligned} \quad (7)$$

带等式约束的凸二次规划 \Rightarrow 拉格朗日乘数法.

$$\beta X + (2R + ZR - \lambda_1 \mathbf{1}^T - \mu I - \Phi - \beta Z) = 0, \quad (8a)$$

$$-\text{tr}(X) = 0, \quad (8b)$$

$$-(X\mathbf{1} - \rho) = 0. \quad (8c)$$

子问题的求解 - X 子问题 (续)

由(8a),

$$X^+ = -\frac{1}{\beta}(2R + ZR - \lambda_1 \mathbf{1}^T - \mu I - \Phi - \beta Z). \quad (9)$$

由(9)和(8b),(8c), 令

$$\begin{aligned} M_1 &= 2R\mathbf{1} + ZR\mathbf{1} - \Phi\mathbf{1} - \beta Z\mathbf{1} + \beta\rho, \\ m_1 &= 2\text{tr}(R) + \text{tr}(ZR) - \text{tr}(\Phi) - \beta\text{tr}(Z). \end{aligned}$$

乘子为

$$\mu = \frac{1}{n-1} \left(-\frac{1}{n} \mathbf{1}^T M_1 + m_1 \right), \quad \lambda_1 = \frac{1}{n} (M_1 - \mathbf{1}\mu). \quad (10)$$

(10)和(9)给出 X 子问题(7)的解.

子问题的求解 - Z 子问题

$$\begin{aligned} \min_Z \quad & \langle Z, X^+ R \rangle - \langle \Phi, X^+ - Z \rangle + \frac{\beta}{2} \|X^+ - Z\|_F^2 \\ \text{s.t.} \quad & Z^T \mathbf{1} = \rho, \quad Z \geq 0. \end{aligned} \tag{11}$$

同样是凸二次规划, 但是有不等式约束.

子问题的求解 - Z 子问题

$$\begin{aligned} \min_Z \quad & \langle Z, X^+ R \rangle - \langle \Phi, X^+ - Z \rangle + \frac{\beta}{2} \|X^+ - Z\|_F^2 \\ \text{s.t.} \quad & Z^T \mathbf{1} = \rho, \quad Z \geq 0. \end{aligned} \quad (11)$$

同样是凸二次规划, 但是有不等式约束. quadprog()?

子问题的求解 - Z 子问题

$$\begin{aligned} \min_Z \quad & \langle Z, X^+ R \rangle - \langle \Phi, X^+ - Z \rangle + \frac{\beta}{2} \|X^+ - Z\|_F^2 \\ \text{s.t.} \quad & Z^T \mathbf{1} = \rho, \quad Z \geq 0. \end{aligned} \quad (11)$$

同样是凸二次规划, 但是有不等式约束. `quadprog()`?

忽略非负约束, 类似 X 子问题求超平面 $Z^T \mathbf{1} = \rho$ 上的一点 \tilde{Z} :

$$\lambda_2 = \frac{1}{n} \left[R(X^+)^T \mathbf{1} + \Phi^T \mathbf{1} - \beta (X^+)^T \mathbf{1} + \beta \rho \right],$$

$$\tilde{Z} = \frac{1}{\beta} (X^+ R + \Phi - \beta X^+ - \mathbf{1} \lambda_2^T).$$

子问题的求解 - Z 子问题 (续)

问题(11)的等价形式:

$$\begin{array}{ll} \min_Z & \|Z - \tilde{Z}\|_F^2 \\ \text{s.t.} & Z^T \mathbf{1} = \rho, \quad Z \geq 0. \end{array}$$

子问题的求解 - Z 子问题 (续)

问题(11)的等价形式:

$$\begin{aligned} \min_Z \quad & \|Z - \tilde{Z}\|_F^2 \\ \text{s.t.} \quad & Z^T \mathbf{1} = \rho, \quad Z \geq 0. \end{aligned}$$

分块

$$Z = [z_1, \dots, z_n], \quad \tilde{Z} = [\tilde{z}_1, \dots, \tilde{z}_n].$$

$$\begin{aligned} \Leftrightarrow \min_{z_1, \dots, z_n} \quad & \sum_{j=1}^n \|z_j - \tilde{z}_j\|^2 \\ \text{s.t.} \quad & \mathbf{1}^T z_j = \rho_j, \quad z_j \geq 0, \quad j = 1, \dots, n. \end{aligned} \tag{12}$$

$\text{quadprog}() \Rightarrow Z^+$.

子问题的求解 - Z 子问题 (续)

问题(11)的等价形式:

$$\begin{aligned} \min_Z \quad & \|Z - \tilde{Z}\|_F^2 \\ \text{s.t.} \quad & Z^T \mathbf{1} = \rho, \quad Z \geq 0. \end{aligned}$$

分块

$$Z = [z_1, \dots, z_n], \quad \tilde{Z} = [\tilde{z}_1, \dots, \tilde{z}_n].$$

$$\begin{aligned} \Leftrightarrow \min_{z_1, \dots, z_n} \quad & \sum_{j=1}^n \|z_j - \tilde{z}_j\|^2 \\ \text{s.t.} \quad & \mathbf{1}^T z_j = \rho_j, \quad z_j \geq 0, \quad j = 1, \dots, n. \end{aligned} \quad (12)$$

$\text{quadprog}() \Rightarrow Z^+$.

注

若未恰当引入分裂变量 Z , 子问题求解难度不言而喻.

拉格朗日乘子的更新

对一般的等式约束非线性优化问题

$$\begin{array}{ll}\min_x & \bar{f}(x) \\ \text{s.t.} & c_i(x) = 0, \quad i \in \mathcal{E},\end{array}$$

增广拉格朗日函数法 (γ^k 是惩罚因子):

$$\lambda_i^{k+1} = \lambda_i^k - \gamma^k c_i(x^k), \quad \forall i \in \mathcal{E}.$$

拉格朗日乘子的更新

对一般的等式约束非线性优化问题

$$\begin{array}{ll}\min_x & \bar{f}(x) \\ \text{s.t.} & c_i(x) = 0, \quad i \in \mathcal{E},\end{array}$$

增广拉格朗日函数法 (γ^k 是惩罚因子):

$$\lambda_i^{k+1} = \lambda_i^k - \gamma^k c_i(x^k), \quad \forall i \in \mathcal{E}.$$

因此我们的拉格朗日乘子更新策略可选为

$$\Phi^+ = \Phi - \beta(X^+ - Z^+). \quad (13)$$

若带松弛因子 $\alpha > 0$, 则

$$\Phi^+ = \Phi - \alpha\beta(X^+ - Z^+). \quad (14)$$

停机准则与 KKT 违反度

对 X 子问题:

$$X^{k+1} = \arg \min_{X: X\mathbf{1}=\rho, \text{tr}(X)=0} \mathcal{L}_A(X, Z^k, \Phi^k)$$

$\Rightarrow \exists \lambda_1^{k+1}, \mu^{k+1}$ 使得

$$2R + Z^k R - \Phi^k + \beta(X^{k+1} - Z^k) - \lambda_1^{k+1} \mathbf{1}^T - \mu^{k+1} I = 0, \quad (15a)$$

$$X^{k+1} \mathbf{1} = \rho, \text{tr}(X^{k+1}) = 0. \quad (15b)$$

停机准则与 KKT 违反度

对 X 子问题:

$$X^{k+1} = \arg \min_{X: X\mathbf{1}=\rho, \text{tr}(X)=0} \mathcal{L}_A(X, Z^k, \Phi^k)$$

$\Rightarrow \exists \lambda_1^{k+1}, \mu^{k+1}$ 使得

$$2R + Z^k R - \Phi^k + \beta(X^{k+1} - Z^k) - \lambda_1^{k+1} \mathbf{1}^T - \mu^{k+1} I = 0, \quad (15a)$$

$$X^{k+1} \mathbf{1} = \rho, \text{tr}(X^{k+1}) = 0. \quad (15b)$$

等式(15a),(15b)分别对应于问题(4)KKT 条件(5)中的

$$2R + Z^* R - \lambda_1^* \mathbf{1}^T - \mu^* I - \Phi^* = 0, \quad (16a)$$

$$X^* \mathbf{1} = \rho, \text{tr}(X^*) = 0. \quad (16b)$$

停机准则与 KKT 违反度 (续)

- 使用更新策略(13):

$$\begin{aligned} & 2R + Z^k R - \Phi^k + \beta(X^{k+1} - Z^k) - \lambda_1^{k+1} \mathbf{1}^T - \mu^{k+1} I \\ & = 2R + Z^{k+1} R - \Phi^{k+1} - \lambda_1^{k+1} \mathbf{1}^T - \mu^{k+1} I + (Z^{k+1} - Z^k)(\beta I - R), \end{aligned}$$

- 使用更新策略(14):

$$\begin{aligned} & 2R + Z^k R - \Phi^k + \beta(X^{k+1} - Z^k) - \lambda_1^{k+1} \mathbf{1}^T - \mu^{k+1} I \\ & = 2R + Z^{k+1} R - \Phi^{k+1} - \lambda_1^{k+1} \mathbf{1}^T - \mu^{k+1} I \\ & \quad + (Z^{k+1} - Z^k)(\beta I - R) + (1 - \alpha)\beta(X^{k+1} - Z^{k+1}), \end{aligned}$$

停机准则与 KKT 违反度 (续)

- 使用更新策略(13):

$$\begin{aligned} & 2R + Z^k R - \Phi^k + \beta(X^{k+1} - Z^k) - \lambda_1^{k+1} \mathbf{1}^T - \mu^{k+1} I \\ & = 2R + Z^{k+1} R - \Phi^{k+1} - \lambda_1^{k+1} \mathbf{1}^T - \mu^{k+1} I + (Z^{k+1} - Z^k)(\beta I - R), \end{aligned}$$

- 使用更新策略(14):

$$\begin{aligned} & 2R + Z^k R - \Phi^k + \beta(X^{k+1} - Z^k) - \lambda_1^{k+1} \mathbf{1}^T - \mu^{k+1} I \\ & = 2R + Z^{k+1} R - \Phi^{k+1} - \lambda_1^{k+1} \mathbf{1}^T - \mu^{k+1} I \\ & \quad + (Z^{k+1} - Z^k)(\beta I - R) + (1 - \alpha)\beta(X^{k+1} - Z^{k+1}), \end{aligned}$$

\Rightarrow 关注 $(Z^{k+1} - Z^k)(\beta I - R)$, $X^{k+1} - Z^{k+1}$

停机准则与 KKT 违反度 (续)

对 Z 子问题:

$$Z^{k+1} = \arg \min_{Z: Z^T \mathbf{1} = \rho, Z \geq 0} \mathcal{L}_A(X^{k+1}, Z, \Phi^k)$$

$\Rightarrow \exists \lambda_2^{k+1}, \Omega^{k+1} \geq 0$ 使得

$$X^{k+1} R - \Phi^k - \beta(X^{k+1} - Z^{k+1}) - \mathbf{1} \left(\lambda_1^{k+1} \right)^T - \Omega^{k+1} = 0, \quad (17a)$$

$$\left(Z^{k+1} \right)^T \mathbf{1} = \rho, Z^{k+1} \geq 0, \quad (17b)$$

$$\Omega^{k+1} \geq 0, \quad (17c)$$

$$\Omega^{k+1} \circ Z^{k+1} = 0. \quad (17d)$$

停机准则与 KKT 违反度 (续)

$$X^{k+1}R - \Phi^k - \beta(X^{k+1} - Z^{k+1}) - \mathbf{1} \left(\lambda_1^{k+1} \right)^T - \Omega^{k+1} = 0, \quad (17a)$$

$$\left(Z^{k+1} \right)^T \mathbf{1} = \rho, Z^{k+1} \geq 0, \quad (17b)$$

$$\Omega^{k+1} \geq 0, \quad (17c)$$

$$\Omega^{k+1} \circ Z^{k+1} = 0. \quad (17d)$$

等式(17a)-(17d)分别对应于问题(4)KKT 条件(5)中的

$$X^*R - \mathbf{1} (\lambda_2^*)^T + \Phi^* - \Omega^* = 0, \quad (18a)$$

$$(Z^*)^T \mathbf{1} = \rho, Z^* \geq 0, \quad (18b)$$

$$\Omega^* \geq 0, \quad (18c)$$

$$\Omega^* \circ Z^* = 0. \quad (18d)$$

停机准则与 KKT 违反度 (续)

- 使用更新策略(13):

$$\begin{aligned} & X^{k+1}R - \Phi^k - \beta(X^{k+1} - Z^{k+1}) - \mathbf{1} \left(\lambda_2^{k+1} \right)^T - \Omega^{k+1} \\ &= X^{k+1}R - \Phi^{k+1} - \mathbf{1} \left(\lambda_2^{k+1} \right)^T - \Omega^{k+1}. \end{aligned}$$

- 使用更新策略(14):

$$\begin{aligned} & X^{k+1}R - \Phi^k - \beta(X^{k+1} - Z^{k+1}) - \mathbf{1} \left(\lambda_2^{k+1} \right)^T - \Omega^{k+1} \\ &= X^{k+1}R - \Phi^{k+1} - \mathbf{1} \left(\lambda_2^{k+1} \right)^T - \Omega^{k+1} - (1 - \alpha)\beta(X^{k+1} - Z^{k+1}). \end{aligned}$$

停机准则与 KKT 违反度 (续)

- 使用更新策略(13):

$$\begin{aligned} & X^{k+1}R - \Phi^k - \beta(X^{k+1} - Z^{k+1}) - \mathbf{1} \left(\lambda_2^{k+1} \right)^T - \Omega^{k+1} \\ &= X^{k+1}R - \Phi^{k+1} - \mathbf{1} \left(\lambda_2^{k+1} \right)^T - \Omega^{k+1}. \end{aligned}$$

- 使用更新策略(14):

$$\begin{aligned} & X^{k+1}R - \Phi^k - \beta(X^{k+1} - Z^{k+1}) - \mathbf{1} \left(\lambda_2^{k+1} \right)^T - \Omega^{k+1} \\ &= X^{k+1}R - \Phi^{k+1} - \mathbf{1} \left(\lambda_2^{k+1} \right)^T - \Omega^{k+1} - (1 - \alpha)\beta(X^{k+1} - Z^{k+1}). \end{aligned}$$

\Rightarrow 关注 $X^{k+1} - Z^{k+1}$

停机准则与 KKT 违反度 (续)

KKT 违反度:

$$\begin{aligned} t^{k+1} &\triangleq \|X^{k+1} - Z^{k+1}\|_{\infty} && \text{原始残差,} \\ s^{k+1} &\triangleq \|(Z^{k+1} - Z^k)(\beta I - R)\|_{\infty} && \text{对偶残差.} \end{aligned} \tag{19}$$

停机准则与 KKT 违反度 (续)

KKT 违反度:

$$\begin{aligned} t^{k+1} &\triangleq \|X^{k+1} - Z^{k+1}\|_{\infty} && \text{原始残差,} \\ s^{k+1} &\triangleq \|(Z^{k+1} - Z^k)(\beta I - R)\|_{\infty} && \text{对偶残差.} \end{aligned} \quad (19)$$

停机准则:

情形 1 t^{k+1}, s^{k+1} 都足够小;

情形 2 对某个 $p^{k+1} \in (0, 1)$, $p^{k+1}s^{k+1} + (1 - p^{k+1})t^{k+1}$ 足够小. 其中 p^{k+1} 反映了我们主观上对两类残差所赋的权值.

停机准则与 KKT 违反度 (续)

KKT 违反度:

$$\begin{aligned} t^{k+1} &\triangleq \|X^{k+1} - Z^{k+1}\|_{\infty} && \text{原始残差,} \\ s^{k+1} &\triangleq \|(Z^{k+1} - Z^k)(\beta I - R)\|_{\infty} && \text{对偶残差.} \end{aligned} \quad (19)$$

停机准则:

情形 1 t^{k+1}, s^{k+1} 都足够小;

情形 2 对某个 $p^{k+1} \in (0, 1)$, $p^{k+1}s^{k+1} + (1 - p^{k+1})t^{k+1}$ 足够小. 其中 p^{k+1} 反映了我们主观上对两类残差所赋的权值.

我们使用

$$E^{k+1} = (1 - p^{k+1})t^{k+1} + p^{k+1}s^{k+1} \quad (20)$$

作为 **KKT** 违反度.

完整算法

算法 3 求解问题 (4) 的 ADMM 算法

输入: $X^0, Z^0, \Phi^0, \beta^0, \epsilon, k := 0, s^0 := 1, t^0 := 1, \alpha > 0$ (默认值为1),
 $p^0 \in (0, 1)$.

输出: X^k, Z^k, Φ^k

- 1: $E^k = (1 - p^k)t^k + p^k s^k$;
- 2: **while** $E^k > \epsilon$ **do**
- 3: 由公式(10),(9)计算 X^{k+1} ;
- 4: 使用 MATLAB 内置函数 `quadprog()` 求解列子问题(12)得到 Z^{k+1} ;
- 5: $\Phi^{k+1} = \Phi^k - \alpha\beta^k(X^{k+1} - Z^{k+1})$;
- 6: 由公式(19)计算 t^{k+1}, s^{k+1} ;
- 7: 更新 β^k 得到 β^{k+1} ;
- 8: 更新 p^k 得到 p^{k+1} ;
- 9: 由公式(20)得到 E^{k+1} ;
- 10: $k := k + 1$;
- 11: **end while**

提纲

- 1 引言
- 2 预备知识
- 3 最优性条件
- 4 算法设计
- 5 收敛性分析**
- 6 数值实验
- 7 总结与展望

收敛性分析

定理 (算法3的收敛性定理)

假设算法3每一步, Z 子问题均精确求解, 且产生的迭代序列 $\{X^k\}, \{Z^k\}, \{\Phi^k\}$ 分别收敛到 X^*, Z^*, Φ^* , 满足 $X^* = Z^*$. 则 (X^*, Z^*, Φ^*) 为问题(4)的稳定点.

收敛性分析

定理 (算法3的收敛性定理)

假设算法3每一步, Z 子问题均精确求解, 且产生的迭代序列 $\{X^k\}, \{Z^k\}, \{\Phi^k\}$ 分别收敛到 X^*, Z^*, Φ^* , 满足 $X^* = Z^*$. 则 (X^*, Z^*, Φ^*) 为问题(4)的稳定点.

关键点:

- ① $X^* \in \arg \min_{X \mathbf{1} = \rho, \text{tr}(X) = 0} f(X, Z^*) - \langle \Phi^*, X - Z^* \rangle.$
- ② $Z^* \in \arg \min_{Z^T \mathbf{1} = \rho, Z \geq 0} f(X^*, Z) - \langle \Phi^*, X^* - Z \rangle.$

提纲

- 1 引言
- 2 预备知识
- 3 最优性条件
- 4 算法设计
- 5 收敛性分析
- 6 数值实验**
- 7 总结与展望

数值实验

使用 MATLAB 内置函数 `randn()` 和 `abs()` 生成随机问题. 小型问题以 $n = 3, 4, 5$ 各一问题为代表, 大型问题以 $n = 20, 30, 40$ 各一问题为代表.

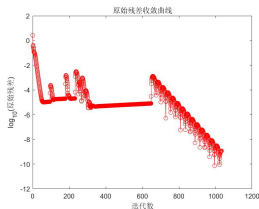
小型问题中的常量取法:

$$\alpha = 1, \quad \beta^k \equiv 10^3, \quad \epsilon = 10^{-8}, \quad p^k \equiv 0.5.$$

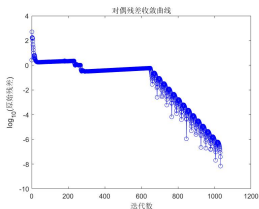
大型问题中的常量取法:

$$\alpha = 1, \quad \beta^k \equiv 10^4, \quad \epsilon = 10^{-6}, \quad p^k \equiv 0.5.$$

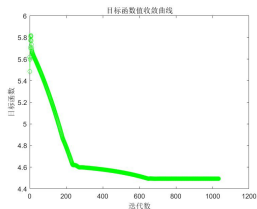
数值实验 - 收敛曲线



(a) 原始残差



(b) 对偶残差



(c) 目标函数值

图 1: $n = 4$ 收敛曲线

数值实验 - 松弛因子 α

$\alpha = 0.1, 0.2, \dots, 1.0$.

表 1: $n = 3$, 不同的松弛因子

α	$n = 3$			
	迭代数	所耗时间 (s)	KKT 违反度	目标值
0.1	452	0.0072	3.40×10^{-9}	1.1722
0.2	491	0.0079	8.45×10^{-9}	1.1722
0.3	510	0.0088	7.07×10^{-9}	1.1722
0.4	517	0.0098	2.86×10^{-9}	1.1722
0.5	503	0.0080	7.45×10^{-9}	1.1722
0.6	534	0.0103	7.79×10^{-9}	1.1722
0.7	538	0.0093	5.69×10^{-9}	1.1722
0.8	525	0.0082	3.24×10^{-9}	1.1722
0.9	548	0.0105	8.97×10^{-9}	1.1722
1.0	560	0.0097	8.02×10^{-9}	1.1722

数值实验 - 松弛因子 α $\alpha = 0.1, 0.2, \dots, 1.0.$ 表 2: $n = 30$, 不同的松弛因子

α	$n = 30$			
	迭代数	所耗时间 (s)	KKT 违反度	目标值
0.1	117886	74.7150	9.74×10^{-7}	26.2298
0.2	69365	42.6729	8.60×10^{-7}	25.8871
0.3	70989	43.2812	8.88×10^{-7}	25.8828
0.4	65533	40.0219	6.46×10^{-7}	25.8757
0.5	122029	76.4424	9.28×10^{-7}	25.6165
0.6	112336	82.8559	9.63×10^{-7}	25.6165
0.7	112002	70.1342	8.81×10^{-7}	25.6165
0.8	109121	67.5131	7.99×10^{-7}	25.6165
0.9	113655	70.1853	8.83×10^{-7}	25.6165
1.0	207788	129.1006	6.00×10^{-7}	24.9461

数值实验 - 惩罚因子 β

$$\begin{cases} \beta = 10^2, 10^3, 10^4, 10^5, & n = 3, 4, 5, \\ \beta = 10^3, 10^4, 10^5, & n = 20, \\ \beta = 10^4, 10^5, & n = 30. \end{cases}$$

表 3: $n = 3$, 不同的惩罚因子

β	$n = 3$			
	迭代数	所耗时间 (s)	KKT 违反度	目标值
100	224	0.0043	5.46×10^{-9}	1.1722
1000	560	0.0095	8.02×10^{-9}	1.1722
10000	3998	0.0647	4.92×10^{-9}	1.1722
100000	38377	0.5818	7.68×10^{-9}	1.1722

数值实验 - 惩罚因子 β

$$\begin{cases} \beta = 10^2, 10^3, 10^4, 10^5, & n = 3, 4, 5, \\ \beta = 10^3, 10^4, 10^5, & n = 20, \\ \beta = 10^4, 10^5, & n = 30. \end{cases}$$

表 4: $n = 5$, 不同的惩罚因子

β	$n = 5$			
	迭代数	所耗时间 (s)	KKT 违反度	目标值
100	969	0.0465	5.00×10^{-9}	2.7741
1000	860	0.0384	9.54×10^{-9}	2.7741
10000	3857	0.1698	4.64×10^{-9}	2.7741
100000	33470	1.4344	9.47×10^{-9}	2.7741

数值实验 - 惩罚因子 β

$$\begin{cases} \beta = 10^2, 10^3, 10^4, 10^5, & n = 3, 4, 5, \\ \beta = 10^3, 10^4, 10^5, & n = 20, \\ \beta = 10^4, 10^5, & n = 30. \end{cases}$$

表 5: $n = 20$, 不同的惩罚因子

β	$n = 20$			
	迭代数	所耗时间 (s)	KKT 违反度	目标值
1000	12122	4.1522	9.95×10^{-7}	9.8692
10000	79052	26.7695	5.02×10^{-7}	9.8692
100000	753429	255.0805	9.98×10^{-7}	9.8692

数值实验 - 惩罚因子 β

$$\begin{cases} \beta = 10^2, 10^3, 10^4, 10^5, & n = 3, 4, 5, \\ \beta = 10^3, 10^4, 10^5, & n = 20, \\ \beta = 10^4, 10^5, & n = 30. \end{cases}$$

表 6: $n = 30$, 不同的惩罚因子

β	$n = 30$			
	迭代数	所耗时间 (s)	KKT 违反度	目标值
10000	207788	133.6154	6.00×10^{-7}	24.9461
100000	2053406	1327.5987	7.08×10^{-7}	24.9542

数值实验 - 惩罚因子 β

$$\begin{cases} \beta = 10^2, 10^3, 10^4, 10^5, & n = 3, 4, 5, \\ \beta = 10^3, 10^4, 10^5, & n = 20, \\ \beta = 10^4, 10^5, & n = 30. \end{cases}$$

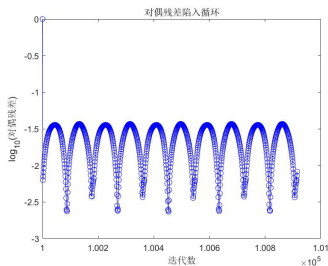
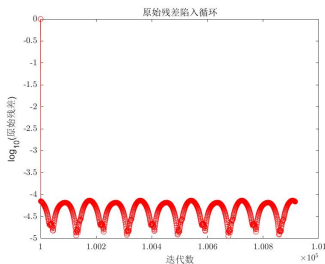


图 2: 残差陷入循环. 左: 原始残差; 右: 对偶残差

数值实验 – 测试问题

- 目的: 表明算法3的确可以达到全局最优解.
- 构造方式: 给定一数对 $(p, q) : p \neq q, p, q \in \{1, 2, \dots, n\}$, 定义 R 为

$$r_{ij} = \begin{cases} 1, & i = p, j = q \text{ 或 } i = q, j = p, \\ 0, & \text{其它.} \end{cases}$$

$$\rho := \mathbf{1}.$$

$$\begin{aligned} \min_X \quad & 2x_{pq} + 2 \sum_i x_{ip} x_{iq} \\ \text{s.t.} \quad & X\mathbf{1} = \rho, X^T\mathbf{1} = \rho, X \geq 0, \text{tr}(X) = 0. \end{aligned}$$

数值实验 - 测试问题 (续)

$$\begin{aligned} \min_X \quad & 2x_{pq} + 2 \sum_i x_{ip} x_{iq} \\ \text{s.t.} \quad & X\mathbf{1} = \rho, X^T\mathbf{1} = \rho, X \geq 0, \text{tr}(X) = 0. \end{aligned}$$

- 一个天然下界: 0.
- 可行点

$$X = \begin{pmatrix} & p & & q & \\ & 1 & & 0 & \\ & 0 & & 1 & \\ p & 0 & & 0 & \\ & * & \vdots & * & \vdots & * \\ q & 0 & & 0 & \\ & 0 & & 0 & \\ & 0 & & 0 & \end{pmatrix}$$

处的函数值为 0.

数值实验 - 测试问题 (续)

$$\begin{aligned} \min_X \quad & 2x_{pq} + 2 \sum_i x_{ip} x_{iq} \\ \text{s.t.} \quad & X\mathbf{1} = \rho, X^T\mathbf{1} = \rho, X \geq 0, \text{tr}(X) = 0. \end{aligned}$$

$n = 5, 10, 15, 20, (p, q) = (3, 4), \alpha = 1, \beta = 10^3, \epsilon = 10^{-8}.$

表 7: 测试问题, $n = 5, 10, 15, 20$

n	迭代数	所耗时间 (s)	KKT 违反度	目标值
5	3187	0.1536	3.00×10^{-9}	-1.44×10^{-11}
10	1447	0.1766	1.65×10^{-9}	-4.94×10^{-15}
15	2243	0.3284	2.30×10^{-9}	6.54×10^{-13}
20	2030	0.3299	4.53×10^{-9}	-8.07×10^{-13}

数值实验 - 与求解非凸二次规划的算法比较

- MATLAB 内置函数 `fmincon()`, 调用算法 `'sqp'`, `'active-set'` 求解(3)和(2), 设置停机准则 `ConstraintTolerance` 为上文默认值. 在使用算法3求解 $n = 50, 60, 70, 80$ 的问题, 设置惩罚因子

$$\begin{cases} \beta = 10^4, & n = 50, 60, \\ \beta = 2 \times 10^4, & n = 70, 80. \end{cases}$$

数值实验 - 与求解非凸二次规划的算法比较 (续)

表 8: 使用现成算法

$n = 3$	所耗时间 (s)	目标值	$n = 4$	所耗时间 (s)	目标值
算法3	0.0097	1.1722	算法3	0.0375	4.4934
SQP	0.0033	1.1722	SQP	0.0039	4.4934
$n = 5$	所耗时间 (s)	目标值	$n = 20$	所耗时间 (s)	目标值
算法3	0.0438	2.7741	算法3	28.2271	9.8692
SQP	0.0048	2.7741	SQP	0.6388	9.9096
$n = 30$	所耗时间 (s)	目标值	$n = 40$	所耗时间 (s)	目标值
算法3	129.1006	24.9461	算法3	177.2822	19.0920
SQP	12.0369	26.5668	SQP	42.4949	21.6995

数值实验 - 与求解非凸二次规划的算法比较 (续)

表 8: 使用现成算法

$n = 50$	所耗时间 (s)	目标值	$n = 60$	所耗时间 (s)	目标值
算法3	195.8105	42.8754	算法3	507.7331	30.3166
SQP	132.2582	46.8143	SQP	472.3746	33.0913
$n = 70$	所耗时间 (s)	目标值	$n = 80$	所耗时间 (s)	目标值
算法3	1440.9720	30.2839	算法3	2275.5156	40.6460
SQP	2174.3117	30.1568	SQP	3589.5316	43.1043

数值实验 - 与求解非凸二次规划的算法比较 (续)

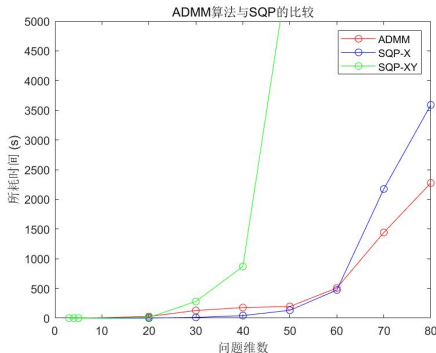


图 3: 算法3和 SQP 运行时间对比

结论: 对于小型问题, 积极集法和 SQP 最为便捷; 对于大型问题, 算法3更具优势. 算法3的缺点在于, 我们需要根据问题的维度不断重新设置惩罚因子 β .

提纲

- 1 引言
- 2 预备知识
- 3 最优性条件
- 4 算法设计
- 5 收敛性分析
- 6 数值实验
- 7 总结与展望**

总结与展望

总结:

- 针对问题(4)特殊结构设计了 ADMM 算法.
- 证明了一定条件下 ADMM 算法3收敛到问题(4)的稳定点.
- 在随机生成的问题上进行数值实验; 讨论了多个人工给定因子对算法效果的影响; 与求解非凸二次规划的算法做了比较, 得出算法3在大型问题上更具优势的结论.

展望:

- 证明假设.
- 设计求解 Z 子问题(11)更加高效、精确的算法.
- 证明算法3在更弱条件下的收敛性.
- 对松弛因子、惩罚因子做进一步的讨论.
- 讨论初始点对算法的影响.
- 与现有的双线性规划算法进行比较.
- 子问题的并行求解.

感谢聆听!

huyukuan2015@tongji.edu.cn