

# 线性规划对偶内点法<sup>\*</sup>

魏紫奎      吴 力

(中国科学院计算中心,北京,100080)

**摘 要** 本文叙述在仿射变换下对偶内点方法的数值实现及应用 拉朗日乘子的高阶估计提高最优解的精度,同时给出一些数值试验结果。

**关键词** 线性规划,对偶内点法,拉格朗日乘子,高阶估计

## 1. 引言

近年来求解线性规划问题的各种不同形式的多项式时间算法的相继出现,使得对线性规划问题的研究耳目一新。有趣的是人们发现这些算法产生的搜索方向是由两个基本向量的线性组合:目标函数在变换空间中的最速下降方向和“中心”方向,而且它们都可以由对数障碍函数方法推得,只是选取的参数不同而已。在实践上人们发现应用仿射变换下的线性规划问题的原始及对偶内点法于求解实际问题是十分有效的,尽管它们在理论上不具有多项式时间的性质。它们的主要优点是可以直接应用线性规划的标准形式,对于其不等式约束的对偶形式,可以放松对搜索方向计算的精度要求,因为它只要求迭代的点列在可行域内。

本文讨论在仿射变换下对偶内点法的计算步骤及其实现,在适当的假设下其拉格朗日乘子在最优解的邻域内有高阶估计值,因而当迭代点列进入最优解的邻域时,可以应用这一估计来确定问题的解,从而减少迭代次数,提高解的精度。我们进行的数值测试的结果表明应用这一估计值是非常有效的。

## 2. 算法的描述

首先考虑如下标准形式的线性规划问题:

$$\text{Min } C^T X \quad (2.1)$$

$$\text{s. t. } AX = b$$

$$X \geq 0 \quad (2.2)$$

其中  $A \in R^{m \times n}$ ,  $b \in R^m$ ,  $C \in R^n$ ,  $A, b, c$  是给定的,  $X \in R^n$  是未知向量。假设矩阵  $A$  的行秩为  $m$ , 且其约束(2.2)的可行区域是非空的。当可行区域的严格内点集合非空时,可以应用仿射变换定义求解问题(2.1)–(2.2)的算法,它的实际运行的效果是很好的,然而应用它解 Klee–Minty 问题时,如果初始点选取得太靠近可行顶点,要用指数次幂的迭代次数才

\* 本文1991年4月17日收到。

国家自然科学基金资助项目。

能收敛到最优解。与之相似的另一方法是在仿射变换下的对偶内点法,现考虑问题:

$$\text{Max } z = b^T y \quad (2.3)$$

$$\text{s. t. } A^T y \leq C \quad (2.4)$$

令  $S = \{y: A^T y \leq C, y \in R^m\}$ , 对给定的问题(2.1)–(2.2), 我们直接考虑其对偶形式(2.3)–(2.4), 假设  $S$  是非空的, 其内点的集合  $S^\circ \neq \emptyset$ , 即存在  $y \in S$ , 使得

$$v = C - A^T y > 0 \quad (2.5)$$

成立。显然, 我们可以定义内点算法的一般形式, 即对任意的  $y \in S^\circ$ , 如果存在一搜索方向  $u$  和一适当的正常数  $\alpha$ , 使对  $\bar{y} = y + \alpha u$  恒有  $z(y) < z(\bar{y})$  (即  $b^T u > 0$ ), 且  $\bar{v} = C - A^T \bar{y} > 0$ , 如果给出如何确定搜索方向  $u$  的方法, 那么由一个内点搜索至另一个内点的算法就完全确定了。

现在我们定义在仿射变换下求解问题(2.3)–(2.4)的内点算法。

算法 A:

设  $y^{(0)}$  是  $S$  的一已知严格内点,  $\varepsilon_1, \varepsilon_2$  是给定的充分小的允许误差, 令  $K = 0$ , 执行以下计算步骤:

(1) 定义

$$D_K = \text{diag}(1/V_1^{(k)}, \dots, 1/V_n^{(k)})$$

其中

$$V^{(k)} = \bar{C} - A^T y^{(k)}$$

(2) 计算

$$u^{(k)} = (A D_K^2 A^T)^{-1} b \quad (2.6)$$

并生成向量

$$P^{(k)} = A^T u^{(k)} \quad (2.7)$$

若  $\text{Max}_{1 \leq k \leq n} \{P^{(k)}\} < \varepsilon_1$ , 终止计算, 问题(2.3)–(2.4)的目标函数值  $Z$  为无界; 否则转(3)。

(3) 选取适当的步长  $\alpha_k$ , 并产生新的迭代点,

$$y^{(k+1)} = Y^{(k)} + \alpha_k u^{(k)}$$

使得

$$V^{(k+1)} = v^{(k)} + \alpha_k P^{(k)} > 0$$

若

$$\frac{|\alpha_k \cdot b^T y^{(k)}|}{\text{Max}(1, |b^T y^{(k+1)}|)} < \varepsilon_2 \quad (2.8)$$

则终止计算,  $y^{(k+1)}$  是问题(2.3)–(2.4)的近似最优解; 否则返回(1)。

本算法尚未被证明具有多项式时间性质, 但亦没有一反例说明其迭代次数为指数次的其计算复杂性尚需探索。

在上述算法的定义中, 如令

$$\bar{X} = D_K^2 A^T u^{(k)} \quad (2.9)$$

容易验证

$$Ax=b \tag{2.10}$$

因而,如果问题(2.1)–(2.2)是非退化的,且其解为唯一时,由(2.9)所定义的  $\bar{x}$  亦是问题(2.1)–(2.2)的一近似最优解,但在一般情况下,(2.10)式仍成立,而  $\bar{x}$  的所有分量的非负性是不能保证的,于是求得问题(2.3)–(2.4)的近似最优解,并不能得到相应问题(2.1)–(2.2)的近似最优解。然而尚若把  $\bar{x}$  作为问题(2.1)–(2.2)的近似最优解,则有

$$\begin{aligned} |C^T - b^T y^{(k+1)}| &= |(v^{(k+1)} + A^T y^{(k+1)})^T \bar{x} - b^T y^{(k+1)}| \\ &= |\bar{x} v^{(k+1)} + (A \bar{x})^T y^{(k+1)} - b^T y^{(k+1)}| \\ &= |\bar{x}^T v^{(k+1)}| \end{aligned} \tag{2.11}$$

(2.1)式恰是最优性条件中的互补松弛性

$$|\bar{x}^T v^{(k+1)}| \leq \sum_{i=1}^n |\bar{x}_i \cdot v_i^{(k+1)}| < n\epsilon \tag{2.12}$$

其中  $\epsilon$  为给定的允许误差,

$$\max_{1 \leq i \leq n} \{|\bar{x}_i v_i^{(k+1)}|\} < \epsilon \tag{2.13}$$

在一般情况下以(2.8)式为终止规则就可以了,而(2.12)或(2.13)式亦可以同时作为计算的终止规则。

### 3. 算法的实现

在上述所定义的算法 A 中,最重要的是确定一初始的严格内点和计算搜索方向  $u$ ,本节简要地叙述计算初始严格内点和搜索方向  $u$  的方法。

为计算问题(2.3)–(2.4)的初始可行解,我们定义以下算法:

设  $\epsilon$  是给定的允许误差,令

$$y^{(0)} = \frac{\|C\|_2}{\|A^T\|_2} \cdot b$$

计算

$$\bar{v}^{(0)} = C - A^T y^{(0)}$$

如果  $\bar{v}^{(0)} > \epsilon e$  ( $e \in R^n$ , 且其分量均为 1), 则  $y^{(0)}$  为所求的严格可行内点; 否则令

$$\delta = \min_{1 \leq i \leq n} \{\bar{v}_i^{(0)}\}$$

容易验证

$$V^{(0)} = C - A^T y^{(0)} - Z_0 e > 0,$$

$$V_{n+1}^{(0)} = -2\delta > 0,$$

其中  $Z_0 = 2\delta$ , 令  $K=0$ , 执行以下计算步骤:

(1) 定义

$$D_k = \text{diag} \{1/V_1^{(k)}, \dots, 1/V_n^{(k)}\}$$

$$A_k = A D_k$$

(2) 计算

$$\bar{u}^{(k)} = (A_k A_k^T)^{-1} A_k D_k e \quad (3.1)$$

和

$$g^{(k)} = e - A^T \bar{u}^{(k)}$$

(3). 选取最小比率  $\lambda_k = \min \{V_i^{(k)} / g_i^{(k)}, g_i^{(k)} > 0\}$  (3.2)

$$1 \leq i \leq n$$

若  $\lambda_k \leq V_{n+1}^{(k)}$ , 则转(4); 否则令  $\alpha_k = V_{n+1}^{(k)}$ , 按下式计算  $y^{(k+1)}, V^{(k+1)}$ :

$$y^{(k+1)} = y^{(k)} - \alpha_k \bar{u}^{(k)} \quad (3.3)$$

$$V^{(k+1)} = V^{(k)} - \alpha_k g^{(k)} \quad (3.4)$$

$y^{(k+1)}$  是问题(2.3)–(2.4)的严格可行内点。

(4) 按公式(3.3)–(3.4)产生新的迭代点  $y^{(k+1)}, V^{(k+1)}$ , 并计算

$$Z_{(k+1)} = Z_k + \alpha_k$$

$$V_{n+1}^{(k+1)} = V_{n+1}^{(k)} - \alpha_k$$

其中  $\alpha_k = \alpha \cdot \lambda_k$ ,  $\alpha \in (0, 1)$ , 若  $\alpha_k > \epsilon$  令  $k = k + 1$ , 返回(1); 否则终止计算, 且当  $|Z_{k+1}| > \epsilon$  问题(2.3)–(2.4)是不相容的; 当  $|Z_{k+1}| < \epsilon$  时,  $y^{(k+1)}$  是问题(2.3)–(2.4)的一可行解。

大量的数值试验表明当问题(2.3)–(2.4)的可行域的内点集合非空时, 算法产生的点列皆终止于步骤(3)。本算法的详细叙述见[8]。

无论是算法 A 还是计算初始严格可行内点的算法, 在每次迭代中都必须求解一系数矩阵为  $A_k \cdot A_k^T$  (对称正定) 的线性代数方程组来确定搜索方向  $u^k$  和  $\bar{u}^k$ 。对此有许多可供选择的方法, 我们反对讨论数值计算较稳定的 Cholesky 分解方法, 其中  $A_k$  可以是大型的稀疏结构形式。

由表达式(2.6)和(3.1)可见, 在每次迭代中只是  $D_k$  发生了变化, 这说明当 A 是稀疏矩阵时, 矩阵  $A D_k^2 A^T$  的稀疏性是与  $D_k$  无关的, 而与矩阵  $A A^T$  的稀疏性相同的, 因而为了得到矩阵  $A_k \cdot A_k^T$  的 Cholesky 分解的下三角矩阵  $L_k$  的稀疏结构形式, 我们只要研究  $A A^T$  的 Cholesky 分解的下三角矩阵 L 的稀疏结构就可以了。

要使  $A \cdot A^T$  的 Cholesky 分解下三角矩阵 L 有较好的稀疏性(即非零元素的个数最少), 必须对矩阵 A 的行的顺序进行重新排列。在理论上必定存在一置换矩阵 P, 使得

$PAA^T P^T$  的 Cholesky 分解的下三角矩阵  $L$  的非零元素的个数为最少,而要找到这样的置换矩阵  $P$  的代价是很昂贵的,实际上只是采用局部极小化方向确定  $P$ ,使得  $PAA^T P^T$  的 Cholesky 分解的下三角元素有较少的非零元素的个数。

在找到置换矩阵  $P$  后,最重要的问题就是要确定  $PAA^T P^T$  的 Cholesky 分解的下三角矩阵中非零元素的位置和个数。如果可以预先知道这些信息,就可以预先分配其存储空间,而在数值计算时只需按照已排好的非零元素的位置进行计算。这一过程称之为形式分解。它不需要真正计算下三角矩阵  $L$  的非零元素的值,只是根据  $PAA^T P^T$  的非零无结构的信息,模拟其 Cholesky 分解的过程,确定  $L$  的非零元素的位置和个数。

由于  $PA D_k^2 A^T P^T$  和  $PAA^T P^T$  有相同的非零结构形式,因而在每次迭代中,只需根据  $D_k$  的值计算  $PA D_k^2 A^T P^T$  的各非零元素的值,然后由形式分解的信息,进行数值分解确定下三角矩阵的各非零元素的值,最后求解两个三角方程组就得到所求的搜索方向  $u^{(k)}$  (或  $\bar{u}^{(k)}$ )。

有关本算法详细叙述见[10]

#### 4. 拉格朗日乘子的估计及应用

以上我们讨论了求解问题(2.3)–(2.4)的算法 A 及其实现。更多的实际问题则是直接计算问题(2.1)–(2.2)的解,显然一般说算法 A 并不能得到问题(2.1)–(2.2)的解,然而在适当的假设下也可以做到这一点,这就是本节所要叙述的问题(2.3)–(2.4)的拉格朗日乘子的估计并应用它求解问题(2.1)–(2.2)。

不失一般性,设  $y^*$  是问题(2.3)–(2.4)的最优解,且存在  $m$  个线性独立的向量  $a_1, a_2, \dots, a_m$  使得

$$V_i^* = C_i - a_i^T y^* = 0, i=1, 2, \dots, m \tag{4.1}$$

$$V_i^* = C_i - a_i^T y^* > 0, i=m+1, \dots, n \tag{4.2}$$

$N(y^*, \delta)$  表示以  $y^*$  为中心,  $\delta$  为半径的欧氏超球,于是我们有以下的定理。

定理 4.1, 设问题(2.3)–(2.4)的可行域是非空的,且  $S^0 \neq \emptyset$ , 其最优解  $y^*$  满足(4.1)–(4.2), 则必存在一充分小的  $\delta > 0$ , 使得当  $\bar{y} \in N(y^*, \delta) \cap S^0$  时有

$$\bar{X} = X^* + O(\delta^2)$$

其中  $X^*$  是问题(2.1)–(2.2)的最优解,  $\bar{X}$  是由(2.9)式定义的, 如果令

$$\bar{u} = (AD^k A^T)^{-1} b$$

$$\bar{X} = D^k A^T \bar{u}$$

并且定理 4.1 的假设条件成立, 则以下的估计式

$$\bar{X} = X^* + O(\delta^K) \tag{4.4}$$

成立。(其中,  $K \geq 1, K$  为整数)。

如果对  $\delta$  能够给出估计值, 就可以应用估计式(4.3)或(4.4)。由上节的叙述可知为求得搜索方向  $u$ , 通常对  $AD^2 A^T$  进行 Cholesky 分解, 即

$$AD^2 A^T = L \cdot L^T \tag{4.5}$$

其中  $L$  是下三角矩阵, 令

$$l = \min_{1 \leq i \leq m} l_i \tag{4.6}$$

其中  $l_{ii}$  是下三角矩阵  $L$  的对角线元素, 以  $A_2^T$  表示 (4.2) 式中的行向量形成的矩阵,  $u$  表示  $A_2^T$  最大奇异值, 对任意的  $\bar{y} \in S^0$ , 令

$$\sigma = \min_{1 \leq i \leq m} \{\bar{V}_i\} \quad (4.7)$$

其中  $\bar{V}$  是由 (2.5) 定义的, 于是有以下定理

定理 4.2, 设定理 4.1 的假条件满足, 如果对给定的  $\bar{Y} \in S^0$ ,  $\mu/1 \cdot \sigma < \frac{1}{\sqrt{2}}$ , 则估计式 (4.3) 亦成立。

然而, 在实际计算中要给出  $\mu$  的值是很困难有, 我们可以应用它的上界估计值。

推论 4.3, 设定理 4.1 的假设条件满足, 如果对给定的  $\bar{y} \in S^0$ , 有  $m(n-m)M^2/l^2\sigma^2 < \frac{1}{2}$ , 则估计式 (4.3) 成立。

$$\text{其中 } M = \max_{1 \leq i \leq m} \max_{m+1 \leq j \leq n} \{|a_{ij}|\}$$

以上结论的证明详见 [9]

根据以上的理论分析, 我们可以对算法 A 进行适当修正, 使之既能求得问题 (2.3) — (2.4) 的解, 又可以求得问题 (2.1) — (2.2) 的解

算法 A':

设  $y^{(0)}$  是  $S$  的严格可行内点,  $\varepsilon$  是给定的允许误差, 令  $K=0$ , 执行以下计算步骤:

(1) 定义

$$D_K = \text{diag}\{1/V_1^{(K)}, \dots, 1/V_m^{(K)}\}$$

其中

$$V^{(K)} = C - A^T y^{(K)}$$

(2) 计算

$$u^{(K)} = (AD_1^T A^T)^{-1} b$$

并生成向量

$$p^{(K)} = A^T u^{(K)}$$

若  $\max_{1 \leq i \leq m} \{p^{(K)}\} < \varepsilon$ , 终止计算, 问题 (2.3) — (2.4) 的目标函数值  $Z$  无上界, 问题 (2.1) — (2.2)

没有可行解; 否则转 (3)。

(3) 选取适当的步长  $\alpha_k$ , 并产生新的迭代点

$$y^{(k+1)} = y^{(k)} + \alpha_k u^{(k)}$$

使得  $v^{(k+1)} = v^{(k)} + \alpha_k p^{(k)} > 0$

若  $m(n-m)M^2/l^2\sigma^2 < \frac{1}{2}$  (其中  $M = \max_{m+1 \leq i \leq n} \max_{1 \leq j \leq m} \{|a_{ij}|\}$ )

则转 (4), 否则  $K=K+1$ , 返回 (1)

(4) 计算

$$\bar{X} = D^t (AD^t A^T)^{-1} b$$

令  $I_1 = \{i: |X_i| > \varepsilon, i=1, 2, \dots, n\}$ ,  $I_2 = \{1, 2, \dots, n\} \setminus I_1$ , 求解以下方程组

$$A_{I_1} X_{I_1} = b$$

$$A_1^T y^* = C_1$$

令  $x_i = 0, i \in I_2$ , 于是  $(x_1^T, x_2^T), y^*$  分别是问题(2.1)–(2.2)和(2.3)–(2.4)的解。

在实际计算中可以把  $v^{(k)}$  的分量按其大小的顺序排列, 取第  $m+1$  个为  $\sigma$  的值

### 5. 数值试验

对实际的数值运算, 应用拉格朗日乘子估计是十分有效的, 对以下几组数据, 我们对直接应用算法 A 进行计算的和算法 A 的基础上应用拉格日乘子进行计算做了比较

问题	行数	列数	非零元个数	算法 A 迭代次数	算法 A + lagrange 乘子估计迭代次数
1	27	51	102	15	10
2	41	87	178	18	16
3	96	162	777	25	23
4	117	253	1179	35	22
5	282	548	9941	28	24
6	442	868	16036	36	34
7	674	1606	4808	36	32

### 参 考 文 献

1. D. M. Gay, "A variant of karmarkar's linear programming algorithm for problem in standard form", Math. Prog. 37, 81–90, 1987
2. C. C. Gonzaga, "An algorithm for solving linear programming in  $O(n^3L)$  operations", Technical report UCB/ERL M87/10. Electronic Research Laboratory, University of California, Berkeley, CA 94720, Marth 1987
3. M. Iri and H. Imai, "A multiplicative barrier method for linear programming", Algorithmica, 455–482, 1986
4. N. K. Karmarkar, "A new polynomial-time algorithm for linear programming", Combinatorica 4, 373–395, 1984
5. J. Renegar, "A polynomial-time algorithm based on Newton's method for linear programming", Math Prog, 40, 59–93, 1988
6. M. J. Todd, "Recent developments and new directions in linear Programming", Technical Report 829, School of OR & IE, Cornell University, 1988
7. Z. L. Wei, "An exact solution to linear Programming Using an interior Point method", J. of Computational Mathematics 5, 3(1987)
8. 魏紫奎, 吴力: 计算线性不等式组可行解的方法, 数值计算与计算机应用, Vol. 13, No. 1, 1992
9. 魏紫奎: 线性规划内点法中拉格朗日乘子的估计, 未发表
10. 吴力 魏紫奎: 图论在稀疏对称矩阵中的应用, 数学的实践与认识, 1992. 2, 52–62
11. 魏紫奎, 吴力: 线性规划内点法的实现, 数值计算与计算机应用, 将发表
12. 魏紫奎, 吴力: LPIPM1.0 用户使用手册——线性规划内点法 Fortran 程序包, 中国科学院计算中心技术报告 (1990.10)