系 统 科 学 与 数 学
J. Sys. Sci. & Math. Scis.
27(5) (2007, 10), 780–790

图的最大二等分问题的低秩可行方向算法*

穆学文 刘红卫 刘三阳

(西安电子科技大学数学系， 西安 710071)

摘要 基于图的最大二等分问题的半定规划松弛模型，利用矩阵的低秩分解技巧，给出了该问题的半定规划松弛的一种低秩可行方向算法. 在一定的条件下，证明了算法的收敛性. 结合 0.699 随机扰动方法得到原问题的近似最优解. 数值实验表明该方法能有效地求解图的最大二等分问题.

关键词 图的最大二等分问题，半定规划松弛，可行方向算法，随机扰动.

MR(2000) 主题分类号 93C35

1 引言

设 $G=(V,E)$ 代表无向赋权简单图，且 $n=|V|$ 为偶数. 用 w_{ij} 表示边 e_{ij} 的权，若顶点 v_i, v_j 之间无边连接，令 $w_{ij}=0$. 图的最大二等分是指将顶点集 V 分为两个子集 S, $\widetilde{S}$, 使得两端点分别在 S, $\widetilde{S}$ 中的边 e_{ij} 的权 w_{ij} 的和最大，且 $|S|=|\widetilde{S}|$, 即

$$(\mathrm{P})\quad \begin{cases} \max & \displaystyle\sum_{i\in S,\, j\in\widetilde{S}} w_{ij} \\ \text{s.t.} & |S|=|\widetilde{S}| \end{cases}$$

引入向量 $x\in\{-1,1\}^n$, 若顶点 $v_i\in S$, 记 $x_i=1$, 否则记 $x_i=-1$. 令 $W=[w_{ij}]$, 易得

$$\begin{aligned} \sum_{i\in S,\, j\in\widetilde{S}} w_{ij} &= \sum_{i<j} w_{ij}\frac{1-x_ix_j}{2} \\ &= \frac{1}{4}\sum_{i,j} w_{ij}(1-x_ix_j) \\ &= \frac{1}{4}x^{\mathrm{T}}\big(\mathrm{Diag}(We)-W\big)x, \end{aligned}$$

其中， $\mathrm{Diag}(x)$ 表示以 x 的元素为对角元，其余元素为 0 的矩阵， $C=\frac{1}{4}\big(\mathrm{Diag}(We)-W\big)$ 表示 Laplace 矩阵， e 表示全 1 的 n 维向量. 则 (P) 等价于如下问题

$$(\mathrm{IP})\quad \begin{cases} \max & x^{\mathrm{T}}Cx \\ \text{s.t.} & e^{\mathrm{T}}x=0 \\ & x\in\{-1,1\}^n. \end{cases} \tag{1.2}$$

* 跨世纪优秀人才基金项目和陕西省自然科学基金资助项目.
收稿日期：2003-12-01, 收到修改稿日期：2006-04-21.

令 $X = xx^{\mathrm{T}}$, 可以得到图的最大二等分的半定松弛模型 [1]

$$
(\text{SDP1}) \quad \begin{cases} \max & C \cdot X \\ \text{s.t.} & (e_i\, e_i^{\mathrm{T}}) \cdot X = 1, \quad i = 1, 2, \cdots, n \\ & (e\, e^{\mathrm{T}}) \cdot X = 0 \\ & X \succeq 0, \end{cases}
$$

其中 $X \succeq 0$ 表示 X 是半正定矩阵． e_i 表示第 i 个元素为 1 的 n 维单位列向量．定义 $A \cdot B = \mathrm{tr}(A^{\mathrm{T}}B)$, $\mathrm{tr}(A)$ 表示 A 的迹．如果 A 是 n 阶对称矩阵， B, C 为 n 阶方阵，则容易证明 $A \cdot BC = AB \cdot C^{\mathrm{T}}$.

图的最大二等分问题是一个经典的组合优化问题，在网络优化、工程技术中有着广泛的应用，但它是 NP 难题 [1] , 在 P 不等于 NP 的时候，不存在求解该问题的多项式时间精确算法，寻求较好的多项式时间近似算法已经成为问题的主要研究方向．目前，图的最大二等分问题的近似算法大部分是基于半定规划松弛得到的．最早由 Frieze 和 Jerrum[2] 提出了求解该问题的 0.651 近似算法，在此基础上 Ye[3] 利用半定规划松弛方法提出了求解该问题的 0.699 近似算法． Halperin 和 Zwick 通过对原问题的半定规划松弛模型添加三角不等式得到了目前为止最好的 0.7016 和 0.7024 近似算法 [4] ．Xu 和 Han[5] 综合利用前面的方法，进一步对图的最大二等分问题作了研究，其得到的结果甚至好于 Halperin 和 Zwick 的方法．前面的方法都是利用内点算法求解最大二等分问题的半定规划松弛，内点算法求解中小规模半定规划问题比较有效，但不能有效求解大规模的半定规划问题 [2,6] ．学者们提出了非线性规划算法求解半定规划松弛问题，取得了不错的效果 [7,8] ．非线性规划算法是一阶算法，能够克服内点算法内存和时间要求高的缺点，快速有效地得到问题的最优解，但这类算法不是多项式时间算法． Liu 对最大割问题提出了一种固定步长的可行方向算法，但该算法不适合求解带任何其他约束的组合优化问题 [8].

对 (SDP1) 进一步松弛，可得到以下两种新的松弛模型

$$
(\text{SDP2}) \quad \begin{cases} \max & C \cdot X \\ \text{s.t.} & (e_i\, e_i^{\mathrm{T}}) \cdot X \le 1, \quad i = 1, 2, \cdots, n \\ & (e\, e^{\mathrm{T}}) \cdot X \le 1 \\ & X \succeq 0 \end{cases}
$$

$$
(\text{SDP3}) \quad \begin{cases} \max & C \cdot X \\ \text{s.t.} & (e_i\, e_i^{\mathrm{T}}) \cdot X = 1, \quad i = 1, 2, \cdots, n \\ & (e\, e^{\mathrm{T}}) \cdot X \le 1 \\ & X \succeq 0 \end{cases}
$$

本文首先把 (SDP2) 通过低秩分解等价转化为一个非线性规划模型，提出了一种可行方向算法求得 (SDP2) 的最优解，再通过简单的矩阵相加得到 (SDP3) 的一个次优解，然后利用 0.699 随机扰动方法对 (SDP3) 的次优解进行扰动，得到了原问题的一个近似解．在数值试验中，我们把该方法与 Ye-0.699 近似算法进行比较，结果表明：二者所得的解性能几乎没有差异，但在运行时间上我们的算法明显少于 Ye-0.699 近似算法．当规模超过 50 时，时间

大约减少 4 倍. 当规模超过 200 时， Ye-0.699 近似算法由于内存不足，无法计算. 我们的算法则可正常运算. 由此可见，我们的算法在时间和内存的使用上要优于 Ye-0.699 近似算法.

2 低秩分解的可行方向算法

设 S_+^n 与 S_{++}^n 分别表示 n 维半正定矩阵和正定矩阵的集合. 考虑如下定理.

定理 1 (SDP2) 存在秩为 r 的最优解 X^*, 且 r 满足如下条件 $r(r+1) \le n+1$.

定理证明参看文 [7]. 假定 X^* 为问题 (SDP2) 的最优解，定义如下变量

$$\begin{aligned} r^* &= \min\{\mathrm{rank}(X^*)\}, \\ \overline{r} &= \max\{r \ge 0 : r(r+1) \le n+1\}. \end{aligned}$$

显然有 $r^* \le \overline{r} \le n$, 对于任意秩为 r 的矩阵 $X \in S_+^n$, 存在一个矩阵 $V \in R^{n\times n}$, 满足 $X = VV^{\mathrm{T}}$, 其中 V 的后 $n-r$ 列的元素都为 0. 所以 (SDP2) 等价为如下的低秩非线性规划模型

$$(\mathrm{NLP1}) \quad \left\{ \begin{array}{ll} \max & C\cdot(VV^{\mathrm{T}}) \\ \mathrm{s.t.} & (e_i\, e_i^{\mathrm{T}})\cdot(VV^{\mathrm{T}}) \le 1, \quad i=1,2,\cdots,n \\ & (e\, e^{\mathrm{T}})\cdot(VV^{\mathrm{T}}) \le 1 \\ & V \in R^{n\times r} \end{array} \right.$$

易知 (NLP1) 是一个光滑的非线性规划问题，且 (NLP1) 的可行域是凸的. 下面我们给出求解 (NLP1) 的可行方向算法. 首先定义如下函数

$$\begin{aligned} f:\ R^{n\times r} \mapsto R, &\quad f(V) = C\cdot(VV^{\mathrm{T}}), \\ g_i:\ R^{n\times r} \mapsto R, &\quad g_i(V) = e_i\, e_i^{\mathrm{T}}\cdot(VV^{\mathrm{T}}) - 1, \quad i=1,2,\cdots,n, \\ g_{n+1}:\ R^{n\times r} \mapsto R, &\quad g_{n+1}(V) = e\, e^{\mathrm{T}}\cdot(VV^{\mathrm{T}}) - 1. \end{aligned}$$

易得函数 $f(V)$ 在点 V 的梯度为 $G = 2CV$, 函数 $g_i(V)$ 在点 V 的梯度为 $\nabla g_i(V) = 2(e_i\, e_i^{\mathrm{T}})V,\ i=1,2,\cdots,n$, 函数 $g_{n+1}(V)$ 在点 V 的梯度为 $\nabla g_{n+1}(V) = 2(e\, e^{\mathrm{T}})V$.

假定矩阵 V^k 是 (NLP1) 的第 k 个迭代点，在 V^k 点的可行方向可以通过求解如下二次规划得到.

$$(\mathrm{QP}) \quad \left\{ \begin{array}{l} \max\ \ t - \dfrac{u}{2} D\cdot D \\ \mathrm{s.t.}\ \ -G^k\cdot D + t \le 0 \\ \qquad (e_i\, e_i^{\mathrm{T}})\cdot\big(V^k(V^k)^{\mathrm{T}}\big) - 1 + 2(e_i\, e_i^{\mathrm{T}})V^k\cdot D + t \le 0, \quad i=1,2,\cdots,n \\ \qquad (e\, e^{\mathrm{T}})\cdot\big(V^k(V^k)^{\mathrm{T}}\big) - 1 + 2(e\, e^{\mathrm{T}})V^k\cdot D + t \le 0 \end{array} \right.$$

这里 u 是小于 1 的与 n 有关的正数， $t \in R \times R^{n\times r}$.

定理 2 假设 (t^k, D^k) 是 (QP) 的最优解，若 $D^k \neq 0$, 则 D^k 必是 (NLP1) 在点 V^k 的可行上升方向.

证 (t^k, D^k) 是 (QP) 的最优解，由于 $t=0,\ D=0$ 是 (QP) 的一个可行解，且 $D^k \neq 0$, 所以有 $-\frac{u}{2}D^k{\cdot}D^k < 0$, 及 $t^k - \frac{u}{2}D^k{\cdot}D^k \geq 0$, 从而 $t^k \geq \frac{u}{2}D^k{\cdot}D^k > 0$. 又因为 $-G^k{\cdot}D^k + t^k \leq 0$, 所以有 $G^k{\cdot}D^k > 0$, 即 D^k 是 (NLP1) 在点 V^k 的上升方向.

下证 D^k 是一个可行方向. 对 $\delta_i > 0,\ i = 1, 2, \cdots, n+1$, 我们有

$$\begin{aligned} g_i(V^k + \delta_i D^k) &= (e_i\, e_i^{\mathrm T}){\cdot}(V^k + \delta_i D^k)(V^k + \delta_i D^k)^{\mathrm T} - 1 \\ &= (e_i\, e_i^{\mathrm T}){\cdot}V^k(V^k)^{\mathrm T} - 1 + 2\delta_i(e_i\, e_i^{\mathrm T})V^k{\cdot}D^k + \delta_i^2(e_i\, e_i^{\mathrm T}){\cdot}D^k(D^k)^{\mathrm T}, \\ g_{n+1}(V^k + \delta_{n+1} D^k) &= (e\, e^{\mathrm T}){\cdot}(V^k + \delta_{n+1} D^k)(V^k + \delta_{n+1} D^k)^{\mathrm T} - 1 \\ &= (e\, e^{\mathrm T}){\cdot}V^k(V^k)^{\mathrm T} - 1 + 2\delta_{n+1}(e\, e^{\mathrm T})V^k{\cdot}D^k + \delta_{n+1}^2(e\, e^{\mathrm T}){\cdot}D^k(D^k)^{\mathrm T}. \end{aligned}$$

若 $(e_i\, e_i^{\mathrm T})V^k{\cdot}D^k \geq 0$, 设 $\delta_i < 1$, 且满足 $\delta_i^2(e_i\, e_i^{\mathrm T}){\cdot}D^k(D^k)^{\mathrm T} < t^k$. 则必有

$$g_i(V^k + \delta_i D^k) < (e_i\, e_i^{\mathrm T}){\cdot}\big(V^k(V^k)^{\mathrm T}\big) - 1 + 2(e_i\, e_i^{\mathrm T})V^k{\cdot}D^k + t^k \leq 0;$$

若 $(e_i\, e_i^{\mathrm T})V^k{\cdot}D^k < 0$, 取 δ_i 满足 $2\delta_i(e_i\, e_i^{\mathrm T})V^k{\cdot}D^k + \delta_i^2(e_i\, e_i^{\mathrm T}){\cdot}D^k(D^k)^{\mathrm T} < 0$, 解得

$$0 < \delta_i \leq \frac{-2(e_i\, e_i^{\mathrm T})V^k{\cdot}D^k}{(e_i\, e_i^{\mathrm T}){\cdot}D^k(D^k)^{\mathrm T}}.$$

则必有 $g_i(V^k + \delta_i D^k) \leq 0$.

同理，若 $(e\, e^{\mathrm T})V^k{\cdot}D^k \geq 0$, 取 δ_{n+1} 满足 $\delta_{n+1} < 1$ 及 $\delta_{n+1}^2(e\, e^{\mathrm T}){\cdot}D^k(D^k)^{\mathrm T} < t^k$. 有

$$g_{n+1}(V^k + \delta_{n+1} D^k) < (e\, e^{\mathrm T}){\cdot}\big(V^k(V^k)^{\mathrm T}\big) - 1 + 2(e\, e^{\mathrm T})V^k{\cdot}D^k + t^k \leq 0$$

若 $(e\, e^{\mathrm T})V^k{\cdot}D^k < 0$, 取 δ_{n+1} 满足 $2\delta_{n+1}(e\, e^{\mathrm T})V^k{\cdot}D^k + \delta_{n+1}^2(e\, e^{\mathrm T}){\cdot}D^k(D^k)^{\mathrm T} < 0$, 解得

$$0 < \delta_{n+1} \leq \frac{-2(e\, e^{\mathrm T})V^k{\cdot}D^k}{(e\, e^{\mathrm T}){\cdot}D^k(D^k)^{\mathrm T}}.$$

则必有 $g_{n+1}(V^k + \delta_{n+1} D^k) \leq 0$.

取 $\delta = \min\{\delta_i,\ i = 1, 2, \cdots, n+1\}$, 必满足 $g_i(V^k + \delta D^k) \leq 0,\ i = 1, 2, \cdots, n+1$. 由于 (NLP1) 的可行域是凸的，所以 D^k 是一个可行方向. 结论得证.

由于 (QP) 维数较高，直接求解比较困难，我们通过求解其对偶规划来得到 (QP) 的最优解. 其对偶规划如下

$$(\mathrm{DQP})\quad \begin{cases} \min \dfrac{1}{2u}\| - \lambda_1 G^k + \displaystyle\sum_{i=2}^{n+1} \lambda_i (2\, e_{i-1}\, e_{i-1}^{\mathrm T})V^k + \lambda_{n+1}(2\, e\, e^{\mathrm T})V^k\|_F^2 \\ \qquad - \displaystyle\sum_{i=2}^{n+1} \lambda_i\big((e_{i-1}\, e_{i-1}^{\mathrm T}){\cdot}V^k(V^k)^{\mathrm T} - 1\big) - \lambda_{n+2}\big((2\, e\, e^{\mathrm T})(V^k(V^k)^{\mathrm T}) - 1\big) \\ \text{s.t. } \displaystyle\sum_{i=1}^{n+2} \lambda_i = 1, \quad \lambda_i \geq 0, \quad i = 1, 2, \cdots, n+2 \end{cases}$$

这里 $\| A \|_F$ 表示 A 的 Frobenius 范数. 由于问题 (QP) 是凸二次规划, 由对偶原理可知 (DQP) 也是一个凸二次规划问题, 求解该问题存在有效的多项式时间算法 [9,10]. 通过求解 (DQP) 得到原问题的解. 有如下定理.

定理 3 (1)(QP) 存在最优解 (t^k, D^k).

(2)(QP) 存在最优解 (t^k, D^k) 当且仅当存在矩阵 $P^k \in R^{n\times n}$ 以及乘子 λ_i^k, $i=1,2,\cdots,n+2$, 满足以下条件成立

a) $\sum\limits_{i=1}^{n+2} \lambda_i^k = 1, \quad \lambda_i^k \geq 0, \quad i=1,2,\cdots,n+2,$

b) $\lambda_1^k(-G^k\cdot D^k + t^k) = 0,$

c) $\lambda_{i+1}^k\big((e_i\, e_i^{\rm T})\cdot V^k(V^k)^{\rm T} - 1 + 2(e_i\, e_i^{\rm T})V^k\cdot D^k + t^k\big) = 0, \quad i=1,2,\cdots,n,$

d) $\lambda_{n+2}^k\big((e\, e^{\rm T})\cdot V^k(V^k)^{\rm T} - 1 + 2(e\, e^{T})V^k\cdot D^k + t^k\big) = 0,$

e) $P^k = -\lambda_1^k G^k + \sum\limits_{i=2}^{n+1} \lambda_i^k (2\, e_{i-1}\, e_{i-1}^{\rm T})V^k + \lambda_{n+2}^k(2\, e\, e^{\rm T})V^k,$

f) $D^k = -\dfrac{1}{u}P^k,$

g) $t^k = \dfrac{1}{u}\|P^k\|_F^2 - \sum\limits_{i=2}^{n+1} \lambda_i^k\big((e_{i-1}\, e_{i-1}^{\rm T})\cdot V^k(V^k)^{\rm T} - 1\big) - \lambda_{n+2}^k\big((e\, e^{\rm T})\cdot V^k(V^k)^{\rm T} - 1\big),$

h) $-G^k\cdot D^k + t^k \leq 0,$

i) $(e_i\, e_i^{\rm T})\cdot V^k(V^k)^{\rm T} - 1 + 2(e_i\, e_i^{\rm T})V^k\cdot D^k + t^k \leq 0, \quad i=1,2,\cdots,n,$

j) $(e\, e^{\rm T})\cdot V^k(V^k)^{\rm T} - 1 + 2(e\, e^{\rm T})V^k\cdot D^k + t^k \leq 0.$

(3) 乘子 λ_i^k, $i=1,2,\cdots,n+2$ 满足 a), b), $\cdots$, j) 成立当且仅当它是 (DQP) 的解.

证 参看文 [11].

利用定理 3 中的结论 e), f) 可求得 (NLP1) 在 V^k 点的可行方向 D^k . 由定理 2 的证明过程可知, 若 $D^k \neq 0$, 则存在 $\delta > 0$, 满足 $g_i(V^k + \delta D^k) \leq 0$, $i=1,2,\cdots,n+1$.

设 $\delta_i \geq 0$, 求解等式 $g_i(V^k + \delta D^k) = 0$, $i=1,2,\cdots,n+1$. 得到

$$\delta_i = \frac{-(e_i\, e_i^{\rm T})V^k\cdot D^k + \sqrt{\big((e_i\, e_i^{\rm T})V^k\cdot D^k\big)^2 - \big((e_i\, e_i^{\rm T})\cdot V^k(V^k)^{\rm T} - 1\big)\big((e_i\, e_i^{\rm T})\cdot D^k(D^k)^{\rm T}\big)}}{(e_i\, e_i^{\rm T})\cdot D^k(D^k)^{\rm T}},$$

$$\delta_{n+1} = \frac{-(e\, e^{\rm T})V^k\cdot D^k + \sqrt{\big((e\, e^{\rm T})V^k\cdot D^k\big)^2 - \big((e\, e^{\rm T})\cdot V^k(V^k)^{\rm T} - 1\big)\big((e\, e^{\rm T})\cdot D^k(D^k)^{\rm T}\big)}}{(e\, e^{\rm T})\cdot D^k(D^k)^{\rm T}}.$$

取

$$\delta = \min\{\delta_i,\ i=1,2,\cdots,n+1\}. \tag{1}$$

定理 4 若 $D^k \neq 0$, 由 (1) 所取的 δ 一定是可行点 V^k 在方向 D^k 的精确搜索步长.

证 任取 $h>0$, 则

$$f(V^k+hD^k)=C\cdot(V^k+hD^k)(V^k+hD^k)^{\mathrm{T}}$$
$$=C\cdot V^k(V^k)^{\mathrm{T}}+2hCV^k\cdot D^k+h^2C\cdot D^k(D^k)^{\mathrm{T}}.$$

定义函数 $\varphi:R\mapsto R$, $\varphi(h)=f(V^k+hD^k)$, 则 $\varphi'(h)=2CV^k\cdot D^k+2hC\cdot D^k(D^k)^{\mathrm{T}}$.

由 D^k 是可行上升方向可知 $2CV^k\cdot D^k>0$. C 是 Laplacian 矩阵，所以 $C\succeq 0$, 则 $C\cdot D^k(D^k)^{\mathrm{T}}\geq 0$. 当 $h>0$, 有 $\varphi'(h)>0$, $\varphi(h)$ 是单调上升函数. 由 δ 的取法可知 δ 是满足 $V^k+\delta D^k$ 可行的最大步长. 定理得证.

下面我们给出可行方向算法的计算步骤.

步骤 1 假定 V^0 是 (NLP1) 的一个初始可行解， ε 是很小的正常数， $u=\frac{0.5}{n}$, $k=0$;

步骤 2 计算函数 f 在 V^k 的梯度 $G^k=2CV^k$;

步骤 3 求解二次规划 (QP) 的对偶模型 (DQP). 由定理 2 中的条件 e), f) 得到 V^k 的可行上升方向 D^k;

步骤 4 计算 $\|D^k\|_F$, 若 $\|D^k\|_F<\varepsilon$, 则停止， $V^*=V^k$ 就是要求的点；否则转步骤 5;

步骤 5 由 (1) 的方法取得 δ, 令 $V^{k+1}=V^k+\delta D^k$, $k=k+1$, 转步骤 2.

3 收敛性分析

下面给出算法的收敛性证明.

定理 5[5] 假定 $A,B\in S_+^n$, 则有如下结论成立

$$\lambda_{\min}(A)\,\mathrm{tr}(B)\leq A\cdot B\leq\lambda_{\max}(A)\,\mathrm{tr}(B),$$

这里 $\lambda_{\min}(A)$, $\lambda_{\max}(A)$ 分别表示矩阵 A 的最小特征值和最大特征值.

定理 6 在低秩可行方向算法中有如下结论成立

$$\lambda_{\min}\|V^{k+1}-V^k\|_F^2\leq f(V^{k+1})-f(V^k)$$
$$\leq n^{\frac{1}{2}}\lambda_{\max}(C)\|V^{k+1}-V^k\|+\lambda_{\max}(C)\|V^{k+1}-V^k\|_F^2.$$

证 由于 $V^{k+1}=V^k+\delta D^k$, 则可得

$$f(V^{k+1})-f(V^k)=C\cdot(V^k+\delta D^k)(V^k+\delta D^k)^{\mathrm{T}}-C\cdot V^k(V^k)^{\mathrm{T}}$$
$$=2\delta CV^k\cdot D^k+\delta^2C\cdot D^k(D^k)^{\mathrm{T}}$$
$$=2CV^k(V^{k+1}-V^k)+C\cdot[(V^{k+1}-V^k)(V^{k+1}-V^k)^{\mathrm{T}}].$$

由定理 2 和定理 4 可得 $2\delta CV^k\cdot D^k\geq 0,\delta^2C\cdot D^k(D^k)^{\mathrm{T}}\geq 0$, 根据定理 5 可得

$$f(V^{k+1})-f(V^k)\geq C\cdot\left[(V^{k+1}-V^k)(V^{k+1}-V^k)^{\mathrm{T}}\right]=\lambda_{\min}(C)\|V^{k+1}-V^k\|_F^2,$$

又根据文 [7] 可得

$$f(V^{k+1})-f(V^k)\leq n^{\frac{1}{2}}\lambda_{\max}(C)\|V^{k+1}-V^k\|+\lambda_{\max}(C)\|V^{k+1}-V^k\|_F^2.$$

定理得证.

定理 7 如果在低秩可行方向算法中 $\|D^k\| < \varepsilon$ 不成立, 则当 $k \to \infty$ 时, $V^{k+1} - V^k \to 0$, 即算法收敛于问题 NLP1 的一个稳定点 (KKT 点).

证 类似于文 [8] 中定理 6 的证明方法.

由定理 7, 有如下结论成立.

定理 8 对任意初始点 V^0 以及精度要求 $\varepsilon > 0$, 低秩可行方向算法总在 $\left[\frac{n\lambda_{\max}(C)}{\varepsilon^2\lambda_{\min}(C)}\right]$ 个迭代步内终止. 这里 $\left[\frac{n\lambda_{\max}(C)}{\varepsilon^2\lambda_{\min}(C)}\right]$ 表示不超过 $\frac{n\lambda_{\max}(C)}{\varepsilon^2\lambda_{\min}(C)}$ 的正整数.

证 类似于文 [8] 中定理 7 的证明方法.

定理 9 如果 $D^k = 0$, 则在点 V^k 处, 存在 $\mu_i^k \in R^{n+1}$, 满足

$$
\begin{aligned}
&2CV^k = 2\sum_{i=1}^{n} \mu_i^k (e_i\, e_i^{\mathrm{T}}) V^k + 2\mu_{n+1}^k (e\, e^{\mathrm{T}}) V^k,\\
&\mu_i^k \big((e_i\, e_i^{\mathrm{T}})\cdot V^k (V^k)^{\mathrm{T}} - 1\big) = 0, \quad i = 1, 2, \cdots, n,\\
&\mu_{n+1}^k \big((e\, e^{\mathrm{T}})\cdot V^k (V^k)^{\mathrm{T}} - 1\big) = 0, \quad \mu_i^k \geq 0, \quad i = 1, 2, \cdots, n+1
\end{aligned}
$$

成立. 即 V^k 是 (NLP1) 的 KKT 点.

证 如果 $D^k = 0$, 由定理 3 的结论 h) 可得 $t^k \leq G^k \cdot D^k = 0$, 又因为 $t^k \geq 0$, 所以可得 $t^k = 0$. 由结论 f) 可得 $P^k = -uD^k = 0$. 由结论 e) 可得

$$
P^k = -\lambda_1^k G^k + \sum_{i=2}^{n+1} \lambda_i^k (2\, e_{i-1}\, e_{i-1}^{\mathrm{T}}) V^k + \lambda_{n+2}^k (2e\, e^{\mathrm{T}}) V^k = 0
$$

若 $\lambda_1^k \neq 0$, 令 $\mu_i^k = \frac{\lambda_{i+1}^k}{\lambda_1^k}$, $i = 1, 2, \cdots, n+1$, 可得 $2CV^k = 2\sum\limits_{i=1}^{n} \mu_i^k (e_i\, e_i^{\mathrm{T}}) V^k + 2\mu_{n+1}^k (e\, e^{\mathrm{T}}) V^k$. 由定理 3 的结论 c), d) 可得

$$
\begin{aligned}
&\mu_i^k \big((e_i\, e_i^{\mathrm{T}})\cdot V^k (V^k)^{\mathrm{T}} - 1\big) = 0, \quad i = 1, 2, \cdots, n,\\
&\mu_{n+1}^k \big((e\, e^{\mathrm{T}})\cdot V^k (V^k)^{\mathrm{T}} - 1\big) = 0, \quad \mu_i^k \geq 0, \quad i = 1, 2, \cdots, n+1.
\end{aligned}
$$

若 $\lambda_1^k = 0$, 则有 $\sum\limits_{i=2}^{n+1} \lambda_i^k (2\, e_{i-1}\, e_{i-1}^{\mathrm{T}}) V^k + \lambda_{n+2}^k (2\, e\, e^{\mathrm{T}}) V^k = 0$, 等式两边与 V^k 作内积, 得

$$
\sum_{i=2}^{n+1} \lambda_i^k \big((2\, e_{i-1}\, e_{i-1}^{\mathrm{T}}) V^k \cdot V^k\big) + \lambda_{n+2}^k ((2\, e\, e^{\mathrm{T}}) V^k \cdot V^k) = 0.
$$

由上式和定理 3 的结论 g) 可得

$$
\begin{aligned}
t^k &= \frac{1}{u}\|P^k\|_F^2 - \sum_{i=2}^{n+1} \lambda_i^k \big((e_{i-1}\, e_{i-1}^{\mathrm{T}})\cdot V^k (V^k)^{\mathrm{T}} - 1\big) - \lambda_{n+2}^k \big((e\, e^{\mathrm{T}})\cdot V^k (V^k)^{\mathrm{T}} - 1\big)\\
&= -\sum_{i=2}^{n+1} \lambda_i^k \big((e_{i-1}\, e_{i-1}^{\mathrm{T}})\cdot V^k (V^k)^{\mathrm{T}} - 1\big) - \lambda_{n+2}^k \big((e\, e^{\mathrm{T}})\cdot V^k (V^k)^{\mathrm{T}} - 1\big)
\end{aligned}
$$

$$
\begin{aligned}
&= -\sum_{i=2}^{n+1}\lambda_i^k(e_{i-1}\,e_{i-1}^{\mathrm{T}})V^k{\cdot}V^k - \lambda_{n+2}^k(e\,e^{\mathrm{T}})V^k{\cdot}V^k + \sum_{i=2}^{n+2}\lambda_i^k \\
&= \sum_{i=2}^{n+2}\lambda_i^k = \sum_{i=1}^{n+2}\lambda_i^k.
\end{aligned}
$$

由定理 3 的结论 a) 可得 $t^k = \sum\limits_{i=1}^{n+2}\lambda_i^k = 1$, 与前所证明 $t_k = 0$ 矛盾. 则 $\lambda_1^k \neq 0$. 定理得证.

由定理 9, 在可行方向算法中，采用 $\|D^k\|_F < \varepsilon$ 作为一个终止条件，其中 $\varepsilon > 0$ 是精度要求. 算法总是在 (NLP1) 的 KKT 点处终止. 下面给出算法全局收敛性的一个充分条件.

定理 10 设 V^k 是 (NLP1) 的 KKT 点，令

$$
S^k = \sum_{i=1}^{n}\mu_i^k(e_i\,e_i^{\mathrm{T}}) + \mu_{n+1}^k(e\,e^{\mathrm{T}}) - C, \quad \mu_i^k \geq 0, \quad i = 1, 2, \cdots, n+1,
$$

若 $S^k \succeq 0$, 则 $X^k = V^k(V^k)^{\mathrm{T}}$ 是 (SDP2) 的全局最优解.

证 由于 V^k 是 (NLP1) 的 KKT 点，由定理 9 得

$$
2CV^k = 2\sum_{i=1}^{n}\mu_i^k(e_i\,e_i^{\mathrm{T}})V^k + 2\mu_{n+1}^k(e\,e^{\mathrm{T}})V^k, \quad \mu_i^k \geq 0, \quad i = 1, 2, \cdots, n+1,
$$

两边与 V^k 作内积，得到

$$
2CV^k{\cdot}V^k = \left(2\sum_{i=1}^{m}\mu_i^k(e_i\,e_i^{\mathrm{T}})V^k + 2\mu_{n+1}^k(e\,e^{\mathrm{T}})V^k\right){\cdot}V^k,
$$

又由定理 9 可得

$$
\sum_{i=1}^{n}\mu_i^k\big((e_i\,e_i^{\mathrm{T}})V^k{\cdot}V^k - 1\big) + \mu_{n+1}^k\big((e\,e^{\mathrm{T}})V^k{\cdot}V^k - 1\big) = 0,
$$

即有

$$
\sum_{i=1}^{n}\mu_i^k(e_i\,e_i^{\mathrm{T}})V^k{\cdot}V^k + \mu_{n+1}^k(e\,e^{\mathrm{T}})V^k{\cdot}V^k = \sum_{i=1}^{n+1}\mu_i^k,
$$

可得

$$
CV^k{\cdot}V^k = C{\cdot}V^k(V^k)^{\mathrm{T}} = C{\cdot}X^k = \sum_{i=1}^{n+1}\mu_i^k.
$$

假设 X 是 (SDP2) 的任意可行解，由 $X \succeq 0$, $S^k \succeq 0$, 可得 $S^k{\cdot}X \geq 0$, 又因为 $S^k = \sum\limits_{i=1}^{n}\mu_i^k(e_i\,e_i^{\mathrm{T}}) + \mu_{n+1}^k(e\,e^{\mathrm{T}}) - C$, 可得 $\Big(\sum\limits_{i=1}^{n}\mu_i^k(e_i\,e_i^{\mathrm{T}}) + \mu_{n+1}^k(e\,e^{\mathrm{T}}) - C\Big){\cdot}X \geq 0$, 即

$$
C{\cdot}X \leq \Big(\sum_{i=1}^{n}\mu_i^k(e_i\,e_i^{\mathrm{T}}) + \mu_{n+1}^k(e\,e^{\mathrm{T}})\Big){\cdot}X.
$$

又 X 是 (SDP2) 的任意可行解, 则有 $\Big(\sum_{i=1}^{n}\mu_i^k(e_i\,e_i^{\mathrm{T}})+\mu_{n+1}^k(e\,e^{\mathrm{T}})\Big)\cdot X \le \sum_{i=1}^{n+1}\mu_i^k$, 即 $C\cdot X \le \sum_{i=1}^{n+1}\mu_i^k = C\cdot X^k$, 可得 $X^k = V^k(V^k)^{\mathrm{T}}$ 是 (SDP2) 的全局最优解. 定理得证.

4 求得原问题的近似解

对于 (SDP2) 的最优解 $X^k = V^k(V^k)^{\mathrm{T}}$, 总可得到 (SDP3) 的一个次优解. 下面我们给出从 $X^k = V^k(V^k)^{\mathrm{T}}$ 得到 (SDP3) 的次优解的方法.

令 $x_{i_0i_0} = \max\{x_{ii}|x_{ii}\ne 1, i=1,2,\cdots,n\}$, 其中 i_0 表示 $X = V^k(V^k)^{\mathrm{T}}$ 不为 1 的对角元素的最大值所在行标的号. $x_{i_1i_1} = \min\{x_{ii}|x_{ii}\ne 1,\ i=1,2,\cdots,n\}$, 其中 i_1 表示 $X = V^k(V^k)^{\mathrm{T}}$ 不为 1 的对角元素的最小值所在行标的号. 令 $Y_1 = X + R$, 其中

$$R = \begin{pmatrix} \cdots & i_0 & \cdots & i_1 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & 1-x_{i_0i_0} & \cdots & x_{i_0i_0}-1 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & 1-x_{i_0i_0} & \cdots & x_{i_0i_0}-1 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix},$$

则 Y_1 中对角元不为 1 的元素减少了一个, 由于 $R\succeq 0,\ 1-x_{i_0i_0}+x_{i_1i_1}\le 1$, 所以 Y_1 是 (SDP2) 的可行解且 $C\cdot Y_1 \le C\cdot X$, 重复上面的方法, 总可找到一个 Y, 其对角线上小于 1 的元素至多为 1 个, 满足 Y 是 (SDP2) 的可行解, 且 $C\cdot Y \le C\cdot X$. 设 Y 中对角线上不为 1 的那个元素为 $y_{i_0i_0}$, 令矩阵 R' 为第 i_0 个对角元素为 $1-y_{i_0i_0}$, 其余元素都是零的 n 阶方阵. 令 $X^* = Y + R'$, 则 X^* 就是 SDP3 的次优解.

基于 (SDP3), 也可得到图的最大二等分问题的 0.699 近似算法. 具体证明参见文 [12]. 得到 SDP3 的次优解 X^* 后, 利用文 [12] 的 0.699 随机扰动方法对 X^* 随机扰动, 便可得到图的最大二等分问题的一个较好的近似解.

5 数值实验

利用 AMD Athlon 单机, Matlab6.1 编程, 对随机赋权 (非负) 简单图分别用算法 1 和算法 2 求解图的最大二等分问题, 所得的数值结果表 1 和表 2 所示

表 1 对小规模问题两种算法性能比较表

n	算法 1 的目标值	算法 1 平均运行时间 (秒)	算法 2 的目标值	算法 2 平均运行时间 (秒)
10	47.75796	0.06200	47.75796	0.04699
20	168.63262	0.21800	168.63262	0.11000
30	402.59497	0.46900	402.59497	0.17200
50	1097.46959	2.31300	1097.40702	0.56200
70	2140.09953	7.50000	2140.13888	2.23400
90	3474.18947	18.62500	3476.39716	4.51600
100	4347.09892	22.18700	4347.34077	6.12500
150	9520.35971	88.922000	9533.19937	19.01600
200	16698.02900	289.359000	16668.70918	55.95400

表 2 对中等规模问题两种算法性能比较表

n	算法 1 的目标值	算法 1 平均运行时间 (秒)	算法 2 的目标值	算法 2 平均运行时间 (秒)
300	//	//	37298.5046	197.8900
400	//	//	64296.5591	211.1230
500	//	//	10135.8995	232.1570
600	//	//	14507.0494	258.1250
700	//	//	2140.13888	283.4210
800	//	//	3476.39716	310.2320
900	//	//	4347.34077	342.1132
1000	//	//	9533.19937	376.1600
1100	//	//	16668.70918	417.1150
1200	//	//	22075.78113	479.1123
1300	//	//	25995.09437	542.3210
1400	//	//	32547.39762	672.3570

其中算法 1 表示利用 Ye[3] 的 0.699 随机扰动方法求得原问题的解. 算法 2 表示通过可行方向算法计算 (SDP2) , 得到其最优解, 利用第 4 节中的方法得到 (SDP3) 的近似解, 然后用文 [12] 的 0.699 随机扰动方法得原问题的次优解.

表 1 和表 2 的数值实验表明, 两种算法得到的次优解的性能没有明显的差异, 但在运行时间上算法明显优于算法 1. 特别当规模超过 50 时, 时间大约减少 4 倍. 当问题规模超过 200 时, 算法 1 由于内存不足, 无法计算. 算法 2 则可正常运算, 在内存为 256M 的单机上最大能够求解到 1400 维的矩阵变量. 所以算法 2 在时间和内存的使用上都要优于算法 1.

下面的例子是 Ginaldi 教授编写的程序 rudy 生成的最大割标准问题, 维数由 800 到 3000 , 由于利用我们有限内存为 256M 的单机不能够求解规模大于 2000 的问题, 所以我们从中选取了规模为 800 的几个算例作为求解图的最大二等分问题的算例, 文 [7] 有具体的介绍. 由于半定规划内点算法的局限性, 在 PC 机上无法实现, 所以只列出了利用我们的方法得到的结果, 见表 3.

表 3 算法的结果

问题	问题的规模	算法 2 的目标值	算法 2 平均运行时间 (秒)	算法 2 的终止误差
G01	800	9921	140.5780	10^{-5}
G02	800	9940	163.9840	10^{-5}
G03	800	9952	171.6410	10^{-5}
G10	800	293	249.9059	10^{-5}
G11	800	144	153.5780	10^{-5}
G12	800	96	74.1870	10^{-5}
G13	800	138	144.8430	10^{-5}
G14	800	2607	444.8430	10^{-5}
G15	800	2725	1006.2350	10^{-5}
G16	800	2687	934.0630	10^{-5}

在表 3 中, 由于 Ye 的 0.699 随机扰动方法在 PC 机上不能够求解规模为 800 的问题, 所以只利用低秩可行方向算法得到了问题的计算结果. 所以我们的方法是求解图的最大二等分问题的一个有效的算法.

参 考 文 献

[1] Frieze A and Jerrum M. Improved approximation algorithms for max k-cut and max BISECTION. *Proc. 4th IPCO Conference*, 1995, **920**: 1–13.
[2] Alizadeh F. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 1995, **5**(1): 13–51.
[3] Ye Y Y. A 0.699-approximation algorithm for max-Bisection. *Math. Programming*, 2001, **90**: 101–111.
[4] Eran Halperin and Uri Zwick. A Unified Framework for Obtaining Improved Approximation Algorithms for Maximum Graph Bisection Problems. Proceedings in 8th International IPCO Conference, 2001, LNCS 2081, 210–225.
[5] Xu D and Han J. Approximation algorithm for max-BISECTION problem with the positive semidefinite relaxation. *Journal of Computational Mathematics*, 2003, **21**(3): 357–366.
[6] Han Q. A projection and contraction methods for semidefinite programming. *Applied mathematics and Computation*, 1998, **95**: 275–289.
[7] Burer S and Monteiro R D C. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program*, 2003, **95**: 329–357.
[8] Liu H W, Wang X H and Liu S Y. Feasible direction algorithm for solving SDP relaxation of the quadratic $\{-1,1\}$ programming problems. *Optimization Methods and Software*, 2004, **19**(2): 125–136.
[9] Takashi Tsuchiya. Global convergence of the affine scaling algorithm for primal degenerate strictly convex quadratic programming problems. *Annals of Operations Research*, 1993, **47**(2): 509–539
[10] Goldfarb D and Liu S. An primal interior point algorithm for convex quadratic programming. *Mathematical Programming*, 1990, **49**(2): 325–340
[11] Marko M, Makeka and Pekka Neittaanmaki. Nonsmooth optimization: analysis and algorithms with application to optimization control. Singapore: World scientific, 1992.
[12] 刘红卫．半定规划及其应用．西安电子科技大学博士学位论文， 2002.

A FEASIBLE DIRECTION ALGORITHM FOR MAX BISECTION VIA LOW-RANK FACTORIZATION

Mu Xuewen　　Liu Hongwei　　Liu Sanyang

(*Department of Applied Mathematics, Xidian University, Xi'an* 710071)

Abstract Based on the semidefinite programming relaxation of max Bisection, a feasible direction algorithm is given to solve the relaxation. Coupled with the 0.699 randomized method, the approximate solution of max Bisection is obtained. Furthermore, its convergent result is given. The numerical experiment shows that the algorithm can solve the max Bisection efficiently.

Key words Max bisection problem, semidefinite programming relaxation, feasible direction, randomized method.