第 17 卷 第 2 期　　中 国 科 学 院 研 究 生 院 学 报　　Vol. 17　No. 2
2000 年　12 月　　Journal of Graduate School, Academia Sinica　　Dec. 2000

一种新的椭球算法

杨德庄　张敏洪
(中国科学技术大学研究生院, 北京 100039)
张利华
(中国科学院自然科学史研究所, 北京 100010)

摘　要　基于更动约束的思想[1]与方法, 提出了求解线性规划问题的新椭球算法. 它与 L. G. Khachian 的椭球算法[2]不同, 在新算法的椭球迭代过程中, 不仅用约束不等式割掉不含约束集的半个椭球(椭球中心不在约束集内时), 称之为约束割; 而且在椭球中心落在约束集内时, 它用目标不等式割掉含约束集的半个椭球, 称之为目标割. 新算法的不等式系统是由原规划(或对偶规划)的约束不等式与目标不等式组成的(规模小), 而不是由原椭球算法的 K-K-T 条件[5]组成的不等式系统(规模大). 这种新椭球算法即有多项式计算复杂性的特性, 又在迭代过程中得到一系列单调趋向最优解的可行解(在解存在时). 如果认为已得满意解, 可随时停机. 对于实际问题, 大多数是变量有界的, 初始椭球不大, 因此新算法更为实际, 有效.

关键词　椭球算法, 约束割, 目标割

0　引言

对于线性规划(简记 LP) 问题, 1947 年 G. B. Dantzig 提出了单纯形算法[3], 但它不是多项式计算复杂性的算法. 后来, 1979 年 L. G. Khachian 提出了第一个多项式计算复杂性的算法[2], 称为椭球算法. 椭球算法的优点在于算法是多项式计算复杂性的, 但是其缺点在于因起始椭球体积过大, 可行集太小, 在实际计算中似乎还是劣于单纯性算法. 在至今为止的所有 LP 算法中, 椭球算法的思路是别具一格的, 它与 G. B. Dantzig 的单纯性算法和 N. Karmarkar 算法[6]的思路完全不同, 它的迭代不是从一个解出发, 确定一个方向, 再决定一个步长, 到达一个新的解, 然后如此反复, 直至最后结果; 而是从一个椭球到更小的椭球, 直至最后.

从 1979 年至今, 为了改进原椭球算法(以下称 Khachian 的椭球算法为原椭球算法), LP 的研究者们发展了几种原椭球算法的变形算法, 比如深切割法(deep cuts), 替代切割法(surrogate cuts), 平行切割法(parrallel cuts) 以及用单纯性替代椭球的算法[6]等.

所有这些椭球算法和变形算法, 都是基于求解 K-K-T 条件:

收稿日期: 2000-11-20
作者简介: 杨德庄, 男, 1939 年 8 月生, 教授, 博士生导师

$$\begin{aligned} &1)\quad Ax \geqslant b, \qquad x \geqslant 0; \\ &2)\quad A^T w \leqslant c, \qquad w \geqslant 0; \\ &3)\quad b^T w = c^T x. \end{aligned} \tag{1}$$

组成的不等式组, 其中原规划为

$$\begin{aligned} &\min\ c^T x \\ &\text{s.t.}\ \ Ax \geqslant b, \qquad x \geqslant 0. \end{aligned} \tag{2}$$

对偶规划为

$$\begin{aligned} &\max\ b^T w \\ &\text{s.t.}\ \ A^T w \leqslant c, \qquad w \geqslant 0. \end{aligned} \tag{3}$$

原椭球算法及其所有的变形算法的思路: 不断迭代构造一系列椭球, 第 k 次迭代的椭球记为 E_k, 并记式(1)的解集在 E_0 中的子集为 P, 使得 $P \subset E_k$. E_k 的体积按几何级数收缩. 要么某次 E_k 的中心 $x^k \in P$, 则迭代终止; 要么某次 E_k 的体积足够小, 以致于其中不含(1)式的可行解, 也终止. 变形算法的研究都集中在如何改善相邻椭球 E_{k+1} 与 E_k 的体积比值上, 因为算法的迭代次数决定于这个比值.

本文提出的新椭球算法的思想来自更动约束的方法, 算法的出发点就与原椭球算法及其变形算法不同, 它不是基于求解 K-K-T 条件(1), 而是基于原规划(或对偶规划)约束与目标构成的不等式组:

$$\begin{cases} Ax \geqslant b \\ c^T x \leqslant \mu \\ x \geqslant 0 \end{cases} \quad \text{或} \quad \begin{cases} A^T w \leqslant c \\ b^T w \geqslant \lambda \\ w \geqslant 0 \end{cases}$$

这就形成了新椭球算法与原椭球算法的第一个不同点: 原椭球算法的标准模型是不等式系统 $Ax < b$, 它实际包括了原规划和对偶规划的所有不等式约束, 变量约束以及等式 $c^T x = b^T w$, 严格不等式的假设是靠摄动实现的. 而新算法的标准模型也是不等式系统 $Ax < b$, 但它仅包括原规划(或对偶规划)的不等式约束, 变量约束以及目标不等式, 同样可通过摄动, 有严格不等式的假设.

第二个不同点: 不论原椭球算法还是新椭球算法, 可行解集$\{x \in R^n \mid Ax < b\}$与 n 维空间的几何体$\{x \in R^n \mid \|x_j\| \leqslant 2^L,\ 1 \leqslant j \leqslant n\}$之交集均记为 P, 但是这两种算法中的 P 是有本质差别的.

第三个不同点: 原椭球算法第 k 步迭代的椭球 E_k, 其中心 x^k, 当 $x^k \in P$ 时, x^k 是最优解, 迭代终止; 当 $x^k \notin P$ 时, 利用某个不满足的约束不等式 $\alpha_i^T x^k \geqslant b_i$ ($Ax < b$ 的第 i 个不等式记为 $\alpha_i^T x < b_i$, 其中 $\alpha_i^T = (\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{in})$, $i = 1, 2, \cdots, m$.) 构造超平面 $\alpha_i^T x = \alpha_i^T x^k$ 形成的割 $\alpha_i^T x \leqslant \alpha_i^T x^k$, 割去 E_k 的一半, 我们称之为约束割. 新椭球算法第 k 步迭代椭球 E_k, 它的中心 $x_k \notin P$ 时, 同样构造以上的约束割; 当 $x^k \in P$ 时, x^k 只是可行解而不是最优解. 此时, 我们引进目标割: $c^T x \leqslant c^T x^k$. 同样割去 E_k 之一半. 这样, 在新算法运算过程中, 我们获得一系列可行解 x^k, $c^T x^k$ 单调趋于最优值(一旦满足精度则停机, 或一旦满意可人工停机), 或趋于 $-\infty$; 或得出无可行解的结论.

在实际的 LP 问题中, 比如资源分配问题, LP 的变量是有界的. 对于这类问题, LP 的结论要么有最优解, 要么无可行解. 同时这类问题的初始椭球可构造的比较小, 再利用深切割等技巧, 新算法是实际有效的, 研究新算法的意义也就在于此.

1 算法的基本思想

我们考虑如下的 LP 问题:

$$\begin{aligned} &\min\ c^T x \\ &\text{s.t.}\ \ Ax \geqslant b, \qquad x \geqslant 0. \end{aligned} \tag{4}$$

通过摄动和变号, 我们考虑不等式系统

$$\begin{cases} Ax < b \\ c^T x < \mu \end{cases}$$

假定 μ 足够大, 使 $P \subset \{c^T x < \mu\}$, 而且

$$\begin{cases} Ax < b \\ c^T x < \mu \end{cases} \tag{5}$$

仍记为 $Ax < b$.

为了今后的叙述简便, 我们将原椭球算法和新椭球算法都成立的若干结论引述如下. 首先引入如下的记号:

①以 d 为中心, r 为半径的 n 维欧氏空间中的球, 记为 $s(d, r)$.

$$s(d, r) = \{x \in R^n \mid (x - d)^T (x - d) \leqslant r^2\}$$

② $n \times n$ 非奇异矩阵 A 和点 d, 可以定义一个仿射变换, 记为 $T(A, d)$,

$$T(A, d) = A(x - d),\ \text{对一切}\ x \in R^n.$$

③ n 维欧氏空间中一个椭球, 记为 E,

$$E = \{x \in R^n \mid (x - d)^T A^T A (x - d) \leqslant 1\}$$

E 的中心为 d, 体积记为 $\mathrm{Vol}(E)$.

$$\mathrm{Vol}(E) = \det(A^{-1}) \cdot \mathrm{Vol}(s(0, 1))$$

④半椭球, 记为 $\frac{1}{2}E$,

$$\frac{1}{2}E = E \cap \{\alpha^T x \leqslant \alpha^T d\}$$

它是 E 与过其中心 d 点的由超平面 $\alpha^T x = \alpha^T d$ 决定的半空间 $\{\alpha^T x \leqslant \alpha^T d\}$ 之交.

⑤我们记在椭球算法中出现的包含 $\frac{1}{2}$E 的椭球为 $\tilde{E}$.

⑥ $L = \lceil 1 + \log m + \log n + \sum_{j=1}^{n}\{1 + \log(1 + |c_j|)\} + \sum_{i=1}^{m}\sum_{j=1}^{n}\{1 + \log(1 + |a_{ij}|)\} + \sum_{i=1}^{m}\{1 + \log(1 + |b_i|)\} \rceil$

这里 $\log t$ 是以 2 为底的对数, $\lceil \cdot \rceil$表示刚刚大于括号值的整数.

引理 1 若 $Ax < b + 2^{-L}\boldsymbol{e}$ 有解, 则 $Ax \leqslant b$ 也有解. 这里, $\boldsymbol{e}$ 为每个分量皆为 1 的 n 维列向量. 因此, 我们可以考虑有摄动因子 2^{-L}之后的严格不等式系统, 仍记为 $Ax < b$.

引理 2 若 $Ax < b$ 有解, 则必有 $|x_j| \leqslant 2^L (1 \leqslant j \leqslant n)$ 的解, 且其解集在此 n 维空间几何体中的体积至少为 $2^{-(n+1)L}$. 因此, 当第 k 次迭代椭球的体积 $\mathrm{Vol}(E_k) < 2^{-(n+1)L}$ 时, 可终止迭代, 这种情况下 $Ax < b$ 无可行解.

引理 3 若以 d 为中心的椭球 $E \supset P$, 但 $d \notin P$, 则一定存在一个过 d 点的超平面, 把 E 切成两个

半椭球, 而 P 必含在其中一个半椭球内, 记含 P 的半椭球为$\frac{1}{2}E$.

引理 4　每个半椭球$\frac{1}{2}E$ 包含在体积小于 $\mathrm{e}^{-\frac{1}{2(n+1)}}\mathrm{Vol}(E)$的椭球 $\tilde{E}$ 内.

引理 5　若以 d 为中心的椭球$E \supset P$, 而 $d \in P$, 则过 d 的目标函数等值超平面$c^Tx = c^Td$ 把E 切成两个半椭球, 同时也把 P 切成P_1, P_2, $P = P_1 \cup P_2$. 其中

$$P_1 = P \cap \{c^Tx \leqslant c^Td\},\ P_2 = P \cap \{c^Tx > c^Td\}$$

此时, 记含 P_1 的半椭球为$\frac{1}{2}E$.

引理 6　$\min\limits_{x \in P} c^Tx = \min\limits_{x \in P_1} c^Tx$.

显然, 割去 P 的子集P_2, 这种更动约束, 不会影响求最优解, 因为 d 是可行解, P_2 中任何可行解都不比它好. 即

$$\min_{x \in P_2} c^Tx \geqslant c^Td$$

$$\min_{x \in P} c^Tx = \min\left\{\min_{x \in P_1} c^Tx,\ \min_{x \in P_2} c^Tx\right\} = \min_{x \in P_1} c^Tx.$$

对于引理 5 的半椭球$\frac{1}{2}E$, 引理 4 的结论仍成立.

因此, 无论包含 P 的椭球 E 的中心 d 是否属于 P, 都存在椭球 $\tilde{E}$: $\tilde{E} \supset P$ 或 $\tilde{E} \supset P_1$, 而且 $\mathrm{Vol}(\tilde{E}) < \mathrm{e}^{-\frac{1}{2(n+1)}}\mathrm{Vol}(E)$.

当引理 5 情况出现时, 我们考虑不等式系统

$$\begin{cases} Ax < b \\ c^Tx < c^Td \end{cases}$$

以代替 $Ax < b$ 系统, 而且引理 1~ 引理 5 仍成立. 以上步骤可以重复进行.

这就指明了新椭球算法的基本思想: 如果第 k 步迭代时, 椭球 E_k 的中心$x^k \in P$, 则找到了一个可行解 x^k, 并进行目标割: $c^Tx < c^Tx^k$, 否则进行约束割: 对某个 i, $\alpha_i^Tx < \alpha_i^Tx^k$, 然后用较小的椭球 E_{k+1}替代 E_k, 重复进行, 每次迭代后, E_k 的体积至少以指数 $\mathrm{e}^{-\frac{1}{2(n+1)}}$的因子缩小, 这种迭代仅仅需要多项式次迭代就可结束. 当 $\mathrm{Vol}(E_k) < 2^{-(n+1)L_k}$时, 停止. 这里

$$L_k = L + 2 + \lceil \log(1 + |c^Tx^k|) \rceil .$$

引理 7　如果给定精度 $\delta > 0$, r_k 为E_k 最大半径, 且 $x^k \in P$, 则当 $r_k\|c\| < \delta$时, 迭代可以停止.

因为一切可能改善目标值的可行解都属于 $P_1 \subset E_k$, E_k 的中心为x^k, 而若 $x \in P_1$, 令 $\mathbf{p} = x - x^k$, $\mathbf{p}$ 是从 x^k 指向x 的向量.

$$c^Tx - c^Tx^k = c^Tp,$$

$$|c^Tx - c^Tx^k| \leqslant \|c\|\,\|\mathbf{p}\| \leqslant \|c\|\,r_k.$$

所以, 在 $x^k \in p$ 时, 不妨先判定引理 7 的停机条件, 再判定 $\mathrm{Vol}(E_k) < 2^{-(n+1)L_k}$的停机条件.

引理 8　如果给定精度 $\delta > 0$, r_k 为E_k 的最大半径, 而 $x^k \notin P$, 则当 $c^Tx^1 - c^Tx^k + r_k\|c\| < \varepsilon$时, 迭代可以停止. 其中 x^1 是迭代过程所得的最后一个可行解. 因此, 当 $x^k \notin P$ 时, 先判定引理 8 的停机条件, 再判定 $Vol(E_k) < 2^{-(n+1)L_k}$的停机条件.

总之, 我们求解式(4) LP 问题, 可以化为求解严格不等式组(5), 式(5)的新椭球算法步骤为:

①做初始椭球 $E_0 = s(0, 2^{2L})$, 令 $k = 0$.　$L_0 = L + 2 + \lceil \log(1+\mu) \rceil$, μ 充分大.

②若 E_k 的中心 $x^k \in \{Ax < b\}$, 则

(a) 得可行解 x^k, 记录 x^k.

(b) 若引理 7 条件成立, 停止; 若 $Vol(E_k) < 2^{-(n+1)L_k}$, 停止; 否则继续以下步骤.

(c) 利用目标割, 得 $\frac{1}{2}E_k = E_k \cap \{c^T x < c^T x^k\}$, 构造 E_{k+1}, 使 $E_{k+1} \supset \frac{1}{2}E_k$, 且 $Vol(E_{k+1}) < e^{-\frac{1}{2}(n+1)} Vol(E_k)$.

(d) $k = k+1$, 返回②.

③若 E_k 的中心 $x^k \notin \{Ax < b\}$,

(a) 若引理 8 条件成立, 停止; 若 $Vol(E_k) < 2^{-(n+1)L_k}$, 停止; 否则继续以下步骤(b).

(b) 利用约束割, 得 $\frac{1}{2}E_k = E_k \cap \{\alpha_i^T x < \alpha_i^T x^k\}$, 构造 E_{k+1} 使 $E_{k+1} \supset \frac{1}{2}E_k$, 且 $Vol(E_{k+1}) < e^{-\frac{1}{2}(n+1)} Vol(E_k)$.

(c) $k = k+1$, 返回②.

这里要特别指明新椭球算法与原椭球算法从第 k 步迭代到第 k+1 步迭代的不同点. 迭代的不同点是由于两种算法中 $Ax < b$ 和 P 的本质性差异造成的. 两种算法从初始至第 k 步迭代过程是一样的, 但是一旦 E_k 的中心 $x^k \in P$ 时, 原椭球算法得到的 x^k 是可行解也是最优解, 迭代终止; 而新椭球算法仅仅得到一个可行解 x^k, 为了得到更好的可行解, 直至最优解, 或判定 $c^T x$ 无界, 还需继续迭代. 此时, 本文引进的目标割起了作用, 利用目标割也同样割去 E_k 的一半, 也用较小的椭球 E_{k+1} 替代 E_k. 椭球 E_k 的体积也至少以指数 $e^{-\frac{1}{2}(n+1)}$ 的因子缩小, 系统迭代重新开始. 所以在新椭球算法中无论第 k 步迭代的椭球 E_k 的中心 x_k 是否属于 p, 都有相应的超平面去切割 E_k, 即目标割或约束割, 第 k+1 步都用较小的椭球 E_{k+1} 替代 E_k, 体积都以 $e^{-\frac{1}{2}(n+1)}$ 因子缩小.

如果 $Ax < b$ 系统无解, 原椭球算法和新椭球算法, 迭代过程完全一样, 直至判定无解时终止.

如果 $Ax < b$ 系统有解, 原椭球算法迭代到某步出现迭代椭球中心落入 P, 则终止. 但对原椭球算法系统这步迭代的出现比较难(即迭代次数要很大, 因为 P 很小). 而新椭球算法, 比较容易出现这种情况. 但是一旦出现只得到可行解, 还要继续迭代. 因此它将出现一串可解 $\{x^k\}$. 这串 x^k 到满足精度停机(或认为已得满意解可人工停机); 或判定 $c^T x^k \to -\infty$. 得一串可行解是新算法的优点之一, 这一串可行解 $\{x^k\}$ 还可以用于我们将要发表的一种新的 LP 算法的求解过程, 那种新的 LP 算法也可以用在新椭球算法的加速迭代上.

2　带目标割的椭球算法

如上所述, 新椭球算法有两种切割椭球 E_k 的办法: 一是约束割, 另一是目标割. 而且对于每步迭代皆取这两种切割之一. 切割后, 我们必须构造 $E_{k+1} \supset \frac{1}{2}E_k$, 使得 $Vol(E_{k+1}) < e^{-\frac{1}{2}(n+1)} Vol(E_k)$.

(1) 如果 E_k 取约束割, 构造 E_{k+1} 的作法与原椭球算法相同, 此时 $x^k \notin \{Ax < b\}$, 则存在下标 i, 使得 $\alpha_i^T x^k \geqslant b_i$.

$$\frac{1}{2}E_k = E_k \cap \{\alpha_i^T x < \alpha_i^T x^k\}$$

$$x^{k+1} = x^k - \frac{n}{n+1}(B_k \alpha_i / (\alpha_i^T B_k \alpha_i)^{\frac{1}{2}})$$

$$B_{k+1} = \frac{n^2}{n^2-1}\{B_k - \frac{n}{n+1}[B_k \alpha_i (B_k \alpha_i)^T / (\alpha_i^T B_k \alpha_i)]\}$$

$$E_{k+1} = \{(x - x^{k+1})^T B_{k+1}^{-1}(x - x^{k+1}) \leqslant 1\}.$$

这里 $E_{k+1} \supset E_k \cap \{\alpha_i^T x < b_i\}$. B_k^{-1} 是对称正定的, B_{k+1}^{-1}也是对称正定的. $\mathrm{Vol}(E_{k+1}) < \mathrm{e}^{-\frac{1}{2}(n+1)}\mathrm{Vol}(E_k)$.

(2) 如果 E_k 取目标割, 此时 $x^k \in \{Ax < b\}$, $\frac{1}{2}E_k = E_k \cap \{c^T x < c^T x^k\}$. 我们有:

$$x^{k+1} = x^k - \frac{1}{n+1}(B_k c / (c^T B_k c)^{\frac{1}{2}}).$$

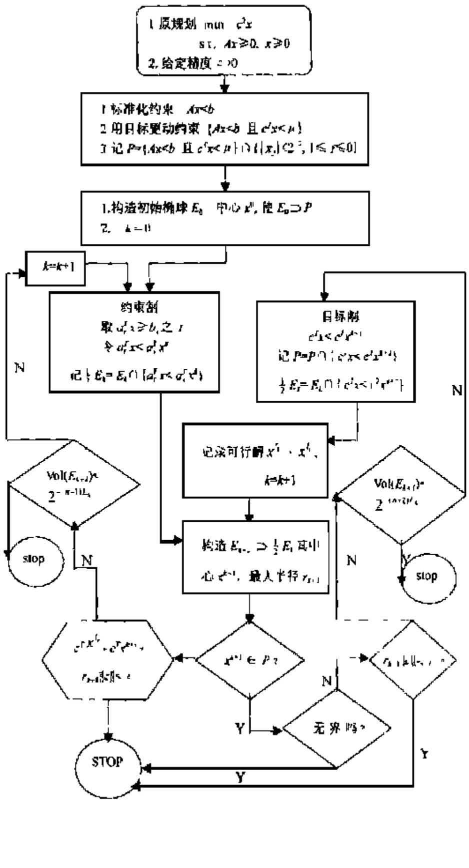
$$B_{k+1} = \frac{n^2}{n^2-1}\{B_k - \frac{n}{n+1}[B_k c(B_k c)^T / (c^T B_k c)]\}.$$

$$E_{k+1} = \{(x - x^{k+1}) B_{k+1}^{-1}(x - x^{k+1}) \leqslant 1\}.$$

这里 $E_{k+1} \supset E_k \cap \{c^T x < c^T x^k\}$. B_k^{-1} 是对称正定的, B_{k+1}^{-1}也是对称正定的. 同样 $\mathrm{Vol}(E_{k+1}) < \mathrm{e}^{-\frac{1}{2}(n+1)}\mathrm{Vol}(E_k)$.

(3) 初始化, 当 $k = 0$ 时, 取 $E_0 = s(0, 2^{2L})$, $B_0 = 2^{2L} I$, $x^0 = 0$.

(4) 新椭球算法迭代框图为:



3 新算法的性能及计算复杂性

原椭球算法最可取的是它是多项式时间算法, 但在解线性规划问题中它有两个严重的缺点. 一是收敛慢. 由于原椭球算法是对原规划和其对偶规划问题所有不等式约束和变量约束形成的可行域求解, 故初始椭球的体积可能是个天文数字, 而且其解在超平面 $c^T x = b^T w$ 中, 即使做了些摄动等处理, 相应的可行集仍然只有一个很小的体积, 这样, 迭代步数就可能非常大. 第二个缺点是若算法经过计算得出是无解的, 却不清楚原线性规划是不可行呢, 还是无界[2,5].

由于一般的线性规划问题可行集的体积不是很小, 故新算法很容易找到满足规划问题的可行解, 并由此得到一系列使目标趋于最优或趋于无界的可行解. 加入适当的停机判断, 还可使计算时间减少.

在现实生活里, 大多数问题的变量都是有界的, 由于起始椭球的体积不大, 且许多实际问题中只需一近似最优解(满意解) 即可, 而一系列逐步趋于最优的可行解的得到, 对于解决问题也非常有用. 因此本算法有重要的实用意义.

新算法引进了约束割、目标割的概念. 当求得的一个椭球中心 x^k 是一可行解时, 转为用目标割去切割此椭球, 新椭球仍按等比体积缩小. 此时, 新算法还是一个多项式时间算法吗?

新算法的终止判别条件实质上只有一个, 即 $\mathrm{Vol}(E_k) < 2^{-(n+1)L_k}$. 在算法框图中, 其它终止判别, 完全是为了节省计算量. 因为这些判别条件可能提前停机(达到精度了) .

(1) 若原问题无解, 新算法与原椭球算法的迭代过程完全一样, L_k 始终是 L_0, 此时新算法具有多项式复杂性.

(2) 若原问题有解, 且有有限最优解, 那么, $\{L_k\}$ 是一串递增数列, 且有上界 L. 每进行一次目标割: $c^T x < c^T x^k$; 即重新考虑一个新问题:

$$\begin{cases} Ax < b \\ c^T x < c^T x^k \end{cases}$$

它的规模为 L_k.

对此新问题实行椭球算法时, 初始椭球不必从很大的单位球 $S(0, 2^{2L_k})$ 做起, 由于对此新问题引理 1~ 引理 5 仍成立, 只不过 L 换成 L_k, 但可以理解为从初始椭球(包含可行解集) 开始已经经过若干次约束割, 形成目前的新问题. 对新问题实行椭球算法, 要么出现新问题无解情形, 此时算法具有多项式复杂性, 要么, 又形成另一个新问题. 然后, 再对新形式的新问题继续进行椭球算法(不必从头做起) . 每一次停机判别为:

$$\mathrm{Vol}(E_k) < 2^{-(n+1)L_k}$$

不难看出, 如果整个迭代过程的停机判别全都更替为:

$$\mathrm{Vol}(E_k) < 2^{-(n+1)L}$$

这样形成的椭球算法是多项式时间算法. 因此每个新问题用 $\mathrm{Vol}(E_k) < 2^{-(n+1)L_k}$ 判别的算法, 是多项式时间算法, 而且整个迭代过程无需知道 L .

(3) 若原问题有解, 但无有限最优解, 那么 $L_k \to \infty$. 这种情形, 在椭球迭代过程中, 我们可以清楚地看出 $c^T x^k \to -\infty$, 由此来判断迭代是否还要进行下去. 还可以对下面问题施行椭球算法

$$\begin{cases} Ax < 0 \\ c^T x \leqslant 1 \\ x \geqslant 0 \end{cases}$$

这个系统的任何一个解, 将产生一条无界的射线, 否则原问题存在有限最优解.

4 讨　论

Bland, Goldfarb and Todd 曾研究过从原规划出发的椭球算法, 即考虑形如

$$\begin{cases} Ax < b \\ c^T x < \mu \end{cases}$$

的不等式组用椭球算法求解问题. 但是, 一旦他们用椭球算法求得一个可行解(原椭球算法, 这时迭代终止), 他们马上走上了另一条路子, 这条路子是 Shor, Yudian and Nemirorshi 1976 年提出的解凸规划的滑动目标法, 他们的思路是很自然的. 因为前人已有这个现成的成果可利用. 我们采用更动约束的思想形成新问题, 继续用椭球算法, 我们引进的约束割和目标割, 实质上是把目标割看成一个可变动的约束割. 滑动目标法永远在可行域中进行, 但所得可行解的目标值不一定单调下降. 直到 1989 年在 Goldfarb and Todd 撰写的线性规划手册中[8], $\mu_0 = \min_{0 \leqslant l \leqslant k} c^t x^l$. 其中 x^l 为滑动目标法第 l 次迭代的可行解. 在我们的新算法中, 每一次迭代不一定都得到一个可行解, 但一旦得到可行解, 则有

$$c^T x^{i_1} > c^T x^{i_2} > c^T x^{i_3} > \cdots\cdots > c^T x^{i_k}$$

其中 x^{il} 是第 l 次目标割对应的可行解. 所以这两种算法是有本质区别的. 1979 年 Khachian 提出椭球算法之后, 国内外学者在改进椭球算法方面, 发展了几种变形算法, 比如深切割法、替代切割法、平行切割法以及用单纯性替代椭球的算法, 等等. 但在专著中不再提到 Bland 等人的滑动目标法, 说明滑动目标法不属椭球算法. 而我们提出的新算法, 完全是椭球算法. 这种用目标函数更动约束的思想, 形成了一种约束割与目标割交错进行的椭球算法, 自始至终一气呵成. 当非负变量有上界时, 令 $d = \max_{1 \leqslant i \leqslant n} d_i$, 其中 $0 \leqslant x_i \leqslant d_i$, 取初始椭球 $E_0 = S(0,\ 2^{2d})$ 即可.

最后, 我们指出线性规划可行解的获得, 除大 M 法和两段法(在单纯法和 Karmarkar 算法中采用的方法)外, 本文对有界变量的线性规划问题给出一个椭球算法获取可行解的方法, 而且可以获得一串趋于最优解的可行解. 这是一种实用的线性规划算法.

参　考　文　献

1 杨德庄. 灵活的运筹学和应用数学. 中国科学, 1995

2 L G Khachian. A Polynomal Algorithm in Linear Programming. Soviet Mathematics Doklady, 1979, 20: 191~ 194

3 G B Dantzig. Linear Programming and Extensions. Princeton New Jersey, Princeton University Press, 1963

4 V Klee, G L Minty. How Good is the Simplex Algorithm? In: O Shisha. Inequalities III. New York: Academic Press, 1972. 159~ 179

5 方述诚, S 普森普拉. 线性优化及扩展理论与算法. 北京: 科学出版社, 1994

6 N Karmarkar. A New Polynomial Time Algorithm for Linear Programming. Combinatorica, 1984, 4: 373~ 395

7 R G Bland, D Goldfard, M J Todd. The Ellipsoid Method : a Survey. Operations Research 29, 1981. 1039~ 1091

8 D Goldfarb, M J Todd. Linear Programming in Optimization, Handbook in Operations Research and Management Science. In: G L Nemhauser, A H G Rinnooy Kan. Amsterdam: Elserier- North Holland, 1989, 1:73~ 170

A New Ellipsoid Method

Yang Dezhuang　Zhang Minhong
(Graduate School, University of Science and Technology of China, Beijing 100039)

Zhang Lihua
(Institute for the History of Natural Science, Chinese Academy of Sciences, Beijing 100010)

Abstract　Base on the mathematical idea and method to alter constraints of a problem, a new algorithm for linear programming to solve has been given. This method is different from the primary ellipsoid method by L. G. Khachian. The step-by-step procedure in the new method is not only cutting down the half-ellipsoid, that does not contain the constraint set by the constraint inequalities named the constraint function cut, when the center of ellipsoid is not in the constraint set; but also cutting down the half-ellipsoid that contain the constraint set by the objective inequalities named the objective function cut when the center of this ellipsoid fall in the constraint set. The inequality set of this new algorithms consist of the constraint inequalities of primary program (or dual program) and objective inequality, which is small in scale, and does not resemble to consist of K-K-T optimality conditions for the primary ellipsoid method inequalities set which is large in scale. This new ellipsoid algorithm not only has characteristic of polynomial complexity, but also can get a series of monotone feasible solutions to approach the optimal solution if this solution exist. If we think to get a satisfied solution, we can terminate the iterative scheme at any time. So the new method is more practical and effective than the primary method. Because the variables and bounded, and the initial ellipsoid is not big in the vast majority of practical problems, the new algorithms is more feasible.

Key words　ellipsoid method, constraint function cut, objective function cut