国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第15卷第2期 1993年6月 Vol. 15 No. 2

线性分式规划原始单纯形算法有限性问题*

陈庆华 李建平
(系统工程与应用数学系)

摘 要 本文用一个数值例子说明用[1]和[2]中的原始单纯形算法求解退化的线性分式规划(LFP)可能会出现基循环,从而得不到最优解。于是就此情形引入了Bland规则,并建立了有限性算法。

关键词 线性分式规划,最优性条件,单纯形算法,Bland规则
分类号 O221.1

线性分式规划(LFP)是非线性规划的一个特殊类型,其目标函数是两个线性函数之商,约束是一组线性不等式,具体形式如下:

$$\text{(LFP)}\quad \max Z=(c^Tx+c_0)/(d^Tx+d_0) \tag{1}$$

$$\text{s.t.}\begin{cases}A_x\leqslant b\\ x\geqslant 0\end{cases} \tag{2}$$

式中 A 是 $m\times n$ 矩阵,b 是 m 维向量且 $b\geqslant 0$,c,d,x 是 n 维向量,c_0,d_0 是数量常数。

记 $P=\{x|Ax\leqslant b,\ x\geqslant 0\}$ 为LFP (1) − (2) 的可行解集,假定

(1) $d^Tx+d_0>0$,$\forall\ x\in P$

(2) P 是有界凸多面体

B. Martos[1]称满足(1)和(2)的LFP为简单问题。本文只讨论LFP简单问题(以下简称LFP)。显而易见,LFP必有最优解,并且有

定理1 LFP必在凸多面体 P 的极点上取到最优解。(证明见文献[1])。

根据定理1,我们可以从LFP的可行域的极点上去寻找LFP的最优解。B. Martos[1]和K. Swarup[2]各自独立地建立了求解LFP的原始单纯形算法。这一算法和线性规划单纯形法十分相似,只是检验数的定义不同,因而它同样具有单纯形法简单、实用和优美的特点。但我们注意到文献[3]和[4]中的算法都只适用于非退化情况。我们给出了一个退化LFP的数值例子,表明应用他们的算法会导致LFP单纯形迭代中出现基循环,从而不能在有限步内得到此问题的最优解。于是我们在他们的算法中引入线性规划中避免循环的Bland规则,同时证明Bland规则对解决LFP有限性问题同样有效,从而得到了求解LFP的有限性算法,它同时适用于求解退化与非退化的LFP问题。

* 1992年4月8日收稿

1 原始单纯形算法

考虑 LFP（1）－（2）的标准形式

$$(\text{LFP}) \qquad \max Z = (c^T x + c_0)/(d^T x + d_0) \tag{3}$$

$$\text{s.t.} \begin{cases} A_1 x = b \\ x \geqslant 0 \end{cases} \tag{4}$$

式中 $\operatorname{rank} A_1 = m$，$b \geqslant 0$.

由于约束是线性的，因此凡不涉及目标函数（3）的有关于线性规划的概念在这里仍然适用。

设 B 为可行基，用 J 表示基变量的指标集，用 R 表示非基变量的指标集，并记 $I=\{1, 2, \cdots, m\}$. LFP（3）－（4）对应的典则形式为

$$(\text{LFP}) \quad \max Z = \frac{u(x)}{v(x)} = \frac{u_0 - \sum\limits_{j \in R} u_j x_j}{v_0 - \sum\limits_{j \in R} v_j x_j} \tag{5}$$

$$\text{s.t.} \begin{cases} x_{J_i} = y_{i0} - \sum\limits_{j \in R} y_{ij} x_j, \quad i = 1, 2, \cdots, m \\ x_j \geqslant 0, \quad j = 1, 2, \cdots, n \end{cases}$$

称

$$x_{J_i} = y_{i0}, \quad i = 1, 2, \cdots, m; x_j = 0, \quad j \in R \tag{6}$$

为对应于可行基 B 的基本可行解，其中 $x_{J_i} = y_{i0} \geqslant 0$，$i=1, 2, \cdots, m$. 称

$$\lambda_j = u_0 v_j - v_0 u_j, j \in R \tag{7}$$

为对应于可行基 B 关于非基变量 x_j 的检验数。

定理 2 设 x^0，其分量 $x_{J_i} = y_{i0}$，$i \in I$；$x_j = 0$，$j \in R$ 是 LFP 的一个基本可行解，若满足

$$\lambda_j \leqslant 0, j \in R \tag{8}$$

则 x^0 是 LFP 的最优解。称（8）为 LFP 的最优性条件。（证明见文献 [2]）.

由定理 2，如果对某个可行基 B，最优性条件（8）成立，那我们就得到最优解。否则我们通过类似于线性规划的单纯形迭代得到一个新的可行基及改进的基本可行解。

求解 LFP（1）－（2）的原始单纯形算法具体步骤如下：

第一步：选取初始可行基及初始基本可行解。

第二步：（最优性检验）若所有 $\lambda_i \leqslant 0$，$j \in R$，则算法结束，得到最优解：

$$\begin{cases} x_{J_i} = y_{i0}, \quad i \in I \\ x_j = 0, \quad j \in R \end{cases}$$

否则转第三步。

第三步：（选取进基变量）若有 $\lambda_k > 0$，则取 x_k 为进基变量。（若有多个 $\lambda_k > 0$ 时，可任选一个 x_k，通常取 $\lambda_k = \max\{\lambda_j | \lambda_j > 0, j \in R\}$ 对应的 x_k 作为进基变量。）

第四步：（选取离基变量） 由 $\frac{y_{r0}}{y_{rk}} = \min\limits_{i \in I}\left\{\frac{y_{i0}}{y_{ik}} \middle| y_{ik} > 0\right\}$ 确定 x_{J_r} 为离基变量。（若上述最小值在几个比值上同时达到，可任取一个。）

第五步：（迭代）作 (r,k) 旋转变换，得新的可行基及新的基可行解。按（7）计算各非基变量的检验数。转第二步。

迭代过程可以在单纯形表上实现。在这里我们采用不同于文献［1］、［2］的一种简略的单纯形表格式。

表 1　迭代前的 T(B)值

$T(B)$		常数列	非基变量 ↓	
			$-x_j$	$-x_k$
检验行	Z	u_0/v_0	λj	λk
目标行	u	u_0	u_j	u_k
	v	v_0	v_j	v_k
约束行	x_{J_i}	y_{i0}	y_{ij}	y_{ik}
	$\leftarrow x_{J_r}$	y_{r0}	y_{rj}	y_{rk}

其中 x_k 为进基变量，x_{J_r}为出基变量，y_{rk}为枢纽元素。

表 2　迭代后 T($\hat{B}$)值

$T(B)$	常数列	$-x_j$	$-x_{J_r}$
z	$\hat{u}_0/\hat{v}_0$	$\hat{\lambda}_j$	$\hat{\lambda}_{J_r}$
u	$u_0-\frac{y_{r0}u_k}{y_{rk}}$	$u_j-\frac{y_{rj}u_k}{y_{rk}}$	$-\frac{u_k}{y_{rk}}$
v	$v_0-\frac{y_{r0}v_k}{y_{rk}}$	$v_j-\frac{y_{rj}v_k}{y_{rk}}$	$-\frac{v_k}{y_{rk}}$
x_{J_i}	$y_{i0}-\frac{y_{r0}y_{ik}}{y_{rk}}$	$y_{ij}-\frac{y_{rj}y_{ik}}{y_{rk}}$	$-\frac{y_{ik}}{y_{rk}}$
x_k	$\frac{y_{r0}}{y_{rk}}$	$\frac{y_{rj}}{y_{rk}}$	$\frac{1}{y_{rk}}$

算法正确性由下面两个定理保证。定理的证明是平凡的。

定理 3　当 $\lambda_k>0$ 时，至少存在一个 $y_{ik}>0$，$i\in I$.

定理 4　对 LFP 应用上述算法，由一可行基开始，每经过一次单纯形迭代都得到一个基本可行解，且目标函数非降。特别，在非退化情况下，目标函数值严格上升。

由基本假设（1）和（2），我们知道，LFP（1）—（2）只有有限个基本可行解，由定理 1 和定理 4，在非退化情况下，只须经有限步迭代就可以得到问题的最优解。

自然要问，在退化情况下是否有同样的结论？下一节给出的数值例子不能保证有限性。

2　一个出现基循环的例子

1955 年 E. M. L. Beale 构造了一个在线性规划的原始单纯形算法中出现基循环的例子[6]。我们将其目标函数改造一下，可说明 LFP 原始单纯形算法会出现同样的情况。

$$\text{(LFP)}\qquad \max Z=\frac{1+\frac{3}{4}x_1-150x_2+\frac{1}{50}x_3+2x_4}{1+8x_4} \tag{9}$$

$$\text{s. t.}\begin{cases} \frac{1}{4}x_1 - 60x_2 - \frac{1}{25}x_3 + 9x_4 + x_5 = 0 \\ \frac{1}{2}x_1 - 90x_2 - \frac{1}{50}x_3 + 3x_4 + x_6 = 0 \\ x_3 + x_7 = 1 \\ x_1, x_2, \cdots, x_7 \geqslant 0 \end{cases} \tag{10}$$

这是一个退化的 LFP，有明显的初始基本可行解$(0,0,0,0,0,0,1)^T$，其中 x_5, x_6, x_7 为基变量。施行上一节中的算法得：(见表 3～表 6)

表 3 初始值

$T(B_0)$	常数列	$\downarrow$ $-x_1$	$-x_2$	$-x_3$	$-x_4$
z	1	3/4	−150	1/50	−6
u	1	−3/4	150	−1/50	−2
v	1	0	0	0	−8
$\leftarrow x_5$	0	1/4	−60	−1/25	9
x_6	0	1/2	−90	−1/50	3
x_7	1	0	0	1	0

表 4 第一次迭代结果

$T(B_1)$	常数列	$-x_5$	$\downarrow$ $-x_2$	$-x_3$	$-x_4$
z	1	−3	30	7/50	−33
u	1	3	−30	−7/50	25
v	1	0	0	0	−8
x_1	0	4	−240	−4/25	36
$\leftarrow x_6$	0	−2	30	3/50	−15
x_7	1	0	0	1	0

表 5 第四次迭代结果

$T(B_6)$	常数列	$\downarrow$ $-x_5$	$-x_6$	$-x_1$	$-x_2$
z	1	1	−1	1/2	−120
u	1	5/3	−13/3	−5/2	440
v	1	8/3	−16/3	−2	320
$\leftarrow x_3$	0	50	−150	−125/2	10500
x_4	0	1/3	−2/3	−1/4	40
x_7	1	−50	150	125/2	−10500

表 6 第六次迭代结果

$T(B_6)$	常数列	$-x_3$	$-x_4$	$-x_1$	$-x_2$
z	1	1/50	−6	3/4	−150
u	1	−1/50	−2	−3/4	150
v	1	0	−8	0	0
x_5	0	−1/25	9	1/4	−60
x_6	0	−1/50	3	1/2	−90
x_7	1	1	0	0	0

表 $T(B_6)$与 $T(B_0)$完全相同，即迭代六次后又出现了 B_0，其按同样规则继续下去，必将导致循环，得不到最优解。

我们知道，1977 年，R.G. Bland 提出的最小下标规则是线性规划原始单纯形算法中避免基循环的一个简单办法。下一节我们将证明 Bland 规则对非线性规划的 LFP 单纯形算法避免基循环仍然适用。

3 Bland 规则与 LFP 有限算法

如果在第一节的算法中第三步，第四步采用 Bland 规则来确定进基变量和离基变量，则有相应变化：第三步：令 $k = \min\limits_{j\in R}\{j \mid \lambda_j > 0\}$，取 x_k 为进基变量。

第四步：令
$$\theta = \min_{i\in I}\left\{\frac{y_{i0}}{y_{ik}} \,\middle|\, y_{ik} > 0\right\}$$
$$J_r = \min_{i\in I}\left\{J_i \,\middle|\, \frac{y_{i0}}{y_{ik}} = \theta\right\}$$

取 x_{J_r} 为离基变量。

现在我们可以用这一算法重新求解第二节中 LFP(9)－(10)，来证实 Bland 规则避免

了基循环。

按第三步,第四步,前 4 次迭代相同。第 5 次迭代(即在 $T(B_4)$中),进基变量应选 x_1 而不是 x_5. 经 6 次迭代得该问题的最优解$\left(\frac{1}{25},0,1,0,\frac{3}{100},0,0\right)^T$,最优值为 21/20.

定理 5 上述求解 LFP 算法必在有限步结束。

证明 根据定理 4,我们知道上述计算过程是由可行基到可行基的迭代,同时目标函数不断改进,因此只须证明计算过程中基不重复出现,那么由于基的总数是有限的,算法必在有限步结束。

用反证法。若不然,某 LFP 运用上述算法时,某可行基重复出现。即

$$T(B_1) \to T(B_2) \to \cdots \to T(B_\omega) \to T(B_1)$$

也就是说,算法进入"循环"。

将变量指标集 $\{1, 2, \cdots, n\}$ 剖分为子集 F, H, G 之和。

$j\in F$,当且仅当 x_j 是所有 B_t, $t=1, 2, \cdots, \omega$. 的基变量,称其为固定基变量指标集。

$j\in H$,当且仅当 x_j 是所有 B_t, $t=1, 2, \cdots, \omega$. 的非基变量,称其为固定非基变量指标集。

$j\in G$,当且仅当 x_j 既是某 B_μ 的基变量又是某 B_ν 的非基变量,称其为循环变量指标集。

将行指标集 $\{1, 2, \cdots, m\}$ 划分为子集 L, M 之和,使得:

L 是对应固定基变量的行指标集,M 是对应循环变量的行指标集。

设由 $T(B_t)$变换到 $T(B_{t+1})$的旋转行为 $r(t)$,离基变量的指标为 r_t,进基变量的指标为 k_t,其中 $t=1, 2, \cdots, \omega$. 而 $B_{\omega+1}=B_1$. 由上述定义知

$$G=\bigcup_{t=1}^{\omega}\{k_t\}=\bigcup_{t=1}^{\omega}\{r_t\},M=\bigcup_{t=1}^{\omega}\{r(t)\}$$

因目标函数非降,故在循环过程中目标函数保持不变,即$\frac{u_0^t}{v_0^t}=\frac{u_0^{t+1}}{v_0^{t+1}}$, $t=1, 2, \cdots, \omega-1$. 注意到

$$u_0^{t+1}=u_0^t-\theta_t u_{k_t},\quad v_0^{t+1}=v_0^t-\theta_t v_{k_t}$$

则

$$u_0^t v_0^{t+1}-v_0^t u_0^{t+1}=-\theta_t\lambda_{k_t}=0,\text{ 其中 }\lambda_{k_t}>0$$

故

$$\theta_t=0,t=1,2,\cdots,\omega \tag{11}$$

所以

$$y_{r(t)0}=0,t=1,2,\cdots,\omega \tag{12}$$

由迭代规则,常数列保持不变,即

$$\begin{cases}y_{i0}^1=y_{i0}^2=\cdots=y_{i0}^\omega\\ u_0^1=u_0^2=\cdots=u_0^\omega,v_0^1=v_0^2=\cdots=v_0^\omega\end{cases} \tag{13}$$

从而由 (12), (13) 有

$$y_{i0}^t=0,i\in M,t=1,2,\cdots,\omega \tag{14}$$

令 $q=\max\{j|j\in G\}$,设 $k_f=r_e=q$. 在 $T(B_f)$中,由旋转列选取规则有

$$\lambda_q^f>0,\lambda_j^f\leqslant 0,j\in G\cap R^f\backslash\{q\} \tag{15}$$

在 $T(B_e)$中，记 $k_e=k$，$r(e)=r$，则 $J_r^e=q$，$\lambda_k^e>0$. 因对所有 $i\in M$，有 $y_{i0}^e=0$，故由旋转行选取规则有

$$y_{rk}^e > 0, y_{ik}^e \leqslant 0, i \in M\backslash\{r\} \tag{16}$$

作辅助函数

$$g(x) = u_0 v(x) - v_0 u(x) = -\sum_{j\in R}\lambda_j x_j \tag{17}$$

（由于不涉及固定基变量，为简化讨论，无妨把 L 对应的行从表中划去）。取 $\tilde{x}$ 如下：

$$\begin{cases}\tilde{x}_k = 1 \\ \tilde{x}_{J_i} = - y_{ik}^e, i \in M, J_i \in J^e \\ \tilde{x}_j = 0, \text{其它}\end{cases} \tag{18}$$

则

$$g(\tilde{x})|_{T(B_e)} = -\sum_{j\in R^e}\lambda_j^e\tilde{x}_j = - \lambda_k^e < 0 \tag{19}$$

而

$$\begin{aligned} g(\tilde{x})|_{T(B_f)} &= -\sum_{j\in R^f}\lambda_j^f\tilde{x}_j \\ &= -\sum_{j\in H}\lambda_j^f\tilde{x}_j - \sum_{j\in G\cap R^f\backslash\{q\}}\lambda_j^f\tilde{x}_j - \lambda_q^f\tilde{x}_q \end{aligned} \tag{20}$$

由(18)，我们有 $\tilde{x}_j=0$，$j\in H$;$\tilde{x}_j\geqslant 0, j\in G\cap R^f\backslash\{q\}$，因对任意 $j\in G\cap R^f\backslash\{q\}$,必有 $j\in G\cap R^e$ 或者 $j\in\{J_i|J_i\in J^e, i\in M\backslash\{r\}.\}$，于是 $\tilde{x}_j$ 要么取 0 或 1，要么取$-y_{ik}^e\geqslant 0$，$i\in M\backslash\{r\}$. 再由(15)有

$$g(\tilde{x})|_{T(B_f)} \geqslant - \lambda_q^f\tilde{x}_q = \lambda_q^f y_{rk}^e > 0 \tag{21}$$

注意到 $g(x)$在 $\tilde{x}$ 处按 $T(B_e)$和 $T(B_f)$计算得到两个不同的结果，故 (19)，(21) 矛盾。

所以，在单纯形迭代中，基不重复出现（证毕）。

参 考 文 献

1 Martos B. Hyperbolic programming. Navel Res. logist Quart., 1964, 11: 135～155

2 Swarup K. Linear Fractional Functional Programming. Opns. Res., 1965, 13: 1029～1036

3 张干宗. 线性规则. 武汉大学出版社，1988

4 陈庆华，谢政. 整数规划. 国防科技大学出版社，1992

On Finitness of the Primal Simplex Algorithm of Linear Fractional Programming

Chen Qinghua　Li Jianping

(Department of System Engineering and Applied Mathematics)

Abstract

In this paper, a numerical example of linear fractional programming (LFP) has been constructed in which a finite sequence of degenerate bases obtained by the LFP's primal simplex algorithm in referenes [1] and [2] may yield basis cycling and hence no optimum solution could be got. So, We propose the finite algorithm using Bland's rule.

Key words linear fractional programming, optimality condition, simplex algorithm, Bland's rule