$$\begin{aligned}
\text{obj}\quad & 3x_1+2x_2+x_4 \\
\text{s. t.}\quad & x_1+2x_2+x_3 = 15 \\
& 2x_1+5x_3 = 18 \\
& 2x_1+4x_2+x_3+x_4 = 10 \\
& x_i \geqslant 0, i = 1,2,3,4
\end{aligned}$$

Solution $B=\begin{bmatrix} c^T \\ A \end{bmatrix}^{-1}=\begin{bmatrix} 3 & 2 & 0 & 1 \\ 1 & 2 & 1 & 0 \\ 2 & 0 & 5 & 0 \\ 2 & 4 & 1 & 1 \end{bmatrix}=\begin{bmatrix} 0.5 & 0.5 & 0 & -0.5 \\ -0.15 & 0.35 & -0.1 & 0.15 \\ -0.2 & -0.2 & 0.2 & 0.2 \\ -0.2 & -2.2 & 0.2 & 1.2 \end{bmatrix}$

$$U= B(1,0,0,0)^T = (0.5, -0.15, -0.2, -0.2)^T$$

$$V= B(0,15,18,10)^T = (2.5, 4.95, 2.6, -17.4)^T$$

$$u_1 > 0,\quad -v_1/u_1 = -5,\quad S_1 = -5$$

$$u_2 < 0,\quad -v_2/u_2 = 33,\quad u_3 < 0,\quad -v_3/u_3 = 13$$

$$u_4 < 0,\quad -v_4/u_4 = -87,\quad S_2 = -87.$$

Since $S_1 > S_2$, the LP problem is infeasible.

Acknowlegement I am grateful to Katta G. Murty who read my paper and encouraged me to extend Algorithm SP in this paper to the generic LP problem. It is still an open question to create a practically strongly polynomial algorithm.

References

1 Edmonds J. System of distinct representatives and linear algebra. J Res Bur Standards, 1967, 71 (B): 241~245

2 Tardos E. A strongly polynomial algorithm to solve combinatorial linear programs. Operations Research, 1986, 34(2): 250~256

一类线性规划问题的强多项式算法

吉训仁*

摘 要 对一类线性规划问题提出了一个强多项式算法．此算法可进行双向搜索．可行解集、目标函数的两个目标值以及相应的最优解，全部可行基与最优基可以一步求得，无需迭代．算法的复杂性为 $O(n^3+n^2+n)$，其中 n为线性规划问题变量的个数．

关键词 线性方程组，线性不等式，线性规划，多项式算法，强多项式算法

分类号 O 211.1

* 中山大学管理学院，广州 510275

第 36卷 第 1期 1997年 1月
中山大学学报 (自然科学版) ACTA SCIENTIARUM NATURALIUM UNIVERSITATIS SUNYATSENI
Vol. 36 No. 1 Jan. 1997

A STRONGLY POLYNOMIAL ALGORITHM FOR A CLASS OF LINEAR PROGRAMMING PROBLEMS[20]

Ji Xunren

(School of Management, Zhongshan University, Guangzhou 510275)

Abstract This paper presents a strongly polynomial algorithm for a class of linear programming problems. With this algorithm, the bi-directional search can be implemented. The set of feasible solutions, two target values and the corresponding optimal solutions of the objective function, all the feasible bases and the optimal bases can be obtained with no iterations. The complexity of the algorithm is $O(n^3+n^2+n)$, where n is the number of variables.

Keywords simultaneous linear equations, linear inequality, linear programming, polynomial algorithm, strongly polynomial algorithm

Classification Number O 211.1

We consider the linear programming problem:

$$\begin{aligned} &\text{obj} \quad c^T x \\ &\text{s.t.} \quad Ax=b, \quad x \geqslant 0 \end{aligned} \tag{1}$$

where obj= "min"| "max", $A\in R^{(n-1)*n}$, $b\in R^{n-1}$, c and x belong to R^n. In the following discussion we assume:

$$\text{rank}[c, A^T]^T = n \tag{2}$$

Transform linear programming problem (1) into simultaneous linear equations with a single parameter in the right hand side. Having inversed a matrix and solved a group of linear inequalities, we can get the following results: ① The set of the feasible solution; ② Two target values and the corresponding optimal solutions of the objective function; ③ All the feasible bases and the optimal bases.

No iterations occurred in our algorithm.

1 Algorithm

This section consists of two algorithms: Sub-algorithm INEQ(U, V, S_1, S_2) and main algorithm SP. The former is for solving the linear inequalities $US+V\geqslant 0$, while the latter

[20] 收稿日期: 1996-03-18 吉训仁,男,58岁,副教授

is for solving LP problem(1).

Sub-algorithm INEQ$(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S}_1,\boldsymbol{S}_2)$

Input: $U=(u_1,u_2,\cdots,u_n)^T$, $V=(v_1,v_2,\cdots,v_n)^T$.

Output: Two real numbers S_1,S_2 and two integer sequences:

$$i1,i2,\cdots,ip,\quad (1\leqslant i1,i2,\cdots,ip\leqslant n)$$

$$j1,j2,\cdots,jq,\quad (1\leqslant j1,j2,\cdots,jq\leqslant n)$$

Step 1 $S_1=-\infty$, $S_2=+\infty$, $i=1$.

Step 2 Case 1: $u_i=0$ and $v_i<0$, Let $S_1=1, S_2=0$, return;

Case 2: $u_i=0$ and $v_i>0$, go to Step 3;

Case 3: $u_i>0$, let $s=-v_i/u_i$, $S_1=\max(s,S_1)$;

Case 4: $u_i<0$, let $s=-v_i/u_i$, $S_2=\min(s,S_2)$.

Step 3 $i=i+1$. if $i>n$, return; otherwise, go to Step 2.

Note: During the computing of S_1 and S_2, there might be the following expressions:

$$S_1=\max_{u_i>0}(-v_i/u_i)=-v_{i1}/u_{i1}=-v_{i2}/u_{i2}=\cdots=-v_{ip}/u_{ip} \tag{3}$$

$$S_2=\min_{u_j<0}(-v_j/u_j)=-v_{j1}/u_{j1}=-v_{j2}/u_{j2}=\cdots=-v_{jq}/u_{jq} \tag{4}$$

Main algorithm SP

Input: A,b,c.

Output:

(1) The set of the feasible solution.

(2) Two target values and the corresponing optimal solutions of the objective function.

(3) All the feasible bases and the optimal bases.

Step 1 Use the Gaussian elimination method to inverse the matrix $[c,A^T]^T$. Let $B=\left([c,A^T]^T\right)^{-1}$.

Step 2 $U=B(1,0,\cdots,0)^T=(u_1,\cdots,u_n)^T$, $V=B(0,b_1,\cdots,b_{n-1})^T=(v_1,\cdots,v_n)^T$.

Step 3 Calling the sub-algorithm INEQ(U,V,S_1,S_2) which returns two real numbers S_1,S_2 and expressions(3),(4).

Step 4 End.

2 Proof of the algorithm

Lemma 1 If $S_1>S_2$, LP problem(1) is infeasible.

Lemma 2 If $S_1\leqslant S_2$, then

(1) The set of the feasible solution of LP problem (1) is:

$$x=US+V,\qquad S\in[S_1,S_2]$$

(2) $\min c^Tx=S_1$, $x_{\min}=US_1+V$; $\max c^Tx=S_2$, $x_{\max}=US_2+V$.

(3) According to (3) and (4), create two group of sequences:

$$(G.P)\begin{cases} x_1,\cdots,x_{i1-1},x_{i1+1},\cdots,x_n, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_1,\cdots,x_{ip-1},x_{ip+1},\cdots,x_n. \end{cases}$$

$$(G.Q)\begin{cases} x_1,\cdots,x_{j1-1},x_{j1+1},\cdots,x_n, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_1,\cdots,x_{jq-1},x_{jq+1},\cdots,x_n. \end{cases}$$

If obj is "min", then (G. P) is the p optimal bases, and (G. Q) q feasible bases;

If obj is "max", then (G. Q) is the q optimal bases, and (G. P) p feasible bases.

Proof of Lemma 1: Rewrite LP problem (1) as normal simultaneous linear equations with a single parameter in the right hand side:

$$\begin{array}{lll} \text{obj} & c^T x & \\ \text{s. t.} & Ax=b & \Rightarrow \\ & x\geqslant 0 & \end{array} \quad \begin{array}{ll} c^T x = S & \\ Ax=b & \Rightarrow \\ x\geqslant 0 & \end{array}$$

$$[c,A^T]^T x=(1,0,\cdots,0)^T S+(0,b_1,\cdots,b_{n-1})^T \tag{5}$$

Here S is a parameter determined by the sub-algorithm INEQ(U,V,S_1,S_2).

Since rank$[c,A^T]^T=n$, we have $B=([c,A^T]^T)^{-1}$. The solution of (5) should be

$x=B(1,0,\cdots,0)^T S+B(0,b_1,\cdots,b_{n-1})^T=(u_1,\cdots,u_n)^T S+(v_1,\cdots,v_n)^T=US+V$.

Calling the sub-algorithm INEQ(U,V,S_1,S_2), there might be two cases which make $S_1>S_2$:

Case 1: $S_1=1$, $S_2=0$. It means that there exists $u_i=0$ and $v_i<0$. So there must be no real number S which makes $x_i=u_iS+v_i\geqslant 0$ hold.

Case 2: For all the positive components $u_i>0$ and $S\in[S_1,\infty)$, $x_i=u_iS+v_i\geqslant 0$ hold; Similarly, for all the negative components $u_j<0$ and $S\in(-\infty,S_2]$, $x_j=u_jS+v_j\geqslant 0$ hold. If $S_1>S_2$, which means that the intersection $(-\infty,S_2]\cap[S_1,\infty)$ is empty, there never exits real number S which makes $x=US+V\geqslant 0$ hold.

Proof of Lemma 2: If $S_1\leqslant S_2$, the intersection $(-\infty,S_2]\cap[S_1,\infty)$ is nonempty. It means that each number in the interval $[S_1,S_2]$ makes $x=US+V\geqslant 0$ hold. Outside the interval $[S_1,S_2]$, no number makes $x=US+V\geqslant 0$ hold. So we have:

(1) The set of the feasible solution is $x=US+V$, $S\in[S_1,S_2]$.

(2) $\min c^T x=S_1, x_{\min}=US_1+V$; $\max c^T x=S_2$, $x_{\max}=US_2+V$.

Now prove conclusion (3) in Lemma 2. For the sake of simplicity, let obj be "min", $p=q=1$ in (3) or (4), both S_1 and S_2 are finite. By the above proof, we have $\min c^T x=S_1=-v_{i1}/u_{i1}$ and the optimal solution $x_{\min}=U(-v_{i1}/u_{i1})+V$. The $i1$ th component of $x_{\min}$ is zero, and the others positive. So, the sequence

$$x_1,\cdots,x_{i1-1},x_{i1+1},\cdots,x_n \tag{6}$$

is an optimal basis. Besides, $x=US_2+V=U(-v_{j1}/u_{j1})+V$ is a feasible solution. The $j1$th component of this feasible solution is zero and the others positive. So, the sequence

$$x_1,\cdots,x_{j1-1},x_{j1+1},\cdots,x_n \tag{7}$$

is a feasible basis. With this discussion it is easy to prove the general situation.

Corollary If LP problem (1) is nondegenerate, there are at most two bases.

Theorem Algorithm SP for LP problem (1) is a strongly polynomial algorithm.

Proof The number of arithmetic operations in each step in Algorithm SP is listed below: Step 1 $O(n^3)$, Step 2 $O(n^2)$ and Step 3 $O(n)$.

The size of numbers occurring during the Gaussian elimination in Step 1 in Algorithm SP is polynomially bounded by the size of the matrix $[c, A]^T$ (Edmonds, 1967). There is no iterations in our algorithm. So the complexity of Algorithm SP is $O(n^3 + n^2 + n)$. Algorithm SP meets the criterion of the strongly polynomial algorithm (Tardos, 1985). Therefore the Theorem above holds.

3 Geometrical interpretation of the Algorithm

When S changes from $-\infty$ to $+\infty$, the hyperplane $c^T x = S$ scans the space of the feasible solution— one segment. The intersection between the segment and the moving hyperplane $c^T x = S$ is also this segment. During the movement of the hyperplane $c^T x = S$, what it touches at first is the point $x = US_1 + V$, and finally the point $x = US_2 + V$.

4 Examples

Example 1: obj $2x_1 + x_2 + x_3$

$$\text{s. t.}\quad \begin{aligned} 2x_1 - x_2 + x_3 &= 2 \\ x_1 \qquad + x_3 &= 2 \\ x_i \geqslant 0,\ i &= 1, 2, 3. \end{aligned}$$

Solution:
$$\begin{bmatrix} c^T \\ A \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & -1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} c^T \\ A \end{bmatrix}^{-1} = \begin{bmatrix} 0.5 & 0.5 & -1 \\ 0.5 & -0.5 & 0 \\ -0.5 & -0.5 & 2 \end{bmatrix}$$

$$U = B(1,0,0)^T = (0.5, 0.5, -0.5)^T, \qquad V = B(0,2,2)^T = (-1, -1, 3)^T$$

$$u_1 > 0,\ -v_1/u_1 = 2,\ u_2 > 0,\ -v_2/u_2 = 2,\ u_3 < 0,\ -v_3/u_3 = 6,$$

$$S_1 = \max_{u_i > 0}(-v_i/u_i) = 2, \qquad S_2 = \min_{u_j < 0}(-v_j/u_j) = 6,$$

$S_1 < S_2$, so we have:

(1) The set of the feasible solution is: $x = (0.5, 0.5, -0.5)^T S + (-1, -1, 3)^T$, $S \in [2, 6]$.

(2) $\min c^T x = 2$, $x_{\min} = (0.5, 0.5, -0.5)^T 2 + (-1, -1, 3)^T = (0, 0, 2)^T$,

$\max C^T x = 6$, $x_{\max} = (0.5, 0.5, -0.5)^T 6 + (-1, -1, 3)^T = (2, 2, 0)^T$.

(3) $S_1 = -v_1/u_1 = -v_2/u_2 = 2$, $S_2 = -v_3/u_3 = 6$

Case 1: obj= "min"

The optimal bases are: (x_2, x_3), (x_1, x_3).

The feasible basis is: (x_1, x_2).

Case 2: obj= "max"

The optimal basis is: (x_1, x_2).

The feasible bases are: (x_2, x_3), (x_1, x_3).

Example 2: