第13卷第2期
1993年5月
数学研究与评论
JOURNAL OF MATHEMATICAL RESEARCH AND EXPOSITION
Vol. 13 No. 2
May 1993

单纯形方法的两个变种*

潘平奇　　　　欧阳梓祥
(南京林业大学数学系,210037)　(南京大学数学系,210008)

考虑标准形式的线性规划问题:

$$(\mathrm{LP})\qquad \text{maximize}\quad z=c^Tx$$
$$\text{s.t.}\quad Ax=b,\ x_j\geqslant 0,\ j\in E=\{1,\cdots,n\}.$$

其中 $A\in R^{m\times n}, b\in R^m, c\in R^n$,且 A 为行满秩.设有一基矩阵 $A_m=(a_{j_1},\cdots,a_{j_m})$,相应的基变量构成的向量为 $x_B=(x_{j_1},\cdots,x_{j_m})^T$.引入列指标集 $J_B=\{j_1,\cdots,j_m\}$ 及其补集 $\bar{J}_B=E-J_B$ 和行指标集

$$I=\{i\mid(A_m^{-1}b)_i<0,\quad i=1,\cdots,m\}\tag{1}$$

及其补集 $\bar{I}=\{1,\cdots,m\}-I$.我们有如下类型的子问题:

$$\text{maximize } z=\bar{z}_0+\sum_{j\in\bar{J}_B}\bar{c}_jx_j\tag{2a}$$

$$\text{s.t. } x_B=A_m^{-1}b-\sum_{j\in\bar{J}_B}(A_m^{-1}a_j)x_j;\tag{2b}$$

$$x_j\geqslant 0,\ \forall\ j\in\bar{J}_B;\tag{2c}$$

$$x_{j_i}\geqslant 0,\ \forall\ i\in\bar{I}.\tag{2d}$$

若 I 非空,取行指标 $\hat{i}$ 使得

$$\hat{i}=\operatorname{Argmin}(A_m^{-1}b)_i,\tag{3}$$

则第二种类型子问题可这样构成:目标函数为

$$x_{j_{\hat{i}}}=(A_m^{-1}b)_{\hat{i}}-\sum_{j\in\bar{J}_B}(A_m^{-1}a_j)_{\hat{i}}x_j.\tag{4}$$

即(2)的第 $\hat{i}$ 个等式,而约束与第一型子问题的约束相同.不过对第二型子问题,其"最优解"并不一定使 $x_{j_{\hat{i}}}$ 取到最大值,只要取非负值即可.显然,典则形式(2)对这两类子问题都是可行的.第一型问题或者有最优解,或者无上界.第二型子问题或者有或者无最优解,无最优解的情形表明了(LP)无最优解.

算法1　设 A_m^{-1} 为初始可行基的逆矩阵,I 由(1)式确定.

1. 若 I 是空集转向 6;　2. 由(3)式确定行指标 $\hat{i}$;

3. 解第二型子问题;　4. 若该子问题无解,停止;

5. 重新定义 I 并转向 1;　6. 解第一型子问题.

算法2　初始状态与算法1相同.

1. 解第一型子问题;　2. 若 I 是空集,停止;

* 1991年10月26日收到.国家自然科学基金资助项目.

3. 由(3)式确定行指标 $\hat{i}$； 4. 解第二型子问题；

5. 若该子问题无解，停止； 6. 重新定义 I 并转向 1.

定理 在非退化的假设下，算法 1 或算法 2 可产生(LP)的一个最优解，或判定(LP)无可行解，或判定(LP)为无界问题(证明略).

上述算法中求解第一和第二型子问题的方法可使用通常的单纯形法. 算法 1 是首先求提(LP)的可行解，再改进可行解最终达到最优解，其思路与通常的二阶段单纯形法类似. 算法 2 则不同，它每次在一个约束子集上求最优解，然后逐渐扩大约束子集，最终求(LP)的最优解；虽然产生的点列也导致最优顶点，但这些点一般不再是多胞形的顶点，甚至不是可行点. 此外，也无需引进人工变量. 尤其值得注意的是，初始基有选择余地，有可能选择一个基使之接近最优基从而减少迭代次数，而这可通过把进基的优先数给予那些有较大“选主元指标”的变量做到.(关于选主元指标，请参[1]和[2])

参 考 文 献

[1] Ping－Qi Pan, Practical Finite Pivoting Ruler for the Simplex Method, OR Spectrum 12(1990), 219－225.

[2] Ping－Qi Pan, A Simplex－line Method with Bisection tor Linear Programming, Qptimization (1991), 22.