

行政院國家科學委員會補助專題研究計畫成果報告

混合型線性規劃的計算方法

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC - 89 - 2213 - E - 004 - 004

執行期間：88年08月01日至89年07月31日

計畫主持人：陸行

共同主持人：蔡瑞煌

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

執行單位：政治大學應用數學系

中 華 民 國 89 年 07 月 31 日

行政院國家科學委員會專題研究計畫成果報告

混合型線性規劃的計算方法

計畫編號：NSC 89-2213-E-004-004

執行期限：88 年 08 月 01 日至 89 年 07 月 31 日

主持人：陸行 政治大學應用數學系

共同主持人：蔡瑞煌 政治大學資訊管理學系

1. Abstract

In this paper, we present an auxiliary algorithm that, in terms of the speed of obtaining the optimal solution, is effective in helping the simplex method for commencing a better initial basic feasible solution. The idea of choosing a direction towards an optimal point presented in this paper is new and easily implemented. From our experiments, the algorithm will release a corner point of the feasible region within few iterative steps, independently of the starting point. The computational results show that after the auxiliary algorithm is adopted as phase I process, the simplex method consistently reduce the number of required iterations by about 40%.

Keywords: Linear Programming, Interior-point-based Solution.

中文摘要

在本篇報告中，我們以求解速度的角度呈現一輔助的運算法則，幫助單體法從較佳的初始可行解開始求解。從實驗當中，選擇一新且易於執行的方向，且和起始點無關，免除在遞迴步驟計算可行解區域的角點的目標函數值。計算結果顯示，採納在階段 I 使用輔助運算法則，在階段 II 使用單體法，則降低所需遞迴步驟約百分之四十。

關鍵詞：線性規劃、內點解法。

2. The Proposed Approach

Consider the following standard form of LP problem (1). In (1), $\mathbf{x}^t = (x_1 \ x_2 \ \dots \ x_n)$, $\tilde{\mathbf{A}}$ is an $m \times n$ matrix, $\tilde{\mathbf{b}}^t = (\tilde{b}_1 \ \tilde{b}_2 \ \dots \ \tilde{b}_m)$, $\tilde{\mathbf{c}}^t = (\tilde{c}_1 \ \tilde{c}_2 \ \dots \ \tilde{c}_n)$, and $\mathbf{0}$ is denoted a column vector, of appropriate dimension, with each component equals zero.

$$\begin{aligned} \text{Minimize } z &= \tilde{\mathbf{c}}^t \mathbf{x} \\ \text{Subject to } \tilde{\mathbf{A}} \mathbf{x} &\geq \tilde{\mathbf{b}} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned} \quad (1)$$

Let LP problem (2) be a normalization version of LP (1). That is, $\mathbf{c} = \frac{\tilde{\mathbf{c}}}{\|\tilde{\mathbf{c}}\|}$, $\mathbf{A}_{i \cdot}$

$\frac{\tilde{\mathbf{A}}_{i \cdot}}{\|\tilde{\mathbf{A}}_{i \cdot}\|}$, and $b_i = \frac{\tilde{b}_i}{\|\tilde{\mathbf{A}}_{i \cdot}\|}$, where $\tilde{\mathbf{A}}_{i \cdot}$ is the i th row vector of $\tilde{\mathbf{A}}$ and the Euclidean norm is adopted for $\|\cdot\|$. It is easy to check the sets of feasible solutions as well as the optimal solutions of both LP (1) and LP (2) are equivalent.

$$\begin{aligned} \text{Minimize } z &= \mathbf{c}^t \mathbf{x} \\ \text{Subject to } \mathbf{A} \mathbf{x} &\geq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned} \quad (2)$$

Below, we take LP (2) as our standard form of input to illustrate the idea of the improving feasible direction in details.

$$\mathbf{h}^{(k)} \equiv \frac{\sum_{j \in \Omega_1^k} \mathbf{A}_{j \cdot}^t + \sum_{j \in \Omega_2^k} \mathbf{e}_j}{\left\| \sum_{j \in \Omega_1^k} \mathbf{A}_{j \cdot}^t + \sum_{j \in \Omega_2^k} \mathbf{e}_j \right\|} \quad (3)$$

$$\mathbf{g}^{(k)} = \begin{cases} \mathbf{0} & \text{if } \mathbf{h}^{(k)} = \mathbf{c} \\ \frac{\mathbf{h}^{(k)} - \mathbf{c}}{\|\mathbf{h}^{(k)} - \mathbf{c}\|} & \text{if } \mathbf{h}^{(k)} \neq \mathbf{c} \end{cases} \quad (4)$$

$$\begin{aligned} & \left. \begin{aligned} & \min \left\{ \begin{aligned} & -\frac{x_j^{(k)}}{g_j^{(k)}}, \quad g_j^{(k)} < 0, \quad 1 \leq j \leq n; \\ & \frac{b_i - \mathbf{A}_{i\bullet} \mathbf{x}^{(k)}}{\mathbf{A}_{i\bullet} \mathbf{g}^{(k)}}, \quad \mathbf{A}_{i\bullet} \mathbf{g}^{(k)} < 0, \quad 1 \leq i \leq m \end{aligned} \right\} \end{aligned} \right\} \quad (5) \end{aligned}$$

Table 1 displays the proposed algorithm. From Lemma 2, the criterion of $\mathbf{h}^{(k)} = \mathbf{c}$ in Step 3 is a sufficient stopping criterion of the proposed algorithm. Step 5 guarantees $\mathbf{x}^{(k+1)}$

Table 1. The proposed algorithm. \mathbf{e}_j is an n-dimensional unit vector with 1 for its jth component and 0 for the rest, and $\mathbf{A}_{i\bullet}$ is the ith row of \mathbf{A} . $\varepsilon < 10^{-6}$, and L is a large number.

is always a boundary point of \mathbf{P} . Theorem 1 says that the criterion in Step 6 is also a sufficient stopping criterion of the proposed algorithm. As to find an initial feasible solution of LP(2) in Step 0, we can apply the same algorithm to LP (2) with artificial variables.

<p>Step 0. Start with a feasible solution of LP (2), $\mathbf{x}^{(0)}$. Let $k = 0$.</p> <p>Step 1. Let $\Omega_1^k = \{i \mid \mathbf{A}_{i\bullet} \mathbf{x}^{(k)} = b_i\}$ and $\Omega_2^k = \{j \mid x_j^{(k)} = 0\}$.</p> <p>Step 2. Calculate $\mathbf{h}^{(k)} = \frac{\sum_{i \in \Omega_1^k} \mathbf{A}_{i\bullet} + \sum_{j \in \Omega_2^k} \mathbf{e}_j}{\left\ \sum_{i \in \Omega_1^k} \mathbf{A}_{i\bullet} + \sum_{j \in \Omega_2^k} \mathbf{e}_j \right\ }$.</p> <p>Step 3. If $\mathbf{h}^{(k)} = \mathbf{c}$, then STOP and claim $\mathbf{x}^{(k)}$ is an optimal solution.</p> <p>Step 4. Calculate $\mathbf{g}^{(k)} = \frac{\mathbf{h}^{(k)} - \mathbf{c}}{\ \mathbf{h}^{(k)} - \mathbf{c}\ }$.</p> <p>Step 5. If $\forall i \in \Omega_1^k \quad \mathbf{A}_{i\bullet} \mathbf{g}^{(k)} \geq 0$ and $\forall j \in \Omega_2^k \quad g_j^{(k)} \geq 0$, then go to Step 10.</p> <p>Step 6. Calculate $\mathbf{d} = -[\mathbf{I} - \mathbf{A}_{i\bullet} \quad -\mathbf{A}_{i\bullet}^t \mathbf{A}_{q\bullet}] \mathbf{c}$, $i \in \Omega_1^k$ & $\mathbf{A}_{i\bullet} \mathbf{g}^{(k)} < 0$ Or $\mathbf{d} = -\mathbf{c} + c' \mathbf{e}_j$, $j \in \Omega_2^k$ & $g_j^{(k)} < 0$</p> <p>Step 7. Let $\eta^{(k)} = \min \left\{ \begin{aligned} & \frac{-x_j^{(k)}}{d_j}, \quad d_j < 0, \quad 1 \leq j \leq n, \quad j \notin \Omega_2^k; \\ & \frac{b_i - \mathbf{A}_{i\bullet} \mathbf{x}^{(k)}}{\mathbf{A}_{i\bullet} \mathbf{d}}, \quad \mathbf{A}_{i\bullet} \mathbf{d} < 0, \quad 1 \leq i \leq m, \quad i \notin \Omega_1^k \end{aligned} \right\}$</p> <p>Step 8. If $\eta^{(k)} = 0$, then STOP and claim $\mathbf{x}^{(k)}$ is an optimal solution.</p> <p>Step 9. Let $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \eta^{(k)} \mathbf{d}$, $k+1 \rightarrow k$. Go to Step 13.</p> <p>Step 10. Let $\eta^{(k)} = \min \left\{ \begin{aligned} & \frac{-x_j^{(k)}}{g_j^{(k)}}, \quad g_j^{(k)} < 0, \quad 1 \leq j \leq n, \quad j \notin \Omega_2^k; \\ & \frac{b_i - \mathbf{A}_{i\bullet} \mathbf{x}^{(k)}}{\mathbf{A}_{i\bullet} \mathbf{g}^{(k)}}, \quad \mathbf{A}_{i\bullet} \mathbf{g}^{(k)} < 0, \quad 1 \leq i \leq m, \quad i \notin \Omega_1^k \end{aligned} \right\}$</p> <p>Step 11. If $\eta^{(k)} = 0$, then STOP and claim $\mathbf{x}^{(k)}$ is an optimal solution.</p> <p>Step 12. Let $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \eta^{(k)} \mathbf{g}^{(k)}$, $k+1 \rightarrow k$.</p> <p>Step 13. Let $\Omega_1^k = \{i \mid \mathbf{A}_{i\bullet} \mathbf{x}^{(k)} = b_i\}$ and $\Omega_2^k = \{j \mid x_j^{(k)} = 0\}$.</p> <p>Step 14. If $\ \Omega_1^k + \Omega_2^k\ = n$, then STOP and claim $\mathbf{x}^{(k)}$ is a corner point feasible solution.</p> <p>Step 15. If $\varepsilon < \eta^{(k-1)} < L$, then go to Step 2.</p> <p>Step 16. Let \mathbf{A}_q be defined as composed of the row vectors of $\mathbf{A}_{i\bullet}$, $\forall i \in \Omega_1^k$ and $\mathbf{e}_j' \quad \forall j \in \Omega_2^k$.</p> <p>Step 17. Calculate $\mathbf{d} = -[\mathbf{I} - \mathbf{A}_q^t (\mathbf{A}_q \mathbf{A}_q^t)^{-1} \mathbf{A}_q] \mathbf{c}$.</p> <p>Step 18. Let $\eta =$</p>

$$\min \left\{ \frac{-x_j^{(k)}}{d_j}, d_j < 0, 1 \leq j \leq n, j \notin \Omega_2^k; \frac{b_i - \mathbf{A}_{i \cdot} \mathbf{x}^{(k)}}{\mathbf{A}_{i \cdot} \mathbf{d}}, \mathbf{A}_{i \cdot} \mathbf{d} < 0, 1 \leq i \leq m, i \notin \Omega_1^k \right\}$$

Step 19. Let $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \eta \mathbf{d}$, $k+1 \rightarrow k$.

Step 20. Let $\Omega_1^k = \{i \mid \mathbf{A}_{i \cdot} \mathbf{x}^{(k)} = b_i\}$ and $\Omega_2^k = \{j \mid x_j^{(k)} = 0\}$.

Step 21. If $\|\Omega_1^k + \Omega_2^k\| = n$, then STOP and claim $\mathbf{x}^{(k)}$ is a corner point feasible solution; else go to Step 16.

In order to avoid the zig-zag situation during the process, the loop of Steps 16-21 is set for determining a basic feasible solution to start the simplex method. Generally, Steps 16-21 implement the gradient projection method to obtain a corner point of feasible solution. The matrix $\mathbf{M} \equiv \mathbf{I} - \mathbf{A}_q^t (\mathbf{A}_q \mathbf{A}_q^t)^{-1} \mathbf{A}_q$ is the project matrix corresponding to the boundary. Action by it on any vector yields the projection of that vector onto the boundary. Since $\mathbf{d} = -\mathbf{M}\mathbf{c}$ in Step 17, \mathbf{d} is a feasible direction of descent on the boundary if $\mathbf{d} \neq \mathbf{0}$. That is, we have $\mathbf{c}^t \mathbf{x}^{(k+1)} < \mathbf{c}^t \mathbf{x}^{(k)}$, if $\mathbf{x}^{(k+1)} (= \mathbf{x}^{(k)} + \eta \mathbf{d})$ does not equal $\mathbf{x}^{(k)}$. Step 18 guarantees $\mathbf{x}^{(k+1)}$ is always a boundary point. Before obtaining a basic feasible solution, it is easy to check there are at most n iterations in the loop of Steps 16-21.

3. Computational Results

Before we present our computational results, some remarks about the test

Table 2. The summary of the average numbers of total iterations required for the simplex method. The numbers shown in the parenthesis is the associated standard deviation.

m	n	Auxiliary + Simplex	Simplex
10	10	3.04 (1.01)	6 (2.89)
20	20	11.41 (4.62)	17.57 (5.38)
30	30	19.28 (7.43)	31.63 (4.50)

Here we propose an auxiliary algorithm for the simplex method. The auxiliary algorithm has the following advantages: there is no need to know the value of an optimal solution, and no special format for the problem is needed. It is a practical algorithm because its computation complexity is $O(m+n)$. Furthermore, the

problems are in order. All test problems essentially were randomly generated, but were first screened by a preprocessor to remove any possible inconsistent problems. The problems were then scaled by normalizing each row of \mathbf{A} . For each test problem, there are ten different initial points, which are picked up randomly, for the auxiliary algorithm. According to the problem size in terms of m and n of matrix \mathbf{A} , we report the required iterations versus that required by the simplex method.

Table 2 shows the summary of the computational results. It shows that the auxiliary algorithm help the simplex method consistently reducing the number of required iterations by about 40%.

does help the simplex method consistently reducing the number of required iterations by about 40%. For future studies, we shall test it with more larger-scale problems as well as take the same test problems used in NETLIB for further comparisons.

4. Conclusion and Future Work

Here we propose an auxiliary algorithm for the simplex method. The auxiliary algorithm has the following advantages: there is no need to know the value of an optimal solution, and no special format for the problem is needed. It is a practical algorithm because its computation complexity is $O(m+n)$. Furthermore, the simplex method with this help can possess a better worst-case complexity bound than those without this help. Although there lacks a proof for the better global complexity for our method, we have explored its computational performance on some random test problems. The computational results show that the auxiliary algorithm does help the simplex method consistently reducing the number of required iterations by about 40%. For future studies, we shall test it with more larger-scale problems as well as take the same test problems used in NETLIB for further comparisons.

5. Reference

- [1] I. Adler, N. Karmarkar, M.G.C. Resende and G. Veiga, An implementation of Karmarkar's algorithm for linear programming. *Mathematical Programming* 1989; 44: 297-335.
- [2] E.D. Anderson, and Yinyu Ye, Combining interior-point and pivoting algorithms for linear programming. *Management Science* 1996; 42, (12): 1719-1731.
- [3] K.M. Anstreicher, and P. Watteyne, A family of search directions for Karmarkar's algorithm. *Operations Research* 1993, 41, (4): 759-767.
- [4] R.E. Bixby, J.W. Gregory, I.J. Lustig, R.E. Marsten and D.F. Shanno, Very large-scale linear programming: A case study in combining interior point and simplex methods. *Operations Research* 1992, 40, (5): 885-897..
- [5] S.C. Fang, and S. Puthenpura. *Linear Optimization and Extensions*, Prentice-Hall, Englewood Cliffs New Jersey, USA, 1993.
- [6] N. Karmarkar, A new polynomial-time algorithm for linear programming. *Combinatorica* 1984; 4: 373-395.
- [7] C.L. Monma, and A.J. Morton, Computational experience with a dual affine variant of Karmarkar's method for linear programming. *Operations Research Letters* 1987; 6: 261-267.
- [8] M.J. Todd, A Dantzig-Wolfe-Like Variant of Karmarkar's interior-point linear programming algorithm. *Operations Research* 1990; 38, (6): 1006-1018.