アルゴリズム 1－6
（1988. 5. 23）

勢力圏上の最適化によるＬＰの一解法

鈴木　誠道
上智大学理工学部機械工学科

本報告では，目的関数の既知な線形計画法の問題を勢力圏上の最適化問題に変換して解く２つのアルゴリズムを提案している．線形計画法の許容領域は，有限個の半空間の積集合になっている．この半空間群に，目的関数値が，最適値以上（最小化問題の場合）なる半空間を付加し，この目的関数の半空間の勢力圏を考える．これは凸閉集合となる．この勢力圏上で目的関数を最大化することにより，もとの線形計画の最適解が求められる．本研究ではこの最適化にニュートン法に類似の方法を用いる．実際には，勢力圏の境界面上の点から出発して，目的関数の最適値のレベルまで進む．その後，再び勢力圏の境界面にもどる．これを繰り返すと，生成される点は最適解に収束する．本報告には簡単な数値例も示している．

A METHOD FOR A LINEAR PROGRAMMING PROBLEM BY MAXIMIZING THE OBJECTIVE FUNCTION ON A REGION OF INFLUENCE

Shigemichi SUZUKI

Department of Mechanical Engineering, Sophia University
Kioi-cho 7-1, Chiyoda-ku, Tokyo 102, Japan

We propose two algorithms for a linear programming problem with a known optimum value of the objective function by converting it to an optimization problem in a region of influence. The feasible region of the linear program is the intersection of the halfspaces of constraint inequalities. After incorporating one more halfspace, i.e., the halhspace consisting of the points for which the values of the objective function are greater than or equal to the optimum value. We conside the region of influence of this added halfspace. This region is a closed and convex set. An optimum solution of the original linear program can be found by maximizing the objective function in the region of influence. We employ a Newton-like method for maximization. The sequence of points generated by this method can be proved to converge to an optimum solution . The complexity of the proposed algorithms is of polynomial order in m and n but not in the total number of input bits. Detailed accounts of the algorithms are presented together with some illustrative numerical examples.

1. Introduction

Since the publication of Karmarkar's algorithm [1] for linear programming, a lot of extentions and variations of his algorithm have been attempted and some new ones have been proposed [2],[3],[4],[5],and [6]. Most of these works employ nonlinear approaches in a sense or another and devise iterative procedures to lead to an optimum solution of linear programming. Introduction of nonlinearity seems unnatural but gives a nice property of computing complexity of polynomial order. The present author is trying to establish a polynomial order algorithm for linear programming using only linear computation. The work is still on the way. In this note we will show a method for linear programming problems by converting them to optimization problems in a region of influence. Simple illustrative examples are also given.

2. Problem

Consider a linear programming problem:

$$\text{Minimize} \quad z = c^t x$$

$$\text{subject to} \quad Ax \geqq b, \quad x \geqq 0 \quad , \qquad (1)$$

where A is an m x m matrix, b is an m vector, c and x are n vectors. After incorporating the nonnegativity constraint $x \geqq 0$ into the constraint matrix, we will denote the augmented matrix as A again. Then the problem we will consider in the following will become:

$$\text{Minimize} \quad z = c^t x \quad \text{subject to} \quad A x \geqq b \; , \qquad (2)$$

where A is (m+n) x n, b is an (m+n) vector, c and x are n vectors.

We will make the following assumptions:

(A1) The set of optimum solutions S^* of linear program (2) is not empty.

(A2) An optimum value z^* of linear program (2) is known.

(A3) $\| a_i \| < \| c \|$ (i = 1, 2, ... , m+n), where $\|a_i\|$ and $\|c\|$ are the Euclidean norms of the i-th row of A and c respectively.

3. Region of Influence and Its Properties

After introducing a constraint $c^t x \geqq z^*$ the distance between a point $x \in R^n$ and the set of half-spaces $H = \{ c^t x \geqq z^*, a_i^t x \geqq b_i$ (i = 1, 2, ..., m+n) $\}$ will be defined as

$$d_H = \min [\min_i a_i^t x - b_i , c^t x - z^*]. \qquad (3)$$

The region of influence of each half-space of H is the set of points in R^n for which the half-space is the closest among H in the sense of $d_H(x)$. Specifically, the region of influence V_i of the half-space $a_i^t x \geqq b_i$ and V_0 of $c^t x \geqq z^*$ will be defined as

$$V_i = \{ x \mid x \in R^n, a_i^t x - b_i \leqq a_j^t x - b_j \ (j \neq i), \; a_i^t x - b_i \leqq c^t x - z^* \} \qquad (4)$$

$$V_0 = \{ x \mid x \in R^n, c^t x - z^* \leqq a_i^t x - b_i , \; i = 1, 2, ..., m+n \} \; . \qquad (5)$$

An example of regions of influence in R^2 is shown in Fig.1. Some of the properties utilized in the sequel will be stated:

<u>Proposition 1</u> V_i (i = 0, 1, 2, ..., m + n) is a closed, convex set in R^n.

<u>Proof</u> Obvious. □

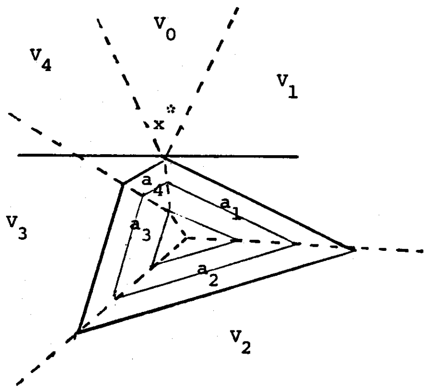

Fig. 1 An Example of Regions of Influence

Proposition 2 Let x^0 be an arbitrary point in R^n. Then

$$\{ x \mid x \in R^n, x = x^0 - \beta c , \beta > 0 \} \quad V_0 \neq \phi . \qquad (6)$$

Proof It is easy to verify the inequality $c^t(x^0 - \beta c) - z^* \leq a_i^t(x^0 - \beta c) - b_i$ for sufficiently large β. □

We will note that the assertion is true without assumptions (A1) and (A2).

Proposition 3

$$\text{Max} \{ c^t x : x \in V_0 \} = z^* . \qquad (7)$$

Proof Let $\hat{x} \in S^*$ which is nonempty by assumption (A1). First note that $\hat{x} \in V_0$. Next we will show $x \in V_0$ implies $c^t x \leq z^*$. To prove this, suppose $c^t x > z^*$. Then $0 < c^t x - z^* \leq a_i^t x - b_i$ ($i = 1, 2, \cdots, m+n$). This implies that x is an interior point of the fesible region of linear program (2). Consider a ray emanating from x in the direction - c. Let x^b and x^c be the intersecting points of the ray with the boundary of the feasible region and the plane $c^t x = z^*$. Let $a_{i_0}^t x^b = b_{i_0}$ for some i_0, then

$$\begin{aligned} \| x^c - x \| &= (c^t x - z^*)/ \|c\| \\ &\geq \| x^b - x \| \\ &\geq (a_{i_0}^t x - b_{i_0})/ \| a_{i_0} \| . \end{aligned} \qquad (8)$$

Since $\| c \| > \| a_{i_0} \|$ from assumption (A3) we will have

$$c^t x - z^* > a_{i_0}^t x - b_{i_0} . \qquad (9)$$

This is a contradiction since $x \in V_0$. This establishes that $x \in V_0$ implies $c^t x \leq z^*$. Hence z^* is an upper bound of $c^t x$ in V_0. Since $\hat{x} \in V_0$ attains this upper bound assertion (7) is proved. □

4. A Method for LP by Optimization in Region of Influence

According to Proposition 3 an optimum solution of linear program (2) can be found by maximizing $c^t x$ in V_0.

We will employ a Newton-like method for maximization. Suppose a point $\tilde{x}^k$ on the boundary of V_0 is found in the k-th iterative cycle. The vector c is projected to the boundary hyperplane at $\tilde{x}^k$. We proceed from $\tilde{x}^k$ in the direction of the projected gradient to the point x^{k+1} where the ray meets the hyperplane $c^t x = z^*$. x^{k+1} does not belong to V_0 in general. By taking appropriate step size in the direction $-c$ from x^k, $\tilde{x}^{k+1} = x^k - \beta c$ will be on the boundary of V_0 again. If x^{k+1} does belong to V_0 then it is a feasible solution of linear program (2). Since $c^t x^{k+1} = z^*$ it is an optimum solution. Otherwise we continue the same procedure as we did at $\tilde{x}^k$. We call the method just described informally as Algorithm I. Algorithm I is illustratively shown in Fig. 2.

After a finite number of iterations we can identify an active constraint at an optimum solution. We can eliminate a variable from the linear program using the active constraint. We will then solve the derived linear program with one variable less than the original one. Eventually we will come to either the situation in which a linear program with one variable has been solved or one in which x^k is feasible and thus an optimum solution is obtained for a reduced linear program. In either case an optimum solution for linear program (2) can be easily obtained by back substitution.

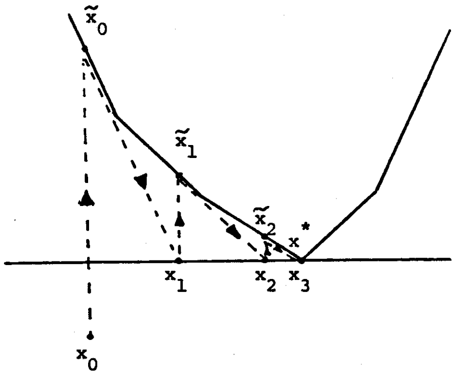

Fig. 2 Illustrative Explanation of Algorithm I

We call the second method as Algorithm II.

Befor we describe the algorithms we will introduce some notations. Let

$$\tilde{a}_i = a_i - c \qquad\qquad (10)$$
$$\tilde{b}_i = b_i - z^*$$
$$\beta_k = \max_i (\tilde{a}_i^{\,t} x^k - \tilde{b}_i) / \tilde{a}_i^{\,t} c$$
$$= (\tilde{a}^{k\,t} x^k - \tilde{b}^k) / \tilde{a}^{k\,t} c$$
$$\mu_k = a^{k\,t} x^k - b^k$$
$$\gamma_k = \tilde{a}^{kt} c \qquad\qquad (11)$$
$$\alpha_k = \|c\|^2 \beta_k / [\|c\|^2 - (\gamma_k / \|\tilde{a}^k\|)^2]$$
$$P_k = I - \tilde{a}^k \tilde{a}^{k\,t} / \| \tilde{a}^k \|^2$$
$$(k = 0, 1, 2, \cdots)$$

Algorithm I

step 0 (Initialization):

Choose x^0 arbitrarily and let $k = 0$.

Step 1:

If $\beta_k < \varepsilon$ then

stop

else

$$\tilde{x}^k = x^k - \beta_k c. \quad (12)$$

Step 2:

$$x^{k+1} = \tilde{x}^k + \alpha_k P_k c \quad (13)$$

and after incrementing k by 1 go back to step 1.

Algorithm II

Step 0 (initialization):

Choose x^0 arbitrarily and let $k = 0$ and $N = n$ (the number of variables).

Step 1:

If $\beta_k < \varepsilon$ then

if $N < n$ then

determine all other variables which are not included in the last reduced LP problem using the a set of equations obtained in (15) in the past iterations and stop

else stop

else go to next step.

Step 2:

If there exists only one index i_0 such that

$$\tilde{a}_{i_0}^t x^k - b_{i_0} < 0 \quad (14)$$

then using the equation

$$\tilde{a}_{i_0}^t x - b_{i_0} = 0 \quad (15)$$

eliminate one of the variables from other constraints and after decrementing N by 1 and then letting x^0 be the restriction of x^k to R^N go to step 1

else go to next step.

Step 3:

$$\tilde{x}^k = x^k - \beta_k c . \quad (16)$$

Step 4:

$$x^{k+1} = \tilde{x}^k + \alpha_k P_k c \quad (17)$$

and after incrementing k by 1 go back to step 1.

5. Validity of Algorithms

Proposition 4 If x^k is not a feasible point of linear program (2) a step-size $\beta_k > 0$ can be determined by the first equation of (11).

Proof If x^k is not feasible there exists at least one constraint such that

$$\tilde{a}_i^t x^k - \tilde{b}_i < 0 .$$

In addition $\tilde{a}_i^t c < 0$ since $\|a_i\| < \|c\|$ by assumption (A3). Therefore $\beta_k > 0$. □

Proposition 5 $\{ x^k \}$ generated by Algorithm I is convergent. Namely

$$x^k \to \hat{x} \quad (k \to \infty) \quad (18)$$

for some $\hat{x} \in S^*$.

Proof From (11)

$$x^{k+1} = x^k - \mu_k (\|c\|^2 \tilde{a}^k - \gamma_k c) / [(\|c\| \|\tilde{a}^k\|)^2 - \gamma_k^2] \quad (19)$$

Let x^* be a point of S^*, then after rather straightforward computation we have

$$\| x^{k+1} - x^* \|^2 = \| x^k - x^* \|^2 + [\|c\|^2 \mu_k(- \mu_k + 2(\tilde{a}^k x^* - \tilde{b}^k))] / [(\|c\| \|\tilde{a}^k\|)^2 - \gamma_k^2]. \quad (20)$$

If $\mu_k = 0$ then $a_i^t x^k - b_i \geqq 0$ for every i and x^k is an optimum solution of linear program (2). Hence the proposition is proved in this case. If $\mu_k \neq 0$ then $\mu_k = \tilde{a}_k^k x^k - \tilde{b}^k < 0$. Noting that $\tilde{a}^k x^* - \tilde{b}^k \geqq 0$ we have

$$\|x^{k+1} - x^*\|^2 < \|x^k - x^*\|^2 - (\|c\| \mu_k)^2/[(\|c\| \|\tilde{a}^k\|)^2 - \gamma_k^2]. \quad (21)$$

From (21) $\{ \|x^{k+1} - x^*\| \}$ is monotone decreasing and converges to $\delta \geqq 0$. There also holds $\mu_k \to 0$ ($k \to \infty$). Since $\{ x^k \}$ is bounded and $\mu_k \to 0$ as $k \to \infty$ there exists a unique accumulation point $\hat{x}$ of $\{ x^k \}$. $\mu_k \to 0$ further implies

$$\tilde{a}^{k\,t} \hat{x} - \tilde{b}^k = 0 \quad (k \to \infty) \quad (22)$$

and hence

$$a_i^t \hat{x} - b_i \geqq 0 \quad (23)$$

for every i together with $c^t \hat{x} = z^*$ establishes that the proposition is true in this case, too. □

<u>Proposition 6</u> If there is only one i_0 such that

$$\tilde{a}_{i_0}^t x^k - \tilde{b}_{i_0} < 0 \quad (24)$$

in step 1 of Algorithm I. Then

$$\tilde{a}_{i_0}^t \hat{x} - \tilde{b}_{i_0} = 0 \quad (25)$$

for some $\hat{x} \in S^*$.

<u>Proof</u> Let x^{**} be an optimum solution which has the minimum distance from the space $A_{i_0} = \{ x \mid \tilde{a}_{i_0}^t x - \tilde{b}_{i_0} = 0 \}$. If x^{**} is in this space we are through. So assume that x^{**} is not in this space. Consider a line segment $\overline{x^k x^{**}}$ and its projection on $c^t x = z^*$ which is $\overline{x^k x^{**}}$. Since x^{**} is the closest to A_{i_0} there exists a constraint such that

$$\tilde{a}_{i_1}^t x^{**} \tilde{b}_{i_1} = 0 \quad (26)$$

and

$$\tilde{a}_{i_1}^t (\lambda x^k + (1 - \lambda) x^{**}) - \tilde{b}_{i_1} < 0 \quad (27)$$

for $1 \geqq \lambda \gtrsim 0$. This implies

$$\tilde{a}_{i_1}^t x^k - b_{i_1} < 0 . \quad (28)$$

Since $i_0 \neq i_1$ this contradicts (24). Therefore equation (25) holds for $\hat{x} = x^{**}$. □

From Proposition 6 we do not have to continue the iteration of Algorithm I when relation (22) holds for only one index i_0. This leads to the construction of Algorithm II which takes into account this situation. It is noted Algorithm II will terminate in a finite number of iterations even if $\varepsilon = 0$.

6. Illustrative Examples

<u>Example 1</u>

$$\begin{array}{llrcrcl} \text{Maximize} & z = & -3\,x_1 & - & 3\,x_2 & & \\ \text{subject to} & & -\,x_1 & + & x_2 & \geqq & -1 \\ & & x_1 & - & x_2 & \geqq & -1 \\ & & x_1 & - & 2\,x_2 & \geqq & -5/2 \\ & & -2\,x_1 & - & x_2 & \geqq & -5 \\ & & -2\,x_1 & - & 3\,x_2 & \geqq & -10 \\ & & x_1 & , & x_2 & \geqq & 0 \;. \end{array} \qquad (29)$$

Assume here that $z^* = -10.5$ is known.

Choose $x^0 = (2, 0)^t$ then the results of computation are as follows.

Algorithm I:

k	$x^{k\ t}$	$\tilde{x}^{k\ t}$
0	(2, 0)	(19, 7)/6
1	(1.5, 2)	------ optimum solution.

<u>Example 2</u> [7]

$$\begin{array}{llrcrcrcl} \text{Minimize} & z = & -3\,x_1 & -2\,x_2 & -4\,x_3 & & \\ \text{subject to} & & -\,x_1 & -\,x_2 & -2\,x_3 & \geqq & -4 \\ & & -2\,x_1 & & -2\,x_3 & \geqq & -5 \\ & & -2\,x_1 & -\,x_2 & -3\,x_3 & \geqq & -7 \\ & & x_1\,, & x_2\,, & x_3 & \geqq & 0\;. \end{array} \qquad (30)$$

Assume also that $z^* = -21/2$ is known.

Choose $x^0 = (0, 0, 0)^t$ then the results of computation are as follows:

Algorithm I:

k	$x^{k\ t}$	$\tilde{x}^{k\ t}$	$\Vert x^k - x^* \Vert^2$
0	(0, 0, 0)	(39, 26, 52)/32	8.5
1	(5/2, 3/10, 3/5)	(137, 23, 46)/50	1.8
2	(59, 37, 16)/30	(121, 76, 36)/60	0.64
3	(375, 161, 32)/150	(5817, 2543, 736)/2250	0.23
4	(3119, 1297, 256)/1350	(3123, 1913, 288)/1350	0.09
5	(16875, 9101, 512)/6750		0.03

Algorithm II:

k	$x^{k\ t}$	$\tilde{x}^{k\ t}$	$\Vert x^k - x^* \Vert^2$
0	(0, 0, 0)	(39, 26, 52)/32	8.5
1	(5/2, 3/10, 3/5)	(137, 23, 46)/50	1.8
2	(59, 37, 16)/30	(--, 29, 0)/30	0.64
3	(--, 3/2, 0)	---- optimum solution	0.00

At the start of Iteration 3 the constraint

$$-\,x_1 - x_2 - 2\,x_3 \geqq -4$$

is identified to be active at an optimum solution. Therefore from the optimum solution of the reduced linear program with two variables x_2 and x_3 we can obtain the value of x_1 at the optimum solution. Namely

$$x_1^* = 4 - x_2^* = 4 - 3/2 = 5/2.$$

The optimum solution x^* is therefore

$$x^* = (5/2, 3/2, 0)^t.$$

7. Infeasible and Unbounded Problems

So far we have assumed that a linear program has an optimum solution. There are, of course, linear programs with no feasible solutions or unbounded values of objective functions. So we will leave out assumptions (A1) and (A2) in this section in order to show how to handle these situations.

Consider a linear programming problem:

$$\text{Minimize} \quad z = c^t x$$
$$\text{subject to} \quad Ax \geqq b, \quad x \geqq 0. \qquad (31)$$

Insted of directly dealing with linear program (31) we will consider the following linear program:

$$\text{Minimize} \quad w = c^t x - b^t y$$
$$\text{subject to} \quad A x \geqq b, \; x \geqq 0, \quad -A^t y \geqq -c, \; y \geqq 0. \qquad (32)$$

Proposition 7

(i) Linear program (32) has either an optimum solution with $w^* = 0$ or no fesible solutions, and

(ii) under assumption (A3)

$$\text{Max} \{ c^t x - b^t y \mid (x^t, y^t)^t \in V_0 \text{ of } c^t x - b^t y \geqq 0 \}$$

$= 0$ if there exists an optimum solution of (32)

< 0 if there does not exist feasible solutions of (32).

Proposition 8 If there do not exist a bounded optimum solution for linear program (31), then under assumption (A3)

$$\text{Max} \{ c^t x \mid x \in V_0 \text{ of } c^t x \geqq M \text{ for linear program (31)} \}$$

$< M$ if linear program (31) is infeasible,

$> M$ if linear program (31) has an unbounded optimum solutions,

for sufficiently small M.

Propositions 7 and 8 suggest some obvious ways of identifying the existence of the optimum, the infeasibility, and the unboundedness of linear program (31).

8. Conclusion

We proposed an approach for a linear programming problem by maximizing $z = c^t x$ in a region of influence with a Newton-like method. The convergence is proved but the method is not of polynomial order.

Acknowledgment The author is grateful to Professor Takao Asano for helpful discussions and valuable comments.

REFERENCES

[1] N. Karmarkar: A New Polynomial-Time Algorithm for Linear Programming, Combinatorica, Vol. 4 (1984), 373--395.

[2] M. Iri and H. Imai: A Multiplicative Barrier Function Method for Linear Programming, Algorithmica, Vol. 1 (1986), 455--482.

[3] M. Kojima, S. Mizuno, and A. Yoshie: A Primal-Dual Interior Point Algorithm for Linear Programming, New Methods for Linear Programming (Joint Research Report No. 5, The Institute of Statistical Mathematics) (1987), 13--24. (in Japanese)

[4] K. Tone: An implementation of a Revised Karmarkar Method, New Methods for Linear Programming (Joint Research Report No. 5, The Istitute of Statistical Mathematics) (1987), 1--12.(in Japanese)

[5] K. Tanabe: Complementarity-Enforcing Centered Newton Method for Mathematical Programming : Global Method, New Methods for Linear Programming (Joint Research Report No. 5, The Institute of Statistical Mathematics) (1987), 118--144. (in Japanese)

[6] Special Issue of Algorithmica (New Approaches to Linear Programming), Vol. 1 (1986), 387--593. (Ten papers including [2].)

[7] H. Konno: Linear Programming, JUSE Publication, Tokyo (1987). (in Japanese)