Journal of the Operations Research
Society of Japan
Vol. 32, No. 3, September 1989

POLYNOMIAL TIME INTERIOR POINT ALGORITHMS FOR TRANSPORTATION PROBLEMS

Shinji Mizuno Kaori Masuzawa
Tokyo Institute of Technology

(Received July 6, 1988; Revised November 14, 1988)

Abstract This paper deals with the Hitchcock transportation problem with m supply points and n demand points. Assume that $m \leq n$ and all the data are positive integers which are less than or equal to an integer M. We propose two polynomial time algorithms for solving the problems. The algorithms are based on the interior point algorithms for solving general linear programming problems. Using some features of the transportation problems, we decrease the computational complexities. We show that one of the algorithms requires at most $O(m^3n^2 \log nM + n^3)$ arithmetic operations and the other requires at most $O(n^4 \log nM)$ arithmetic operations.

1. Introduction

This paper deals with the Hitchcock transportation problem with m supply points and n demand points. The problem is formulated as a linear programming problem in the following way:

$$\begin{array}{lllll} (P) & \min & \sum_{(i,j)\in N} c_{ij}x_{ij}, & \\ & \text{subject to} & \sum_{j\in J} x_{ij} = a_i & \text{for } i \in I, \\ & & \sum_{i\in I} x_{ij} = b_j & \text{for } j \in J, \\ & & x_{ij} \geq 0 & \text{for } (i,j) \in N, \end{array}$$

where

$$I = \{1, 2, \ldots, m\}, \ J = \{1, 2, \ldots, n\} \ \text{ and } \ N = I \times J.$$

We assume that $m \leq n$, $\sum_{i\in I} a_i = \sum_{j\in J} b_j$ and all the $c_{ij}\,((i,j) \in N)$, $a_i(i \in I)$ and $b_j\,(j \in J)$ are positive integers.

There are several algorithms for solving the transportation problem (see, for example, [1] or [10]). Edmonds and Karp [2] and Ikura and Nemhauser [4] especially propose polynomial time algorithms. The methods are based on the simplex algorithm. On the other hand, for a general linear programming problem, there are many polynomial time algorithms (Gonzaga [3], Khachiyan [6], Karmarkar [5], Kojima, Mizuno and Yoshise [7, 8], Monteiro and Adler [9], Renegar [11], Vaidya [14] etc.). Each of the methods generates a sequence of interior feasible points. So they are called the interior point algorithms. Some of the interior point algorithms (Gonzaga [3], Kojima, Mizuno and Yoshise [8], Monteiro and Adler [9] and Vaidya [14]) attain the $O(\bar{n}^3L)$ computational complexity in terms of the number of arithmetic operations, where $\bar{n}$ is the number of variables and L is the input size of the linear programming problem. If we apply these algorithms to the transportation problem (P), the computational complexity is

$$O(m^3n^3L)$$

for

$$L = \lceil \sum_{(i,j)\in N} \log(c_{ij}+1) + \sum_{i\in I}\log(a_i+1) + \sum_{j\in J}\log(b_j+1) + 2mn \rceil,$$

where $\lceil \theta \rceil$ denotes the smallest integer which is not less than θ.

In this paper, we propose two interior point algorithms, Algorithm **A** and Algorithm **B**, for the transportation problems. The algorithms attain the computational complexities lower than $O(m^3n^3L)$. Algorithm **A** is based on the $O(\bar{n}^4L)$ algorithm proposed by Kojima, Mizuno and Yoshise [7]. If we directly apply the algorithm to the transportation problems, the complexity is $O(n^4m^4L)$. Using some features of the transportation problems, we decrease the computational complexity of Algorithm **A** to

$$O(m^3n^2L' + n^3),$$

where

$$\begin{array}{llcl} (1) & L' & = & \log nM, \\ (2) & M & = & \max\{a_i \ (i\in I),\ b_j \ (j\in J),\ c_{ij} \ ((i,j)\in N)\}. \end{array}$$

Note that the value of L' is much less than that of L. Algorithm **B** is based on the $O(\bar{n}^{3.5}L)$ algorithm proposed by Kojima, Mizuno and Yoshise [8] for linear complementarity problems. We show that Algorithm **B** requires at most

$$O(n^4L')$$

arithmetic operations.

The computational complexity of the algorithm [2] is $O(n^3\log M)$ and that of [4] is $O(n^5\log M)$. So the computational complexities of our algorithms are less than that of [4] but greater than that of [2] except for the case $m^3 < n$ in Algorithm **A**. The algorithms [2, 4] need to solve $O(L)$ subproblems which are obtained by scaling an original problem. On the other hand, our methods directly solve an original transportation problem and need not to generate subproblems. Tardos [13] proposes a strong polynomial algorithm, i.e., the computational complexity is a polynomial of m and n. Although our algorithms are not strong polynomial, it is possible to construct a strong polynomial algorithm by using our algorithms for solving the subproblems in Tardos' algorithm.

In Section 2, we outline Algorithms **A** and **B**. Then we describe Algorithms **A** and **B** in detail in Section 3 and Section 4, respectively. In Section 5, we obtain the computational complexities of Algorithms **A** and **B**. Section 6 gives the conclusions.

2. The outline of algorithms

Here we outline Algorithm **A** and Algorithm **B**. The dual problem of (P) is formulated as

$$\begin{array}{llll} (D) & \max & \sum_{i\in I} a_iu_i + \sum_{j\in J} b_jv_j, & \\ & \text{subject to} & u_i + v_j + z_{ij} = c_{ij} & \text{for } (i,j)\in N, \\ & & u_1 = 0,\ z_{ij}\ge 0 & \text{for } (i,j)\in N. \end{array}$$

Since one of the constraints of (P) is redundant, we impose the constraint $u_1 = 0$ on (D). We represent the problems (P) and (D) by the following matrix forms:

$$\begin{array}{llll} (P) & \min & c^Tx, & \\ & \text{subject to} & Ax = d, & x\ge 0. \\ (D) & \max & d^Ty, & \\ & \text{subject to} & A^Ty + z = c, & y_1 = 0,\quad z\ge 0. \end{array}$$

where

$$x^T = (x_{11}, x_{12}, \dots, x_{mn}), \quad y^T = (u_1, u_2, \dots, u_m, v_1, \dots, v_n), \quad z^T = (z_{11}, z_{12}, \dots, z_{mn}),$$

$$c^T = (c_{11}, c_{12}, \dots, c_{mn}), \quad d^T = (a_1, a_2, \dots, a_m, b_1, \dots, b_n)$$

and

$$A = \begin{bmatrix} 1 & 1 & \dots & 1 & & & & & & & & & \\ & & & & 1 & 1 & \dots & 1 & & & & & \\ & & & & & & & & \dots & & & & \\ & & & & & & & & & 1 & 1 & \dots & 1 \\ 1 & & & & 1 & & & & & 1 & & & \\ & 1 & & & & 1 & & & & & 1 & & \\ & & \dots & & & & \dots & & \dots & & & \dots & \\ & & & 1 & & & & 1 & & & & & 1 \end{bmatrix}.$$

Let S_x and S_{yz} denote the primal and dual feasible regions, i.e.,

$$S_x = \{x : Ax = d,\ x \geq 0\},$$

$$S_{yz} = \left\{(y, z) : A^T y + z = c,\ y_1 = 0,\ z \geq 0\right\}.$$

From the duality theorem of the linear programming, a primal feasible solution x^* and a dual feasible solution (y^*, z^*) are optimal if and only if the complementarity condition

(3) $\quad x^*_{ij} z^*_{ij} = 0 \text{ for } (i, j) \in N$

holds. We call the pair of primal and dual feasible solutions a feasible point or simply a point. Each of Algorithms **A** and **B** generates a sequence of feasible points $(x^k, y^k, z^k) \in S_x \times S_{yz}$ for $k = 0, 1, \dots$ such that the complementarity values $x^{kT} z^k$ decrease. Each algorithm consists of the following five steps:

Step 1: Find an initial point $(x^0, y^0, z^0) \in S_x \times S_{yz}$ and set $k \leftarrow 0$.

Step 2: Compute a feasible direction $(\Delta x, \Delta y, \Delta z)$.

Step 3: Get a step size t and compute the next point by

$$(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) - t(\Delta x, \Delta y, \Delta z).$$

Step 4: If a stopping criterion holds then go to Step 5, otherwise set $k \leftarrow k + 1$ and return to Step 2.

Step 5: Compute an optimal solution x^* of (P) from the last point $(x^{k+1}, y^{k+1}, z^{k+1})$.

3. Algorithm A

Algorithm **A** consists of five steps given in the previous section. Now we describe each step in detail.

In Step 1, we take the following point as the initial point of Algorithm **A**:

$$x^0_{ij} = a_i b_j / r \text{ for } (i, j) \in N, \quad y^0 = 0, \quad z^0 = c,$$

where

(4) $\quad r = \sum_{i \in I} a_i = \sum_{j \in J} b_j.$

In Steps 2 and 3, we use the method of [7] so that the iteration number is bounded by a polynomial. According to [7], we shall show how to compute the feasible direction and the step size t. The direction $(\Delta x, \Delta y, \Delta z)$ is Newton direction at the point (x^k, y^k, z^k) for the system

$$Xz = \sigma f^k_{ave} e, \; Ax = d, \; A^T y + z = c, \; y_1 = 0,$$

where σ is a constant in $(0,1)$, $f^k_{ave} = x^{kT} z^k / mn$, $e^T = (1,1,...,1)$ and $X = diag(x)$, which is the mn diagonal matrix with the diagonal elements x_{ij} $((i,j) \in N)$. Then the direction $(\Delta x, \Delta y, \Delta z)$ is the solution of the linear system

$$(5) \quad \begin{array}{rcl} Z^k \Delta x + X^k \Delta z & = & p, \\ A \Delta x & = & 0, \\ A^T \Delta y + \Delta z & = & 0, \\ \Delta y_1 & = & 0, \end{array}$$

where $X^k = diag(x^k)$, $Z^k = diag(z^k)$ and $p = X^k z^k - \sigma f^k_{ave} e$.

In order to compute the step size t, they [7] introduce the following functions:

$$f_{ij}(t) = (x^k_{ij} - t\Delta x_{ij})(z^k_{ij} - t\Delta z_{ij}) \text{ for } (i,j) \in N,$$

$$g(t) = f^k_{min} - \left(f^k_{min} - \tau f^k_{ave}\right) t,$$

where τ is a constant in $(0, \sigma)$ and $f^k_{min} = \min_{(i,j)\in N} x^k_{ij} z^k_{ij}$. Then the step size t is computed as the largest value in $[0,1]$ such that

$$f_{ij}(t) \geq g(t) \text{ for } (i,j) \in N,$$

i.e., the value $f_{ij}(t)$ of each complementarity component is bounded by the positive value $g(t)$

In Step 4, we have to employ a stopping criterion such that we can compute an optimal solution. Such a criterion is given by

$$x_{ij} < \frac{1}{mn - m - n + 2} \text{ or } z_{ij} < \frac{1}{m+n} \text{ for } (i,j) \in N,$$

or equivalently

$$(6) \quad P(x) \cup Q(z) = N,$$

where

$$(7) \quad P(x) = \left\{(i,j) \in N : x_{ij} < \frac{1}{mn - m - n + 2}\right\},$$

$$Q(z) = \left\{(i,j) \in N : z_{ij} < \frac{1}{m+n}\right\}.$$

Theorem 1 *If the condition (6) holds for feasible solutions $x \in S_x$ and $(y,z) \in S_{yz}$ then there is a primal feasible solution x^* which satisfies*

$$(8) \quad x^*_{ij} = 0 \text{ for } (i,j) \in P(x),$$

and such x^* is optimal.

Proof: Let m_1 and m_2 be the dimensions of the convex sets S_x and $S_z = \{z : (y, z) \in S_{yz}\}$, respectively. Since A is the incidence matrix of the transportation problem (P), the rank of A is $m + n - 1$. So we have

(9) $\quad m_1 \leq mn - m - n + 1 \quad$ and $\quad m_2 \leq m + n - 1.$

From Caratheodory's theorem ([12]), we have

$$x = \sum_{h=1}^{m_1+1} \lambda_h q^h + q', \ \sum_{h=1}^{m_1+1} \lambda_h = 1, \ \lambda_h \geq 0 \ \text{ for } \ h = 1, 2, \ldots, m_1 + 1,$$

where q^h $(h = 1, 2, \ldots, m_1 + 1)$ are vertices of S_x and q' is an unbounded direction of S_x (in this case, $q' = 0$ because S_x is bounded). Then there is an index ℓ such that $\lambda_\ell \geq 1/(m_1+1)$. Since $q^h \geq 0$ and $q' \geq 0$, we have

(10) $\quad q^\ell \leq \dfrac{1}{\lambda_\ell} x \leq (m_1 + 1)x.$

From (7), (9) and (10), we obtain

$$q^\ell_{ij} < 1 \ \text{ for } \ (i, j) \in P(x).$$

Since all the components of each vertex of S_x are integral, we have (8) for $x^* = q^\ell$.

In the same way, we can show that there is a dual feasible solution (y^*, z^*) which satisfies

(11) $\quad z^*_{ij} = 0 \ \text{ for } \ (i, j) \in Q(z).$

From (6), (8) and (11), the complementarity condition (3) holds. Therefore x^* is optimal. □

In Step 5, we consider the following problem

(12) $\quad Ax = d, \ x \geq 0, \ x_{ij} = 0 \ \text{ for } \ (i, j) \in P(x^{k+1}),$

where x^{k+1} is the last primal feasible solution obtained in Step 4. From Theorem 1, the solution of the above problem (12) exists and it is an optimal solution of (P). We can easily convert (12) into a max flow problem which will be easily solved (see [10]).

4. Algorithm B

Algorithm **B** consists of five steps given in Section 2. Algorithm **B** is the same as Algorithm **A** in Steps 4 and 5. Here we describe Steps 1, 2 and 3 of Algorithm **B**.

Algorithm **B** is based on the $O(\bar{n}^{3.5}L)$ method of Kojima, Mizuno and Yoshise [8]. For a constant $\alpha \in (0, 0.1]$, it generates a sequence of feasible points in the set

$$S(\alpha) = \left\{(x, y, z) \in S_x \times S_{yz} : \|Xz - f_{ave}e\| \leq \alpha f_{ave}, \ f_{ave} = x^T z/mn\right\}.$$

In order to find an initial point, we construct the following artificial transportation problem:

$$\begin{array}{lll} (\bar{P}) \quad \min & \sum_{(i,j)\in \bar{N}} \bar{c}_{ij}\bar{x}_{ij}, & \\ \quad\quad \text{subject to} & \sum_{j\in \bar{J}} \bar{x}_{ij} = \bar{a}_i & \text{for } i \in \bar{I}, \\ & \sum_{i\in \bar{I}} \bar{x}_{ij} = \bar{b}_j & \text{for } j \in \bar{J}, \\ & \bar{x}_{ij} \geq 0 & \text{for } (i, j) \in \bar{N}, \end{array}$$

where

$$\bar{I} = \{1, 2, \ldots, m + n\}, \ \ \bar{J} = \{1, 2, \ldots, n + m\}, \ \ \bar{N} = \bar{I} \times \bar{J},$$

(13) $\bar{a}_i = \begin{cases} a_i + M_{ab} & \text{for } i \in I, \\ M_{ab} & \text{otherwise,} \end{cases} \quad \bar{b}_j = \begin{cases} b_j + M_{ab} & \text{for } j \in J, \\ M_{ab} & \text{otherwise,} \end{cases}$

(14) $\bar{c}_{ij} = \begin{cases} c_{ij} & \text{for } (i,j) \in N, \\ 0 & \text{for } j = n+i,\ i \in I \text{ and } i = m+j,\ j \in J, \\ M_c & \text{otherwise,} \end{cases}$

for integers M_{ab} and M_c. We illustrate the graph of the artificial problem in Fig. 1. The next theorem shows the relation between optimal solutions of (P) and $(\bar{P})$.

Theorem 2 *Suppose that $M_c \geq M$ and $M_{ab} \geq 0$. For any optimal solution $\bar{x}^*$ of $(\bar{P})$, the mn vector $\bar{x}^*(N)$ of $\bar{x}^*_{ij}$ for $(i,j) \in N$ is an optimal solution of (P).*

Proof: The dual problem of $(\bar{P})$ is formulated as

$$(\bar{D}) \quad \begin{array}{lll} \max & \sum_{i \in \bar{I}} \bar{a}_i \bar{u}_i + \sum_{j \in \bar{J}} \bar{b}_j \bar{v}_j, & \\ \text{subject to} & \bar{u}_i + \bar{v}_j + \bar{z}_{ij} = \bar{c}_{ij} & \text{for } (i,j) \in \bar{N}, \\ & \bar{u}_1 = 0,\ \bar{z}_{ij} \geq 0 & \text{for } (i,j) \in \bar{N}. \end{array}$$

Let x' and (u', v', z') be optimal solutions of (P) and (D), respectively. Then we have

(15) $x'_{ij} z'_{ij} = 0$ for $(i,j) \in N$,

(16) $u'_i + v'_j \leq c_{ij}$ for $(i,j) \in N$.

Since (u', v', z') is optimal, there exist an $i' \in J$ for each $i \in I$ and a $j' \in I$ for each $j \in J$ such that

(17) $u'_i + v'_{i'} = c_{ii'}$,

(18) $u'_{j'} + v'_j = c_{j'j}$.

Let

(19) $\bar{x}'_{ij} = \begin{cases} x'_{ij} & \text{for } (i,j) \in N, \\ M_{ab} & \text{for } j = n+i,\ i \in I \text{ and } i = m+j,\ j \in J, \\ 0 & \text{otherwise,} \end{cases}$

(20) $\bar{u}'_i = \begin{cases} u'_i & \text{for } i \in I, \\ -v'_{i-m} & \text{otherwise,} \end{cases} \quad \bar{v}'_j = \begin{cases} v'_j & \text{for } j \in J, \\ -u'_{j-n} & \text{otherwise,} \end{cases}$

(21) $\bar{z}'_{ij} = \bar{c}_{ij} - \bar{u}'_i - \bar{v}'_j$ for $(i,j) \in \bar{N}$.

Now we shall show that $\bar{x}'$ and $(\bar{u}', \bar{v}', \bar{z}')$ are optimal solutions of $(\bar{P})$ and $(\bar{D})$, respectively. We easily see that $\bar{x}'$ is a feasible solution of $(\bar{P})$. So we shall first prove that $(\bar{u}', \bar{v}', \bar{z}')$ is a feasible solution of $(\bar{D})$, i.e., $\bar{z}' \geq 0$. From (14), (16), (17), (18), (20) and (21), for each $i,\ g \in I\ (i \neq g)$ and $j,\ h \in J\ (j \neq h)$, we have

$$\begin{array}{lrcl} (22) & \bar{z}'_{ij} & = & c_{ij} - u'_i - v'_j = z'_{ij} \geq 0, \\ (23) & \bar{z}'_{in+i} & = & 0 - u'_i - (-u'_i) = 0, \\ (24) & \bar{z}'_{m+jj} & = & 0 - (-v'_j) - v'_j = 0, \\ (25) & \bar{z}'_{gn+i} & = & M_c - u'_g + u'_i \\ & & = & M_c - u'_g + (c_{ii'} - v'_{i'}) \\ & & \geq & M_c - c_{gi'} + c_{ii'} > 0, \\ (26) & \bar{z}'_{m+jh} & = & M_c + v'_j - v'_h \\ & & \geq & M_c - c_{j'h} + c_{j'j} > 0, \\ (27) & \bar{z}'_{m+jn+i} & = & M_c + v'_j + u'_i \\ & & = & M_c + (c_{j'j} - u'_{j'}) + (c_{ii'} - v'_{i'}) \\ & & \geq & M_c - c_{j'i'} + c_{j'j} + c_{ii'} > 0. \end{array}$$

Hence $\bar{z}' \geq 0$. From (15), (19), (23) and (24), we also see that

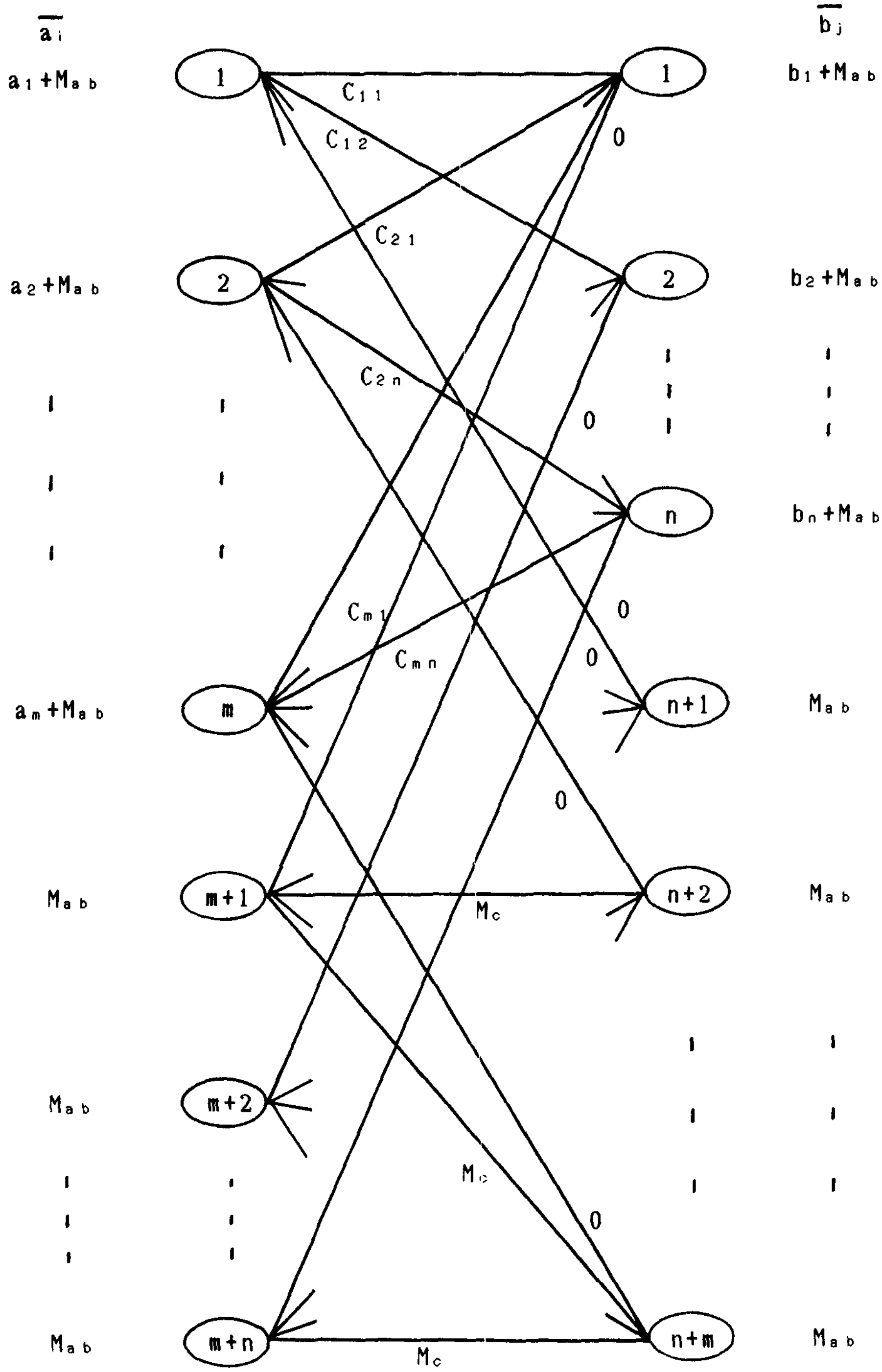

Fig1.The artifitial problem $(\overline{P})$

$$\bar{x}'_{ij}\bar{z}'_{ij} = 0 \ \text{ for } \ (i,j) \in \bar{N}.$$

This implies that $(\bar{u}', \bar{v}', \bar{z}')$ is an optimal solution of $(\bar{D})$.

Since $\bar{x}^*$ is an optimal solution of $(\bar{P})$, we have

$$(28) \quad \bar{x}^*_{ij}\bar{z}'_{ij} = 0 \ \text{ for } \ (i,j) \in \bar{N}.$$

From (25), (26), (27) and (28), we have

$$\begin{array}{lrcl} (29) & \bar{x}^*_{gn+i} & = & 0 \ \text{ for } \ i,\ g \in I \ (i \neq g), \\ (30) & \bar{x}^*_{m+jh} & = & 0 \ \text{ for } \ j,\ h \in J \ (j \neq h), \\ & \bar{x}^*_{m+jn+i} & = & 0 \ \text{ for } \ (i,j) \in N. \end{array}$$

So we see

$$(31) \quad \bar{x}^*_{in+i} = \bar{b}_{n+i} - \sum_{g \neq i} \bar{x}^*_{gn+i} = M_{ab} \ \text{ for } \ i \in I,$$

$$(32) \quad \bar{x}^*_{m+jj} = \bar{a}_{m+j} - \sum_{h \neq j} \bar{x}^*_{m+jh} = M_{ab} \ \text{ for } \ j \in J.$$

From (29), (30), (31) and (32), we obtain

$$\begin{array}{rclclcl} \sum_{j \in J} \bar{x}^*_{ij} & = & \bar{a}_i - \sum_{g \in I} \bar{x}^*_{in+g} & = & \bar{a}_i - M_{ab} & = & a_i \ \text{ for } \ i \in I, \\ \sum_{i \in I} \bar{x}^*_{ij} & = & \bar{b}_j - \sum_{h \in J} \bar{x}^*_{m+hj} & = & \bar{b}_j - M_{ab} & = & b_j \ \text{ for } \ j \in J. \end{array}$$

Therefore $\bar{x}^*(N)$ is a feasible solution of (P). Since (u', v', z') is a feasible solution of (D) and (28) holds, $\bar{x}^*(N)$ is an optimal solution of (P). □

From the above theorem, we may solve the artificial problem $(\bar{P})$ instead of (P). Let

$$\beta = \lceil 4(m+n)/\alpha \rceil.$$

Algorithm **B** solves the artificial problem $(\bar{P})$ with $M_{ab} = \beta M$ and $M_c = M$. Let

$$\bar{r} = \sum_{i \in \bar{I}} \bar{a}_i = \sum_{j \in \bar{J}} \bar{b}_j,$$

$$\bar{S}(\alpha) = \left\{ (\bar{x}, \bar{u}, \bar{v}, \bar{z}) : \begin{array}{l} \|\bar{X}\bar{z} - \bar{f}_{ave}\bar{e}\| \leq \alpha \bar{f}_{ave}, \ \ \bar{f}_{ave} = \bar{x}^T\bar{z}/(m+n)^2, \\ \bar{x} \text{ and } (\bar{u}, \bar{v}, \bar{z}) \text{ are feasible solutions of } (\bar{P}) \text{ and } (\bar{D}) \end{array} \right\},$$

where $\bar{X} = diag(\bar{x})$ and $\bar{e}^T = (1, 1, ..., 1)$. We take the following point $(\bar{x}^0, \bar{u}^0, \bar{v}^0, \bar{z}^0)$ as the initial point of Algorithm **B**:

$$\bar{x}^0_{ij} = \bar{a}_i\bar{b}_j/\bar{r} \ \text{ for } \ (i,j) \in \bar{N},$$

$$\bar{u}^0_i = 0 \ \text{ for } \ i \in \bar{I}, \quad \bar{v}^0_j = -\beta M \ \text{ for } \ j \in \bar{J}, \quad \bar{z}^0_{ij} = \bar{c}_{ij} + \beta M \ \text{ for } \ (i,j) \in \bar{N}.$$

Theorem 3 *Suppose that $M_{ab} = \beta M$ and $M_c = M$. Then the above point $(\bar{x}^0, \bar{u}^0, \bar{v}^0, \bar{z}^0)$ belongs to $\bar{S}(\alpha)$.*

Proof: It will be trivial that the point is feasible. So we shall only show that

$$\|\bar{X}^0\bar{z}^0 - \bar{f}^0_{ave}\bar{e}\| \ \leq \ \alpha \bar{f}^0_{ave},$$

where $\bar{X}^0 = diag(\bar{x}^0)$ and $\bar{f}^0_{ave} = \bar{x}^{0T}\bar{z}^0/(m+n)^2$. From (13) and (14), we have

$$\beta M \leq \bar{a}_i \leq (1+\beta)M \text{ for } i \in \bar{I},$$

$$\beta M \leq \bar{b}_j \leq (1+\beta)M \text{ for } j \in \bar{J},$$

$$0 \leq \bar{c}_{ij} \leq M \text{ for } (i,j) \in \bar{N}.$$

From the above inequalities and the definition of the initial point, we see

$$\beta^3 M^3/\bar{r} \leq \bar{x}^0_{ij}\bar{z}^0_{ij} \leq (1+\beta)^3 M^3/\bar{r} \text{ for } (i,j) \in \bar{N},$$

$$\beta^3 M^3/\bar{r} \leq \bar{f}^0_{ave} \leq (1+\beta)^3 M^3/\bar{r}.$$

Hence we obtain

$$\begin{aligned} \|\bar{X}^0\bar{z}^0 - \bar{f}^0_{ave}\bar{e}\| &\leq (m+n) \max_{(i,j)\in\bar{N}} \mid \bar{x}^0_{ij}\bar{z}^0_{ij} - \bar{f}^0_{ave} \mid \\ &\leq (m+n)\left\{(1+\beta)^3 - \beta^3\right\} M^3/\bar{r} \\ &\leq 4(m+n)\beta^2 M^3/\bar{r} \\ &\leq \alpha\beta^3 M^3/\bar{r} \\ &\leq \alpha \bar{f}^0_{ave}, \end{aligned}$$

where the third inequality follows from $\beta \geq 4$. □

In Steps 2 and 3, we use the method of [8] so that the iteration number is bounded by a polynomial. The feasible direction $(\Delta x, \Delta y, \Delta z)$ is the solution of the system (5) for the artificial problem $(\bar{P})$, where

$$\sigma = 1 - \frac{\alpha}{(1-\alpha)\sqrt{\bar{n}}}.$$

We always set the step size $t = 1$ because the next point for $t = 1$ belongs to $\bar{S}(\alpha)$ whenever $\alpha \in (0, 0.1]$ (see [8]).

5. Computational complexities

In this section, we obtain the computational complexities of Algorithms **A** and **B**.

Theorem 4 *Algorithm* **A** *computes an optimal solution of* (P) *with* $O(m^3n^2 \log nM + n^3)$ *arithmetic operations.*

Recall that $m \leq n$. The above theorem is obtained from the following two lemmas and the fact that there is an $O(n^3)$ algorithm for solving the max flow problem appeared in Step 5 (see [10]).

Lemma 5 *Algorithm* **A** *terminates in* $O(mn \log nM)$ *iterations.*

Proof: From Corollary 1 of [7], Algorithm **A** finds a point (x^k, y^k, z^k), which satisfies $x^{kT}z^k < \epsilon$ for $\epsilon > 0$, after

(33) $O(mn(\log \pi^0 + \log(x^{0T}z^0/\epsilon)))$

iterations, where (x^0, y^0, z^0) is the initial point and

$$\pi^0 = \frac{x^{0T}z^0}{mn \min_{(i,j)\in N} x^0_{ij}z^0_{ij}}.$$

From the definition of the initial point of Algorithm **A**, we see that

$$x^{0T}z^0 = \sum_{(i,j)\in N} a_i b_j c_{ij}/r \le mnM^3/r,$$

$$x^0_{ij}z^0_{ij} = a_i b_j c_{ij}/r \ge 1/r.$$

So we have

(34) $\pi^0 \;\le\; M^3.$

Let $\epsilon = 1/n^4$. If $x^{kT}z^k < \epsilon$, we have

$$x^k_{ij} < \frac{1}{n^2} \text{ or } z^k_{ij} < \frac{1}{n^2} \text{ for } (i,j) \in N.$$

The above inequalities imply that the stopping criterion (6) holds for $x = x^k$ and $z = z^k$. Then we see

(35) $x^{0T}z^0/\epsilon \le n^6M^3.$

From (33), (34) and (35), the stopping criterion (6) holds after $O(mn \log nM)$ iterations.

Lemma 6 *Each iteration of Algorithm* **A** *requires at most $O(m^2n)$ arithmetic operations.*

Proof: From the system (5), we have

(36) $A(Z^k)^{-1}X^kA^T\Delta y = -A(Z^k)^{-1}p$ and $\Delta y_1 = 0.$

Let

- $w_{ij} = x^k_{ij}/z^k_{ij}$,
- B be the $m \times n$ matrix with (i,j) elements w_{ij},
- D_1 be the $m \times m$ diagonal matrix with ith diagonal elements $\sum_{j\in J} w_{ij}$,
- D_2 be the $n \times n$ diagonal matrix with jth diagonal elements $\sum_{i\in I} w_{ij}$.

Since A is the incidence matrix and $\Delta y = (\Delta u, \Delta v)$, the first system of (36) is represented as

$$\begin{aligned} D_1\Delta u + B\Delta v &= p_1, \\ B^T\Delta u + D_2\Delta v &= p_2 \end{aligned}$$

for $(p_1, p_2) = -A(Z^k)^{-1}p$. From the above two equalities, we have

$$(D_1 - BD_2^{-1}B^T)\Delta u = p_1 - BD_2^{-1}p_2.$$

We can compute the matrix $BD_2^{-1}B^T$ with $O(m^2n)$ arithmetic operations and solve the above linear equation under $\Delta u_1 = 0$ with $O(m^3)$ arithmetic operations. Since we can compute the other part with at most $O(mn)$ arithmetic operations, we have the result. □

Theorem 7 *Algorithm* **B** *compute an optimal solution of (P) with $O(n^4 \log nM)$ arithmetic operations.*

This theorem is obtained from the following two lemmas.

Lemma 8 *Algorithm* **B** *terminates in $O(n \log nM)$ iterations.*

Proof: Algorithm **B** is the same as the $O(\bar{n}^{3.5}L)$ algorithm of [8] in Steps 2 and 3. Since the number of variables of the artificial problem $(\bar{P})$ is $(m+n)^2$, Algorithm **B** finds a point $(\bar{x}^k, \bar{y}^k, \bar{z}^k)$, which satisfies $\bar{x}^{kT}\bar{z}^k \leq \epsilon$ for $\epsilon > 0$, after

(37) $O(n \log(\bar{x}^{0T}\bar{z}^0/\epsilon))$

iterations, where $(\bar{x}^0, \bar{y}^0, \bar{z}^0)$ is the initial point. Hence the result is shown in the same way as the proof of Lemma 5. □

Lemma 9 *Each iteration of Algorithm* **B** *requires at most* $O(n^3)$ *arithmetic operations.*

Proof: Since the artificial transportation problem $(\bar{P})$ has $(m+n)$ supply points and $(m+n)$ demand points, it follows from Lemma 6. □

6. Conclusions

In this paper, we propose two interior point algorithms, Algorithm **A** and Algorithm **B**, for the Hitchcock transportation problem (P) with m supply points and n demand points. Under the conditions that $m \leq n$ and all the data are positive integers which are less than or equal to an integer M, we show that Algorithm **A** requires at most

$$O(m^3n^2 \log nM + n^3)$$

arithmetic operations and Algorithm **B** requires at most

$$O(n^4 \log nM)$$

arithmetic operations.

In this paper, we only show the theoretical computational complexities of the algorithms and do not refer to a practical implementation. According to our numerical experiments for small size problems ($m \leq 50$ and $n \leq 50$), the algorithms were not superior to the primal-dual simplex algorithm. In order to see the efficiency of the interior point algorithms, we need to improve the algorithms from a practical point of view and to attempt the numerical experiments for large size problems.

Acknowledgment The authors wish to thank Professor Masakazu Kojima, Professor Masao Mori and Miss Akiko Yoshise for their helpful suggestions. They are also grateful to the referees and the editor for their valuable comments. Part of the research of the first author was supported by Grant-in-Aids for Encouragement of Young Scientists (63730014) and for General Scientific Research (63490010) of The Ministry of Education, Science and Culture.

References

[1] Derigs, U. : *Programming in Networks and Graphs.* Lecture Notes in Economics and Mathematical Systems 300, Springer-Verlag, Berlin, New York, 1988.

[2] Edmonds, J. and Karp, R. M. : Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *Journal of the Association for Computing Machinery,* Vol.19, No.2 (1972), 248-264.

[3] Gonzaga, C. C. : An Algorithm for Solving Linear Programming Problems in $O(n^3L)$ Operations. *Progress in Mathematical Programming, Interior Point and Related Methods* (ed. N. Megiddo). Springer-Verlag, New York, 1988, 1-28.

[4] Ikura, Y. and Nemhauser, G. L. : A Polynomial-Time Dual Simplex Algorithm for the Transportation Problem. *OR&IE Technical Report No. 602*, Cornell University (1983).

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

[6] Khachiyan, L. G. : A Polynomial Algorithm in Linear Programming. *Soviet Mathematics Doklady*, Vol.20 (1979), 191-194.

[7] Kojima, M., Mizuno, S. and Yoshise, A. : A Primal-Dual Interior Point Algorithm for Linear Programming. *Progress in Mathematical Programming, Interior Point and Related Methods* (ed. N. Megiddo). Springer-Verlag, New York, 1988, 29-47.

[8] Kojima, M., Mizuno, S. and Yoshise, A. : A Polynomial-Time Algorithm for a Class of Linear Complementarity Problems. *Mathematical Programming*, Vol.44, No.1 (1989), 1-26.

[9] Monteiro, R. C. and Adler, I. : Interior Path Following Primal-Dual Algorithms. Part I : Linear Programming. *Mathematical Programming*, Vol.44, No.1 (1989), 27-42.

[10] Papadimitriou, C. H. and Steiglitz, K. : *Combinatorial Optimization : Algorithms and Complexity*. Prentice-Hall, Inc., New Jersey, 1982.

[11] Renegar, J. : A Polynomial-Time Algorithm, Based on Newton's Method, for Linear Programming. *Mathematical Programming*, Vol.40, No.1 (1988), 59-93.

[12] Rockafellar, R. T. :*Convex Analysis*. Princeton University Press, New Jersey, 1972.

[13] Tardos, E. : A Strongly Polynomial Minimum Cost Circularion Algorithm. *Combinatorica*, Vol.5, No.3 (1985), 247-255.

[14] Vaidya, P. M. : An Algorithm for Linear Programming which Requires $O(((m+n)n^2 + (m+n)^{1.5}n)L)$ Arithmetic Operations. *AT&T Bell Laboratories, Merray Hill*, New Jersey (1987).

Shinji MIZUNO: Department of Industial Engineering and Management, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152, Japan.