

AN INAPPROXIMABILITY OF TRANSSHIPMENT PROBLEMS WITH PERMUTABLE TRANSIT VECTORS

Kougaku Yamashita Yoshiyuki Karuno
Kyoto Institute of Technology

Mingzhe Lu
Dongbei University of Finance & Economics

(Received March 29, 2010; Revised June 29, 2010)

Abstract We investigate the hardness of approximating a transshipment problem with a permutable transit vector. We have already known that the transshipment problem of minimizing the total shipping cost is NP-hard. In this paper, we prove that even if the unit shipping cost associated with every arc is given as a positive integer, there is no polynomial time constant factor approximation algorithm for the transshipment problem under the hypothesis that $P \neq NP$. We obtain the inapproximability result by a gap-introducing reduction from an NP-complete problem by the name of 3DM. On the other hand, we propose a preparatory heuristic algorithm to the transshipment problem, which is based on a relaxation into the minimum cost flow problem and the minimum weight perfect matching on a bipartite graph. We also provide an upper bound of the total shipping cost obtained by the proposed heuristic.

Keywords: Combinatorial optimization, transportation, inapproximability, gap technique

1. Introduction

In this paper, we investigate the hardness of approximating a transshipment problem with a permutable transit vector. The transshipment problem is defined on a digraph with three kinds of vertices, i.e., supply, transit and demand vertices. The supply and demand vectors are fixed, while the transit vector is permutable. That is, a transit vertex is one-to-one assigned an entry of the transit vector as the amount of flow passing through the transit vertex so that the total shipping cost is minimized. We call the problem TRANSSHIP for short. Problem TRANSSHIP is a generalization of the transportation problem with a permutable demand vector, which appears in several applications such as an assignment model of component types to storage positions in production of printed circuit boards and a location model of warehouse facilities (Meusel and Burkard [10]). From the NP-hardness of the transportation problem with a permutable demand vector (Hujter, Klinz and Woeginger [6], Meusel and Burkard [10]), we have already known that problem TRANSSHIP is also NP-hard. On the other hand, the transportation problem can be solved in polynomial time if each entry of the permutable demand vector takes zero, one, or two (Hujter, Klinz and Woeginger [6]).

Recently, it has been showed that problem TRANSSHIP remains NP-hard even if each entry of the permutable transit vector takes either zero or two (Karuno, Tachibana and Yamashita [8]). The complexity result is proved by a polynomial time reduction from an NP-complete problem by the name of 3DM (which is the abbreviation for 3-DIMENSIONAL MATCHING, e.g., see Garey and Johnson [4]). Also, it has been showed that problem TRANSSHIP can be solved in polynomial time if each entry of the permutable transit vector takes either zero or one (Karuno, Tachibana and Yamashita [8]). A minimum cost

flow algorithm (Edmonds and Karp [3], Tomizawa [12]) is applied to the special case of problem TRANSSHIP so that the exact algorithm runs in polynomial time.

In this paper, we prove that even if the unit shipping cost associated with every arc is given as a positive integer, there is no polynomial time constant factor approximation algorithm for problem TRANSSHIP under the hypothesis that $P \neq NP$ (e.g., see also Garey and Johnson [4] for the hypothesis). We obtain the inapproximability result by a gap-introducing reduction (e.g., see Asano [1]) from the NP-complete 3DM, which means that any approximate solution with a constant factor is as hard to determine as the optimal solution.

On the other hand, we propose a preparatory heuristic algorithm to problem TRANSSHIP, which is based on a relaxation into the minimum cost flow problem (e.g., see additionally Asano et al. [2]) and the minimum weight perfect matching on a bipartite graph (the matching problem is often called the *assignment problem*, e.g., see also Asano et al. [2]). The proposed heuristic runs in polynomial time. We provide an upper bound of the total shipping cost obtained by the proposed heuristic, which involves the minimum weight of a perfect matching on the bipartite graph (of course, it has no theoretical guarantee of a constant factor under the hypothesis that $P \neq NP$). Also, we generate a few kinds of numerical test instances to examine the performance of the proposed heuristic empirically.

The remainder of this paper is organized as follows. In Section 2, we give the mathematical description of problem TRANSSHIP. In Section 3, we prove the inapproximability of problem TRANSSHIP by a gap-introducing reduction from problem 3DM. In Section 4, we propose the preparatory heuristic algorithm, and provide an upper bound of the total shipping cost obtained by the heuristic. In Section 5, we examine the performance of the heuristic empirically. In Section 6, we make some concluding remarks.

2. Problem Description

Problem TRANSSHIP is defined on a digraph $D = (V_1 \cup V_T \cup V_2, A_1 \cup A_2)$, where $V_1 = \{i \mid i = 1, 2, \dots, m_1\}$ is its set of *sources* (i.e., supply vertices), $V_T = \{j \mid j = 1, 2, \dots, n\}$ is its set of *relays* (i.e., transit vertices), $V_2 = \{k \mid k = 1, 2, \dots, m_2\}$ is its set of *sinks* (i.e., demand vertices), $A_1 = \{(i, j) \mid i \in V_1, j \in V_T\}$ is its set of arcs from the sources to the relays, and $A_2 = \{(j, k) \mid j \in V_T, k \in V_2\}$ is its set of arcs from the relays to the sinks. Figure 1 illustrates the network of problem TRANSSHIP.

Each source $i \in V_1$ has a positive integral supply a_i , and each sink $k \in V_2$ has a positive integral demand b_k . The amount of flow out of a source must equal its supply, and the amount of flow into a sink must equal its demand. We denote a *supply vector* and a *demand vector* by $a = (a_1, a_2, \dots, a_{m_1})$ and $b = (b_1, b_2, \dots, b_{m_2})$, respectively.

We are also given a vector $r = (r_1, r_2, \dots, r_n)$ of transits. Each entry r_j of the *transit vector* r is a non-negative integer, $j = 1, 2, \dots, n$. Let $\varphi = (\varphi(1), \varphi(2), \dots, \varphi(n))$ be a permutation on $\{1, 2, \dots, n\}$, which is a decision variable of problem TRANSSHIP. Then the amount of flow into a relay $j \in V_T$ is required to equal transit $r_{\varphi(j)}$, and the amount of flow out of the relay is also required to equal the $r_{\varphi(j)}$. We refer to the φ as a *transit permutation*. For the total amounts of supplies, transits and demands, it is assumed

$$\sum_{i \in V_1} a_i = \sum_{j \in V_T} r_j = \sum_{k \in V_2} b_k \quad (2.1)$$

as the feasibility condition.

The unit shipping cost from $i \in V_1$ to $j \in V_T$ is denoted by $p_{ij} \geq 0$. If there is no arc from i to j in a certain practical situation, we take p_{ij} to be sufficiently large in modeling.

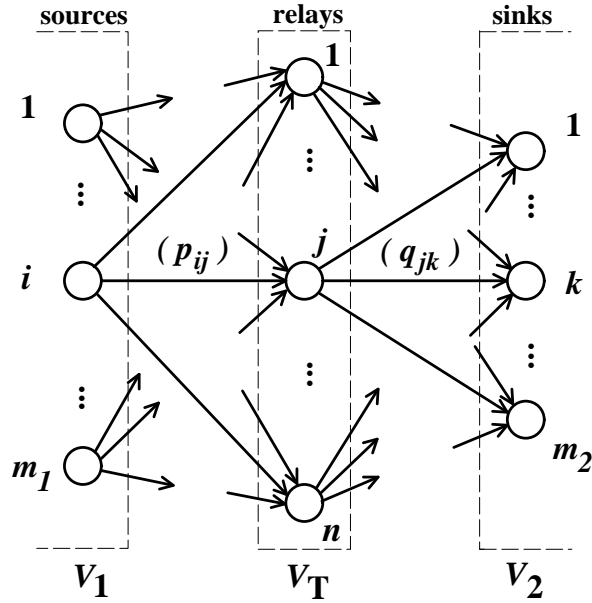


Figure 1: Network of problem TRANSSHIP

Thus, without loss of generality, we can assume that there is an arc from each source to each relay. The unit shipping cost from $j \in V_T$ to $k \in V_2$ is denoted by $q_{jk} \geq 0$. We define $\theta_1 = m_1 \times n = |A_1|$ and $\theta_2 = m_2 \times n = |A_2|$ as the numbers of arcs from sources to relays and from relays to sinks, respectively. We represent by $I = (D, a, b, r, p, q)$ an instance of problem TRANSSHIP.

Let x_{ij} be the amount of flow from $i \in V_1$ to $j \in V_T$, and let y_{jk} be the amount of flow from $j \in V_T$ to $k \in V_2$, which are also decision variables of problem TRANSSHIP. For a transit permutation φ , a solution (x, y) of flow is *feasible* if it satisfies the mass balance constraint at each of sources, relays and sinks, where $x \in R_+^{\theta_1}$ and $y \in R_+^{\theta_2}$. For a given transit permutation φ , let $z(\varphi; x, y)$ be the total shipping cost of a feasible solution (x, y) of flow. Then the problem of minimizing the total shipping cost under a given φ is formulated as follows.

$$\text{minimize} \quad z(\varphi; x, y) = \sum_{i \in V_1} \sum_{j \in V_T} p_{ij} x_{ij} + \sum_{j \in V_T} \sum_{k \in V_2} q_{jk} y_{jk} \tag{2.2}$$

$$\text{subject to} \quad \sum_{j \in V_T} x_{ij} = a_i \quad \text{for } i \in V_1, \tag{2.3}$$

$$\sum_{i \in V_1} x_{ij} = \sum_{k \in V_2} y_{jk} = r_{\varphi(j)} \quad \text{for } j \in V_T, \tag{2.4}$$

$$\sum_{j \in V_T} y_{jk} = b_k \quad \text{for } k \in V_2, \tag{2.5}$$

$$x_{ij} \geq 0 \quad \text{for } i \in V_1, j \in V_T, \tag{2.6}$$

$$y_{jk} \geq 0 \quad \text{for } j \in V_T, k \in V_2. \tag{2.7}$$

Equations (2.3)–(2.5) represent the mass balance constraints at the vertices. In partic-

ular, Equation (2.4) is the remarkable restriction of problem TRANSSHIP. We denote by $\tilde{z}(\varphi)$ the minimum of the total shipping cost $z(\varphi; x, y)$ under a given transit permutation φ , and by $(\tilde{x}(\varphi), \tilde{y}(\varphi))$ a feasible solution of flow that attains $\tilde{z}(\varphi) = z(\varphi; \tilde{x}(\varphi), \tilde{y}(\varphi))$. Notice that given a transit permutation φ , the solution $(\tilde{x}(\varphi), \tilde{y}(\varphi))$ of flow can be computed in polynomial time, for example, by solving the classical transportation problem (e.g., see Katoh [9]).

For a given instance $I = (D, a, b, r, p, q)$ of problem TRANSSHIP, the objective is to find a transit permutation φ^* that satisfies $\tilde{z}(\varphi^*) \leq \tilde{z}(\varphi)$ for any transit permutation φ . We refer to the φ^* as an *optimal permutation*, and $z^* = \tilde{z}(\varphi^*)$ as the *optimal total shipping cost*, or simply as the *optimal value*. Moreover, we call $(\tilde{x}(\varphi^*), \tilde{y}(\varphi^*))$ an *optimal solution of flow*. The $\tilde{z}(\varphi)$ defined in the previous paragraph with a given transit permutation φ is called a *minimal value*, and $(\tilde{x}(\varphi), \tilde{y}(\varphi))$ is called a *minimal solution of flow*.

If the unit shipping costs from $j \in V_T$ to $k \in V_2$ are given by $q_{jk} = 0$ for all arcs $(j, k) \in A_2$, problem TRANSSHIP is equivalent to the transportation problem with a permutable demand vector (Hujter, Klinz and Woeginger [6], Meusel and Burkard [10]). As mentioned in the previous section, problem TRANSSHIP of finding an optimal permutation φ^* is NP-hard (Hujter, Klinz and Woeginger [6], Karuno, Tachibana and Yamashita [8], Meusel and Burkard [10]).

3. An Inapproximability

Let $\ell (> 1)$ be a finite number, which may be a function of the input size of problem TRANSSHIP. For any instance $I = (D, a, b, r, p, q)$ of problem TRANSSHIP, if some polynomial time algorithm delivers an approximate transit permutation φ' such that the total shipping cost satisfies

$$\tilde{z}(\varphi') \leq \ell \cdot z^*, \quad (3.1)$$

it is referred to as a polynomial time approximation algorithm with factor ℓ , or a polynomial time ℓ -approximation algorithm for problem TRANSSHIP. The factor ℓ is often called the *performance ratio* or *approximation ratio*. As the name implies, a polynomial time constant factor approximation algorithm has a constant performance ratio independent of the input size (e.g., see Asano [1]).

Problem TRANSSHIP remains NP-hard even if each entry of the permutable transit vector takes either zero or two, i.e., $r_j \in \{0, 2\}$ for all $j = 1, 2, \dots, n$ (Karuno, Tachibana and Yamashita [8]). The complexity result is obtained by a polynomial time reduction from the following problem.

3DM (3-DIMENSIONAL MATCHING)

Instance: Set $T = \{(\alpha_j, \beta_j, \gamma_j) \mid j = 1, 2, \dots, g\} \subseteq \mathcal{A} \times \mathcal{B} \times \mathcal{R}$ of 3-tuples, where $\mathcal{A} = \{i \mid i = 1, 2, \dots, h\}$, $\mathcal{B} = \{i \mid i = h + 1, h + 2, \dots, 2h\}$ and $\mathcal{R} = \{k \mid k = 2h + 1, 2h + 2, \dots, 3h\}$ are disjoint sets having the same number h of elements.

Question: Does T contain a three-dimensional matching, i.e., a subset $M \subseteq T$ such that $|M| = h$ and no two elements of M agree in any coordinate ?

Problem 3DM has been known to be NP-complete (e.g., see Garey and Johnson [4]). Note that for an instance of problem TRANSSHIP with $z^* = 0$, a polynomial time ℓ -approximation algorithm has to find an optimal permutation φ^* if it exists. Thus, together with the discussion of the proof by Karuno, Tachibana and Yamashita [8], there is no constant factor approximation algorithm for problem TRANSSHIP under the hypothesis

that $P \neq NP$. In this paper, we strengthen the polynomial time reduction to give a gap-introducing reduction, which proves the following theorem.

Theorem 1 For any finite constant $\ell (> 1)$, there is no polynomial time ℓ -approximation algorithm for problem TRANSSHIP even if the unit shipping cost associated with every arc is given as a positive integer under the hypothesis that $P \neq NP$.

Proof. Given an instance of problem 3DM, we transform it polynomially into the following instance of problem TRANSSHIP.

- Digraph: Let $V_1 = \{i \mid i = 1, 2, \dots, 2h\}$, $V_T = \{j \mid j = 1, 2, \dots, g\}$ and $V_2 = \{k \mid k = 1, 2, \dots, h\}$ (and hence, let $m_1 = 2h$, $n = g$ and $m_2 = h$).
- Supplies, transits, and demands: Set $a_i = 1$ for $i = 1, 2, \dots, 2h$. For $j = 1, 2, \dots, h$, set $r_j = 2$ and for $j = h + 1, h + 2, \dots, g$, set $r_j = 0$. Set $b_k = 2$ for $k = 1, 2, \dots, h$. It holds

$$\sum_{i=1}^{m_1} a_i = \sum_{j=1}^n r_j = \sum_{k=1}^{m_2} b_k = 2h.$$

- Unit shipping costs associated with arcs: Set

$$p_{ij} = \begin{cases} 1 & \text{if } i = \alpha_j, \\ 4h\ell & \text{otherwise,} \end{cases} \quad (3.2)$$

$$i = 1, 2, \dots, h, \quad j = 1, 2, \dots, g,$$

$$p_{ij} = \begin{cases} 1 & \text{if } i = \beta_j, \\ 4h\ell & \text{otherwise,} \end{cases} \quad (3.3)$$

$$i = h + 1, h + 2, \dots, 2h, \quad j = 1, 2, \dots, g,$$

$$q_{jk} = \begin{cases} 1 & \text{if } k = \gamma_j - 2h, \\ 2h\ell & \text{otherwise,} \end{cases} \quad (3.4)$$

$$j = 1, 2, \dots, g, \quad k = 1, 2, \dots, h.$$

Figure 2 illustrates the network obtained by the above transformation from problem 3DM. Each relay $j \in V_T$ receives two arcs with $p_{ij} = 1$ and $2h - 2$ arcs with $p_{ij} = 4h\ell$ from the sources, while it gives out one arc with $q_{jk} = 1$ and $h - 1$ arcs with $q_{jk} = 2h\ell$ to the sinks. Note that the unit shipping cost on every arc is given as a positive integer, and hence the optimal value is positive, i.e., $z^* > 0$.

It suffices to show that the transformed instance of problem TRANSSHIP has an optimal permutation φ^* such that the optimal value satisfies

- $z^* \leq 4h$ if the given instance of problem 3DM has a solution M of three-dimensional matching,
- $z^* > 4h\ell$ if the given instance of problem 3DM does not have a solution of three-dimensional matching.

The reason is explained as follows. Suppose that there is a polynomial time ℓ -approximation algorithm for problem TRANSSHIP. Then, it returns the total shipping cost within $4h\ell$ if the transformed instance of problem TRANSSHIP has an optimal permutation φ^* with $z^* = \tilde{z}(\varphi^*) \leq 4h$. Otherwise, it returns the total shipping cost more than $4h\ell$. That is, if there exists a polynomial time ℓ -approximation algorithm for problem TRANSSHIP, it can identify whether the transformed instance of problem TRANSSHIP has an optimal permutation φ^* with $z^* = \tilde{z}(\varphi^*) \leq 4h$. This implies that the polynomial time ℓ -approximation algorithm can solve problem 3DM in polynomial time, which contradicts the NP-completeness of problem 3DM under the hypothesis that $P \neq NP$.

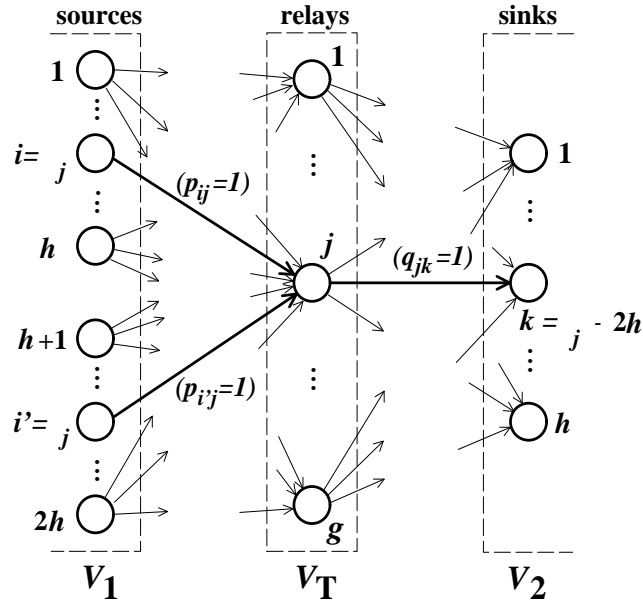


Figure 2: Transformed network of problem TRANSSHIP from 3DM

Suppose that there exists a solution M of three-dimensional matching to the given instance of 3DM. Let $V_M \subseteq V_T$ be the subset of transit vertices corresponding to the 3-tuples of the solution M of three-dimensional matching. Then it satisfies $|V_M| = h$. We assign h transits with $r_j = 2$ to the h vertices of V_M one by one, while we assign $g - h$ transits with $r_j = 0$ to the remaining $g - h$ vertices of $V_T \setminus V_M$ one by one. Let φ_M be a transit permutation based on such an assignment of transits. For the transit permutation φ_M , we construct the following solution (x, y) of flow.

$$x_{ij} = \begin{cases} 1 & \text{if } (\alpha_j, \beta_j, \gamma_j) \in M \text{ and } i = \alpha_j, \\ 0 & \text{otherwise,} \end{cases} \quad (3.5)$$

$$i = 1, 2, \dots, h, \quad j = 1, 2, \dots, g,$$

$$x_{ij} = \begin{cases} 1 & \text{if } (\alpha_j, \beta_j, \gamma_j) \in M \text{ and } i = \beta_j, \\ 0 & \text{otherwise,} \end{cases} \quad (3.6)$$

$$i = h + 1, h + 2, \dots, 2h, \quad j = 1, 2, \dots, g,$$

$$y_{jk} = \begin{cases} 2 & \text{if } (\alpha_j, \beta_j, \gamma_j) \in M \text{ and } k = \gamma_j - 2h, \\ 0 & \text{otherwise,} \end{cases} \quad (3.7)$$

$$j = 1, 2, \dots, g, \quad k = 1, 2, \dots, h.$$

Since each element of $\mathcal{A} \cup \mathcal{B} \cup \mathcal{R}$ appears exactly once in the 3-tuples of M , the mass balance constraints are satisfied (see Equations (2.3)–(2.5)). From Equations (3.2)–(3.7), we also see that the solution (x, y) of flow uses only arcs such that the unit shipping costs are one. Thus, the total shipping cost is obtained as

$$z(\varphi_M; x, y) = (1 + 1 + 2) \times |V_M| = 4h.$$

Contrarily, suppose that there is no solution of three-dimensional matching to the given instance of problem 3DM. We construct any possible transit permutation φ in the following manner. Let $M' \subseteq T$ be any subset of 3-tuples such that $|M'| = h$, and let $V_{M'} \subseteq V_T$ be the corresponding subset of relays to M' (it also satisfies $|V_{M'}| = h$). We assign h transits with

$r_j = 2$ to the h vertices of $V_{M'}$ one by one, while we assign $g - h$ transits with $r_j = 0$ to the remaining $g - h$ vertices of $V_T \setminus V_{M'}$ one by one. Let $\varphi_{M'}$ be a transit permutation based on such an assignment of transits. Then, a possible transit permutation φ can be written as a certain $\varphi_{M'}$, since the permutable transit vector $r = (r_1, r_2, \dots, r_g)$ contains h entries with $r_j = 2$ and $g - h$ entries with $r_j = 0$. Notice that for any subset M' , at least one element of the union $\mathcal{A} \cup \mathcal{B} \cup \mathcal{R}$ does not appear in the 3-tuples of M' . We denote by $e \in \mathcal{A} \cup \mathcal{B} \cup \mathcal{R}$ such an element. We distinguish the following two cases.

If $e \in \mathcal{A} \cup \mathcal{B}$ (as the first case), source $e \in V_1$ must send its supply $a_e = 1$ to some relays $j \in V_{M'}$ with $r_{\varphi_{M'}(j)} = 2$ by using arcs such that the unit shipping costs are $4h\ell$ since either $e \neq \alpha_j$ or $e \neq \beta_j$. Thus, for any feasible solution (x, y) of flow under the transit permutation $\varphi_{M'}$, it holds

$$\begin{aligned} z(\varphi_{M'}; x, y) &> 4h\ell \times \sum_{j=1}^g x_{ej} \\ &= 4h\ell \times a_e \quad (\text{see Equation (2.3)}) \\ &= 4h\ell. \end{aligned}$$

If $e \in \mathcal{R}$ (as the second case), sink $e' = e - 2h \in V_2$ must receive its demand $b_{e'} = 2$ from some relays $j \in V_{M'}$ with $r_{\varphi_{M'}(j)} = 2$ by using arcs such that the unit shipping costs are $2h\ell$ since $e = e' + 2h \neq \gamma_j$. Thus, also in the second case, it holds

$$\begin{aligned} z(\varphi_{M'}; x, y) &> 2h\ell \times \sum_{j=1}^g y_{je'} \\ &= 2h\ell \times b_{e'} \quad (\text{see Equation (2.5)}) \\ &= 4h\ell. \end{aligned}$$

Therefore, if there is no solution of three-dimensional matching to the given instance of problem 3DM, any optimal permutation φ^* has the optimal value such that $z^* > 4h\ell$. ■

4. A Preparatory Heuristic

In this section, we would like to propose a preparatory heuristic algorithm to problem TRANSSHIP in order to solve the problem empirically well for future.

4.1. Algorithm

The proposed heuristic consists of two stages. At the first stage, for a given instance I of problem TRANSSHIP, the heuristic relaxes the mass balance constraints at relays (see Equation (2.4)). That is, it solves the minimum cost flow problem as a relaxation of problem TRANSSHIP. The relaxation delivers a lower bound of the optimal total shipping cost z^* .

More precisely, the heuristic solves the following problem at the first stage, instead of solving problem TRANSSHIP directly.

$$\text{minimize} \quad \lambda(x, y) = \sum_{i \in V_1} \sum_{j \in V_T} p_{ij} x_{ij} + \sum_{j \in V_T} \sum_{k \in V_2} q_{jk} y_{jk} \tag{4.1}$$

$$\begin{aligned} \text{subject to} \quad & \sum_{j \in V_T} x_{ij} = a_i \quad \text{for } i \in V_1, \\ & \sum_{i \in V_1} x_{ij} = \sum_{k \in V_2} y_{jk} \quad \text{for } j \in V_T, \end{aligned} \tag{4.2}$$

$$\begin{aligned} \sum_{j \in V_T} y_{jk} &= b_k \quad \text{for } k \in V_2, \\ x_{ij} &\geq 0 \quad \text{for } i \in V_1, j \in V_T, \\ y_{jk} &\geq 0 \quad \text{for } j \in V_T, k \in V_2. \end{aligned}$$

We name the above problem MCF_RELAX. Notice that the $r_{\varphi(j)}$ in Equation (2.4) is dropped out of Equation (4.2), and problem MCF_RELAX is a special case of the minimum cost flow problem. Since the minimum cost flow problem can be solved in polynomial time (e.g., see Asano et al. [2], Orlin [11]), problem MCF_RELAX can also be solved in polynomial time.

Given an instance $I = (D, a, b, r, p, q)$ of problem TRANSSHIP, we denote by $L(I) = (D, a, b, p, q)$ the corresponding instance of problem MCF_RELAX, and by $\lambda^*(L(I))$ the minimum of a solution of flow, which must be feasible for instance $L(I)$, but does not have to be feasible for instance I (i.e., it must satisfy Equations (2.3), (2.5)–(2.7) and (4.2), but does not have to satisfy Equation (2.4)). From now on, we especially denote by $z^*(I)$ the optimal value of a given instance I of problem TRANSSHIP. It is obvious that the following lemma holds.

Lemma 1 For a given instance $I = (D, a, b, r, p, q)$ of problem TRANSSHIP and the corresponding instance $L(I) = (D, a, b, p, q)$ of problem MCF_RELAX, let $z^*(I)$ be the optimal value of instance I , and let $\lambda^*(L(I))$ be the minimum of a feasible solution of flow of instance $L(I)$. Then, it holds

$$\lambda^*(L(I)) \leq z^*(I). \tag{4.3}$$

When it has solved problem MCF_RELAX at the first stage, it has also known how much amount of flow passes through each relay in the solution (\hat{x}, \hat{y}) of flow that attains the minimum $\lambda^*(L(I))$ ($= \lambda(\hat{x}, \hat{y})$). We denote by $\hat{r} = (\hat{r}_1, \hat{r}_1, \dots, \hat{r}_n)$ the amounts of flows passing through the relays (i.e., $\hat{r}_j = \sum_{i \in V_1} \hat{x}_{ij} = \sum_{k \in V_2} \hat{y}_{jk}$ for each $j \in V_T$ in Equation (4.2)), and call the \hat{r} an *ideal transit vector*.

At the second stage, the heuristic assigns n entries of the given $r = (r_1, r_2, \dots, r_n)$ of transit vector to n relays. The assignment is based on the minimum weight perfect matching on a complete bipartite graph. Let $\mathcal{S} = \{u_s \mid s = 1, 2, \dots, n\}$ and $\mathcal{T} = \{v_t \mid t = 1, 2, \dots, n\}$ be sets of n vertices, and let $G_I = (\mathcal{S} \cup \mathcal{T}, E = \mathcal{S} \times \mathcal{T})$ be a complete bipartite graph with respect to the two sets \mathcal{S} and \mathcal{T} of vertices, where E is its set of $n \times n$ edges. Weight of each edge $(u_s, v_t) \in E$ is defined by the absolute difference between the given transit and ideal transit, i.e.,

$$w(u_s, v_t) = |r_s - \hat{r}_t|. \tag{4.4}$$

For a given instance I of problem TRANSSHIP, the proposed heuristic solves the minimum weight perfect matching problem (MWPM for short) on the complete bipartite graph G_I at the second stage. We denote by $K(I) = (G_I, w)$ the corresponding instance of problem MWPM, and by $\mu^*(K(I))$ the minimum weight of a matching $F \subset E$. The minimum weight $\mu^*(K(I))$ is going to be used later for expressing an upper bound of the performance.

Let $F^* \subset E$ be a matching that attains the minimum weight $\mu^*(K(I))$. Then, the heuristic assigns each entry r_j of the given transit vector to a relay $j' \in V_T$ such that it satisfies $(u_j, v_{j'}) \in F^*$. That is, let $\varphi_H = (\varphi_H(1), \varphi_H(2), \dots, \varphi_H(n))$ be the transit permutation

obtained by the heuristic, then it sets as $\varphi_H(j') = j$. It is well known that the minimum weight perfect matching problem on a complete bipartite graph can be solved in polynomial time (e.g., see Asano et al. [2]), and therefore the proposed heuristic theoretically runs in polynomial time.

The proposed heuristic is summarized as follows.

Algorithm HEURISTIC

Input: An instance $I = (D, a, b, r, p, q)$ of problem TRANSSHIP.

Output: A transit permutation $\varphi_H = (\varphi_H(1), \varphi_H(2), \dots, \varphi_H(n))$.

STAGE 1. Solve the corresponding instance $L(I) = (D, a, b, p, q)$ of problem MCF_RELAX, and obtain an ideal transit vector $\hat{r} = (\hat{r}_1, \hat{r}_1, \dots, \hat{r}_n)$.

STAGE 2. Perform the following three steps.

- (i) Construct the complete bipartite graph G_I and compute the weights w of edges from the given transit vector r and ideal transit vector \hat{r} (by Equation (4.4)).
- (ii) Solve the corresponding instance $K(I) = (G_I, w)$ of problem MWPM, and obtain a matching F^* that attains the minimum weight $\mu^*(K(I))$.
- (iii) Translate the matching F^* into a transit permutation φ_H , and output the φ_H .

4.2. An upper bound of the performance

We introduce the following notations

$$\Delta p = p_{\max} - p_{\min} \quad \text{and} \quad \Delta q = q_{\max} - q_{\min}, \quad (4.5)$$

where

$$\begin{aligned} p_{\max} &= \max\{p_{ij} \mid i \in V_1, j \in V_T\}, & q_{\max} &= \max\{q_{jk} \mid j \in V_T, k \in V_2\}, \\ p_{\min} &= \min\{p_{ij} \mid i \in V_1, j \in V_T\}, & q_{\min} &= \min\{q_{jk} \mid j \in V_T, k \in V_2\}. \end{aligned}$$

For a given instance $I = (D, a, b, r, p, q)$ of problem TRANSSHIP such that it satisfies $p_{ij} > 0$ and $q_{jk} > 0$ for all $i \in V_1$, $j \in V_T$ and $k \in V_2$, by using Δp , Δq and the minimum weight $\mu^*(K(I))$ of the corresponding instance of problem MWPM, we express an upper bound of the performance of the proposed heuristic (although it is not possible to be a constant factor under the hypothesis that $P \neq NP$).

Theorem 2 For a given instance $I = (D, a, b, r, p, q)$ of problem TRANSSHIP such that the unit shipping cost associated with every arc is given as a positive integer, let $z^*(I)$ (> 0) be the optimal value of instance I , and let $\lambda^*(L(I))$ (> 0) and $\mu^*(K(I))$ be the minimum of a solution of flow and the minimum weight for the corresponding instances $L(I)$ and $K(I)$ of problems MCF_RELAX and MWPM, respectively. Then, the proposed heuristic delivers a transit permutation φ_H such that it satisfies

$$\frac{\tilde{z}(\varphi_H)}{z^*(I)} \leq 1 + \frac{\Delta p + \Delta q}{2\lambda^*(L(I))} \times \mu^*(K(I)). \quad (4.6)$$

Proof. For the ideal transit vector $\hat{r} = (\hat{r}_1, \hat{r}_1, \dots, \hat{r}_n)$ and the transit permutation $\varphi_H = (\varphi_H(1), \varphi_H(2), \dots, \varphi_H(n))$ obtained by the proposed heuristic, let

$$V_T^+ = \{j \mid r_{\varphi_H(j)} - \hat{r}_j \geq 0, j \in V_T\} \quad \text{and} \quad V_T^- = \{j \mid r_{\varphi_H(j)} - \hat{r}_j < 0, j \in V_T\}$$

be subsets of relays. We call a relay $j \in V_T^+$ *hankering relay*, and a relay $j \in V_T^-$ *glutted relay*. By the feasibility condition (see Equation (2.1)) and by definition, it holds

$$\begin{aligned} & \sum_{j \in V_T^+} (r_{\varphi_H(j)} - \hat{r}_j) - \sum_{j \in V_T^-} (\hat{r}_j - r_{\varphi_H(j)}) \\ &= \sum_{j \in V_T} (r_{\varphi_H(j)} - \hat{r}_j) = \sum_{j \in V_T} r_{\varphi_H(j)} - \sum_{j \in V_T} \hat{r}_j = 0, \end{aligned}$$

and it also holds

$$\begin{aligned} & \sum_{j \in V_T^+} (r_{\varphi_H(j)} - \hat{r}_j) + \sum_{j \in V_T^-} (\hat{r}_j - r_{\varphi_H(j)}) \\ &= \sum_{j \in V_T} |r_{\varphi_H(j)} - \hat{r}_j| = \mu^*(K(I)). \end{aligned}$$

Hence, we have

$$\sum_{j \in V_T^+} (r_{\varphi_H(j)} - \hat{r}_j) = \sum_{j \in V_T^-} (\hat{r}_j - r_{\varphi_H(j)}) = \frac{\mu^*(K(I))}{2}. \quad (4.7)$$

For the transit permutation φ_H , we construct a feasible solution (x', y') of flow by modifying the solution (\hat{x}, \hat{y}) of flow that attains the minimum $\lambda^*(L(I)) (= \lambda(\hat{x}, \hat{y}))$. That is, we change the routes of *surplus flows* passing through the glutted relays $j \in V_T^-$ so that the surplus flows pass through the hankering relays $j \in V_T^+$, and the mass balance constraints at all n relays are satisfied, maintaining the mass balances at all m_1 sources and all m_2 sinks.

The amount of surplus flows is $\mu^*(K(I))/2$ (see Equation (4.7)). The shipping cost of an unit surplus flow on the new routes is at most $p_{\max} + q_{\max}$, while the shipping cost of it on the original routes is at least $p_{\min} + q_{\min}$. Hence, the change of the routes of surplus flows incurs the increase of the total shipping cost of at most

$$(p_{\max} + q_{\max}) \times \frac{\mu^*(K(I))}{2} - (p_{\min} + q_{\min}) \times \frac{\mu^*(K(I))}{2} = \frac{\Delta p + \Delta q}{2} \times \mu^*(K(I)).$$

By the minimality, it is trivial that $\tilde{z}(\varphi_H) \leq z(\varphi_H; x', y')$, and hence we have

$$\tilde{z}(\varphi_H) \leq \lambda^*(L(I)) + \frac{\Delta p + \Delta q}{2} \times \mu^*(K(I)),$$

which implies the theorem, together with Lemma 1. ■

5. Numerical Results

In this section, we generate a few kinds of test instances, and by using these instances, we examine the performance of the proposed heuristic empirically. We adopt the primal simplex method for the minimum cost flow problem (e.g., see Ibaraki and Fukushima [7]) to compute the minimum $\lambda^*(L(I))$ of problem MCF_RELAX and the minimum weight $\mu^*(K(I))$ of problem MWPM in Equation (4.6). It is well known that the primal simplex method runs fast in practice, although it does not run in polynomial time theoretically. The program is written in C, and it is compiled by Microsoft Visual C++ 2008 Express Edition. The experiment is carried out on a personal computer with Intel Pentium M CPU (1.2GHz) and 1GB memory (Panasonic CF-W4HWSAXC).

The following parameters are set to generate the test instances.

- The number of relays: $n \in \{8, 40, 80, 120, 160\}$.
- The numbers of sources and sinks: $m_1, m_2 \in [4, 16]$ for the case of $n = 8$, and $m_1, m_2 \in \{n/8, n/4, n/2, n, 2n\}$ for the case of $n \in \{40, 80, 120, 160\}$. In the whole of this experiment, these are set as $m_1 = m_2$.
- Supplies: $a_i = 24$ for the case of $n = 8$, and $a_i = 8m_1$ the case of $n \in \{40, 80, 120, 160\}$, which are set to be even for all sources $i = 1, 2, \dots, m_1$.
- Demands: $b_k = 24$ for the case of $n = 8$, and $b_k = 8m_2$ for the case of $n \in \{40, 80, 120, 160\}$, which are set to be even for all sinks $k = 1, 2, \dots, m_2$.
- Transits: r_j 's are generated by the following manner. The coefficient of non-zero transits is newly defined by

$$\eta = \frac{\text{The number of transits such that } r_j > 0 \text{ for } j = 1, 2, \dots, n}{\text{The number of all transits, which is equal to } n}.$$

The coefficient is chosen among $\eta \in \{0.25, 0.5, 1.0\}$. The non-zero transits are randomly assigned to $\eta \times n$ relays, and are set to be even, i.e., for $j = 1, 2, \dots, n$,

$$r_j \in \left\{ 0, \frac{1}{\eta \times n} \times \sum_{i=1}^{m_1} a_i \left(= \frac{1}{\eta \times n} \times \sum_{k=1}^{m_2} b_k \right) \right\}.$$

- The unit shipping costs: p_{ij} 's and q_{jk} 's are uniformly random integers in

$$[c_{\text{mean}} \times (1 - \sqrt{3}CV), c_{\text{mean}} \times (1 + \sqrt{3}CV)], \quad (5.1)$$

where the mean value is fixed as $c_{\text{mean}} = 50$ and the coefficient of variation is chosen among $CV \in \{0.1, 0.3, 0.5\}$. For example, if $CV = 0.3$, the interval is set as $[24, 76]$.

In all the tables of results, each of the data indicates the mean value for ten instances. The notations used in the tables have the following meanings.

- R_{OPT} : The ratio of the total shipping cost obtained by the proposed heuristic to the optimal value, i.e.,

$$R_{\text{OPT}} = \frac{\tilde{z}(\varphi_{\text{H}})}{z^*(I)}.$$

The optimal value $z^*(I)$ is computed by the GLPK solver [5] only for the case of $n = 8$ (which obtained the $z^*(I)$ within 120 [sec] for any test instance with $n = 8$). In order to apply the GLPK solver, problem TRANSSHIP is formulated as a mixed 0-1 integer programming problem, i.e., instead of a transit permutation φ , for $j = 1, 2, \dots, n$ and $\nu = 1, 2, \dots, n$, the following 0-1 variables $\tau_{j,\nu}$ are introduced.

$$\tau_{j,\nu} = \begin{cases} 1 & \text{if transit } r_j \text{ is assigned to relay } \nu \in V_{\text{T}}, \\ 0 & \text{otherwise.} \end{cases}$$

- R_{LB} : The ratio of the total shipping cost obtained by the proposed heuristic to a lower bound obtained by solving problem MCF_RELAX (see Equation (4.3)), i.e.,

$$R_{\text{LB}} = \frac{\tilde{z}(\varphi_{\text{H}})}{\lambda^*(L(I))}.$$

- UB: The upper bound of the performance (see Equation (4.6)), i.e.,

$$UB = 1 + \frac{\Delta p + \Delta q}{2\lambda^*(L(I))} \times \mu^*(K(I)),$$

where Δp and Δq are approximately computed as

$$\Delta p = \Delta q = c_{\text{mean}} \times 2\sqrt{3}CV$$

by using Equation (5.1).

- CPU: The execution time of CPU required to obtain a transit permutation.

Table 1: Results for test instances with $n = 8$

Sources	Relays	Sinks	$\eta = 0.25$			$\eta = 0.5$		
			R_{OPT}	R_{LB}	UB	R_{OPT}	R_{LB}	UB
m_1	n	m_2						
4	8	4	1.06	1.17	2.04	1.03	1.03	1.27
5	8	5	1.10	1.25	2.03	1.01	1.05	1.46
6	8	6	1.06	1.21	2.19	1.03	1.06	1.62
7	8	7	1.04	1.23	2.26	1.02	1.07	1.65
8	8	8	1.02	1.22	2.35	1.03	1.08	1.73
9	8	9	1.02	1.24	2.46	1.04	1.10	1.80
10	8	10	1.03	1.27	2.48	1.01	1.09	1.94
11	8	11	1.03	1.28	2.66	1.04	1.11	1.91
12	8	12	1.04	1.29	2.72	1.02	1.09	1.80
13	8	13	1.03	1.29	2.71	1.03	1.12	1.88
14	8	14	1.04	1.29	2.70	1.01	1.11	1.91
15	8	15	1.06	1.31	2.59	1.02	1.11	1.83
16	8	16	1.02	1.29	2.76	1.02	1.11	1.93
$a_i = 24$ for all i , $b_k = 24$ for all k , $c_{\text{mean}} = 50$, $CV = 0.3$								

Table 1 shows the empirical performance of the proposed heuristic for test instances with $n = 8$. The ratio R_{OPT} is no more than 1.10. Except for the case of $m_1 = m_2 = 5$ and $\eta = 0.25$, the relative error of the proposed heuristic is no more than 6%. On the other hand, we observe a wide gap between the ratio R_{OPT} and the upper bound UB of the performance, for example, $R_{\text{OPT}} = 1.02$ and $UB = 2.76$ when $m_1 = m_2 = 16$ and $\eta = 0.25$.

Table 2 shows the results of the proposed heuristic for test instances with $n \in \{40, 80, 120, 160\}$ and $\eta = 0.25$. Since $R_{\text{LB}} \geq R_{\text{OPT}}$ holds by definition, we can see that the relative error of the proposed heuristic is about 15% at the maximum when the coefficient of variation of the unit shipping costs is set as $CV = 0.1$ and $CV = 0.3$. Of course, $R_{\text{LB}} = 1.00$ does not mean $R_{\text{LB}} = 1$ (i.e., if $R_{\text{LB}} = 1$, it means that the proposed heuristic obtained the optimal values for all the ten instances tested). For the instances with $m_1 = \eta \times n = m_2$ (for example, $m_1 = m_2 = 40$, $n = 160$ and $\eta = 0.25$), the proposed heuristic tends to use $\eta \times n$ disjoint shortest paths from the $m_1 (= \eta \times n)$ sources to the $m_2 (= \eta \times n)$ sinks. As the result, the ratios R_{LB} are relatively smaller than those for other instances.

Table 3 with respect to $\eta = 0.5$ shows a similar performance of the proposed heuristic to that in the above table (i.e., Table 2) with respect to $\eta = 0.25$.

Table 4 shows the execution time of the proposed heuristic. The primal simplex method for the minimum cost flow problem which we adopt seems to perform fast in the proposed heuristic, and the heuristic solves each test instance within 1 [sec]. We would like to remind the readers that, for example, in the case of $m_1 = m_2 = 160$ and $n = 160$, the number of arcs in the network is 51,200, which is equivalent to the number of variables in a solution (x, y) of flow. The minimal solution $(\tilde{x}(\varphi_H), \tilde{y}(\varphi_H))$ of flow can also be obtained by the primal simplex method for the minimum cost flow problem as well as the heuristic transit permutation φ_H .

Table 2: Results for test instances with $n \in \{40, 80, 120, 160\}$ and $\eta = 0.25$

Sources	Relays	Sinks	$CV = 0.1$	$CV = 0.3$	$CV = 0.5$
m_1	n	m_2	R_{LB}	R_{LB}	R_{LB}
5	40	5	1.03	1.15	1.56
10	40	10	1.00	1.01	1.04
20	40	20	1.02	1.09	1.38
40	40	40	1.02	1.10	1.44
80	40	80	1.02	1.11	1.49
10	80	10	1.02	1.09	1.42
20	80	20	1.00	1.01	1.05
40	80	40	1.01	1.05	1.25
80	80	80	1.01	1.05	1.27
160	80	160	1.01	1.06	1.31
15	120	15	1.01	1.07	1.36
30	120	30	1.00	1.00	1.04
60	120	60	1.01	1.03	1.19
120	120	120	1.00	1.03	1.20
240	120	240	1.01	1.04	1.22
20	160	20	1.01	1.06	1.29
40	160	40	1.00	1.00	1.03
80	160	80	1.00	1.03	1.15
160	160	160	1.00	1.03	1.15
320	160	320	1.00	1.03	1.17
$a_i = 8m_1$ for all i , $b_k = 8m_2$ for all k , $\eta = 0.25$, $c_{\text{mean}} = 50$					

Table 3: Results for test instances with $n \in \{40, 80, 120, 160\}$ and $\eta = 0.5$

Sources	Relays	Sinks	$CV = 0.1$	$CV = 0.3$	$CV = 0.5$
m_1	n	m_2	R_{LB}	R_{LB}	R_{LB}
5	40	5	1.05	1.21	1.79
10	40	10	1.02	1.10	1.41
20	40	20	1.00	1.02	1.08
40	40	40	1.01	1.04	1.16
80	40	80	1.01	1.03	1.15
10	80	10	1.03	1.13	1.58
20	80	20	1.01	1.05	1.25
40	80	40	1.00	1.01	1.04
80	80	80	1.00	1.02	1.10
160	80	160	1.00	1.02	1.10
15	120	15	1.02	1.09	1.45
30	120	30	1.01	1.04	1.19
60	120	60	1.00	1.01	1.04
120	120	120	1.00	1.01	1.07
240	120	240	1.00	1.01	1.07
20	160	20	1.01	1.07	1.35
40	160	40	1.00	1.03	1.15
80	160	80	1.00	1.00	1.03
160	160	160	1.00	1.01	1.05
320	160	320	1.00	1.01	1.05
$a_i = 8m_1$ for all i , $b_k = 8m_2$ for all k , $\eta = 0.5$, $c_{\text{mean}} = 50$					

Table 4: Execution time of the proposed heuristic

Sources	Relays	Sinks	CPU
m_1	n	m_2	[msec]
20	40	20	25
40	40	40	42
40	80	40	92
80	80	80	166
60	120	60	207
120	120	120	368
80	160	80	379
160	160	160	725
$a_i = 8m_1$ for all i , $b_k = 8m_2$ for all k			
$\eta = 1.0$, $CV = 0.3$			

6. Concluding Remarks

In this paper, we considered the hardness of approximating a transshipment problem with a permutable transit vector. The problem, which we call TRANSSHIP, is a generalization of the transportation problem with a permutable demand vector. We have already known that problem TRANSSHIP is NP-hard even if each entry of the permutable transit vector takes either zero or two. In this paper, we proved that even if the unit shipping cost associated with every arc is given as a positive integer, there is no polynomial time constant factor approximation algorithm for problem TRANSSHIP under the hypothesis that $P \neq NP$. We obtained the inapproximability result by a gap-introducing reduction from an NP-complete problem by the name of 3DM. Also, we proposed a heuristic algorithm and showed an upper bound of the performance.

The following directions are open for future research. It would be interesting to design a practical exact algorithm such as a branch-and-bound. Also, it would be interesting to examine whether some significant class related to problem TRANSSHIP admits a constant factor or a data-dependent (e.g., logarithmic) factor approximation.

Acknowledgement

We would like to express our gratitude to the anonymous referees whose constructive suggestions contributed to improving the readability of Section 5. We gave a presentation of a part of this paper at the 5th International Conference on Leading Edge Manufacturing in 21st Century (LEM21), JSME, December 2009, Osaka, Japan. This research was partially supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science (JSPS).

References

- [1] T. Asano: *Approximation Algorithms* (Springer-Verlag Tokyo, 2002) (in Japanese). (Translation from the English language edition by V.V. Vazirani: Springer-Verlag, Berlin, 2001).
- [2] T. Asano, T. Hirata, T. Ono and Y. Asano: *Combinatorial Optimization* (Springer-Verlag Tokyo, 2005) (in Japanese). (Translation from the English language edition by B. Korte and J. Vygen: Springer-Verlag, Berlin, 2000, 2002, 2005).
- [3] J. Edmonds and R.M. Karp: Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, **19** (1972), 248–264.

- [4] M.R. Garey and D.S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W.H. Freeman, San Francisco, 1979).
- [5] GLPK (GNU Linear Programming Kit): <http://www.gnu.org/software/glpk/> [last accessed March 2010].
- [6] M. Hujter, B. Klinz and G.J. Woeginger: A note on the complexity of the transportation problem with a permutable demand vector. *Mathematical Methods of Operations Research*, **50** (1999), 9–16.
- [7] T. Ibaraki and M. Fukushima: *FORTRAN77 Optimization Programming, The Iwanami Computer Science Series* (Iwanami Shoten Publishers, Tokyo, 1991) (in Japanese).
- [8] Y. Karuno, T. Tachibana and K. Yamashita: A transshipment problem with a permutable transit vector. *Proceedings of JSME/SSJ International Symposium on Scheduling 2009*, (2009), 163–169.
- [9] N. Katoh: *Introduction to Mathematical Programming* (Corona Publishing Co., Ltd., Tokyo, 2008) (in Japanese).
- [10] S.G. Meusel and R.E. Burkard: A transportation problem with a permuted demand vector. *Mathematical Methods of Operations Research*, **50** (1999), 1–7.
- [11] J.B. Orlin: A faster strongly polynomial minimum cost flow algorithm. *Operations Research*, **41** (1993), 338–350.
- [12] N. Tomizawa: On some techniques useful for solution of transportation network problems. *Networks*, **1** (1971), 173–194.

Yoshiyuki Karuno

Department of Mechanical and System Engineering

Kyoto Institute of Technology

Matsugasaki, Sakyo, Kyoto 606-8585, Japan

E-mail: karuno@kit.ac.jp