

## ON NETWORK SIMPLEX METHOD USING THE PRIMAL-DUAL SYMMETRIC PIVOTING RULE

Masakazu Muramatsu  
*Sophia University*

(Received December 22, 1998; Revised April 12, 1999)

*Abstract* We consider a network simplex method using the primal-dual symmetric pivoting rule proposed by Chen, Pardalos, and Saunders. For minimum-cost network-flow problems, we prove global convergence of the algorithm and propose a new scheme in which the algorithm can start from an arbitrary pair of primal and dual feasible spanning trees. For shortest-path problems, we prove the strongly polynomial time complexity of the algorithm.

### 1. Introduction

Let  $G = (N, \mathcal{A})$  be a directed graph with  $n$  nodes and  $m$  arcs where  $N$  is the set of nodes and  $\mathcal{A}$  is the set of arcs. We consider the uncapacitated minimum-cost network-flow problem on  $G$ :

$$\langle P \rangle \begin{cases} \text{minimize} & \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} \\ \text{subject to} & \sum_{l:(i,l) \in \mathcal{A}} x_{il} - \sum_{l:(l,i) \in \mathcal{A}} x_{li} = b_i, \quad i \in N \\ & x_{ij} \geq 0 \quad (i,j) \in \mathcal{A}. \end{cases} \quad (1.1)$$

It is well known that a capacitated minimum-cost network-flow problem can easily be transformed into an uncapacitated one (for example, see page 40 of the textbook [1] by Ahuja, Magnanti, and Orlin). Therefore, we do not lose generality by using this form. The dual of (1.1) is the following:

$$\langle D \rangle \begin{cases} \text{maximize} & \sum_{i \in N} b_i y_i \\ \text{subject to} & z_{ij} = c_{ij} - y_i + y_j, \quad (i,j) \in \mathcal{A} \\ & z_{ij} \geq 0, \quad (i,j) \in \mathcal{A}. \end{cases} \quad (1.2)$$

In this paper, we apply the primal-dual symmetric pivoting rule proposed by Chen, Pardalos, and Saunders [5] to  $\langle P \rangle$  and  $\langle D \rangle$ , developing a new network simplex method. We show the convergence of the algorithm, relax the assumption for initial solutions, and prove the strongly polynomial-time convergence of the algorithm when it is applied to the shortest-path problems.

Since  $\langle P \rangle$  and  $\langle D \rangle$  can be viewed as linear programming problems, we can apply the well-known simplex method for linear programming to them. Using the nice relationship between spanning tree and basic solution which will be described later, we can reduce the calculation cost of pivoting drastically for minimum-cost network-flow problems. This idea has its origin in Dantzig [7] and Johnson [11]. An implementation of the simplex method using this technique is called *network simplex method*.

There is another famous alternative solution for linear programming: interior-point method. Recent research reveals that the primal-dual interior-point methods which deal with the primal problem and its dual simultaneously are the most powerful and efficient among other interior-point methods. The primal-dual interior-point methods have the nice *primal-dual symmetric property* which means that, even if we view the dual as a primal problem to apply the algorithm, the generated sequence is identical to the original one. See the textbook by Wright [14] for details of the primal-dual interior-point methods. Few simplex methods have the primal-dual symmetric property. Neither the primal simplex method nor the dual has this property. Even the so-called primal-dual simplex method does not. The simplex method recently proposed by Chen, Pardalos, and Saunders [5], is one of the methods having the primal-dual symmetric property. We call their pivoting rule the *primal-dual symmetric pivoting rule* in this paper.

We will show global convergence of the algorithm applied to minimum-cost network-flow problems without nondegeneracy assumption. Specifically, the number of pivots performed by the network simplex method using the primal-dual symmetric pivoting rule is bounded by a pseudopolynomial of the data size. For general linear programming, global convergence of this algorithm is proved under nondegeneracy assumption in the original paper [5]. A program instance for which their algorithm takes exponential time of iterations is also found in Darios and Paparrizos [9]. Our result for minimum-cost network-flow problem is better than these results observed in general linear programming case, but worse than the best bound of the network simplex methods for minimum-cost network-flow problems which is strongly polynomial of the data size. For polynomial network simplex methods, see Orlin, Plotkin and Tardos [12], Armstrong and Jin [3], and Orlin [13]. However, we point out that even for the Dantzig's pivoting rule which is widely used in implementation, only global convergence has been proved.

Another topic of this paper is initialization. The original algorithm proposed in [5] needs a pair of basic solutions having certain special property. The method proposed in [5] to derive such a pair for general linear programming problems introduces large artificial numbers to the coefficient matrix. If the same method is applied to the network problems, then the network structure of the problem will be destroyed. Therefore, we cannot use it in the framework of the network simplex method. In this paper, we give a solution to this problem. Using the basic primal-dual symmetric pivoting rule, we construct a new algorithm which can start from arbitrary spanning trees one of which is feasible for  $\langle P \rangle$  and the other for  $\langle D \rangle$ .

We also apply our algorithm to the shortest-path problems. Our algorithm requires only  $n(n-1)/2$  primal pivots (Theorem 11). This bound is much better than  $O(n^2 \log(nC))$  the bound for the number of pivots performed by Dantzig's rule where  $C$  is the maximum absolute value of the cost (See page 429 of [1]), and comparable to  $(n-1)(n-2)/2$  the current best bound of network simplex method for the shortest-path problem due to Goldfarb, Hao, and Kai [10]. We have more on the comparison of their algorithm and ours in Section 5.

The paper is organized as follows. In the rest of this section, we define some fundamental concepts for the network simplex method. In Section 2, we describe the primal-dual symmetric pivoting rule in the context of network simplex method. In Section 3, we prove global convergence of the algorithm. This algorithm is just a network simplex version of what is proposed by Chen, Pardalos, and Saunders [5] and needs a pair of spanning trees one of which is feasible for  $\langle P \rangle$  and the other for  $\langle D \rangle$  and which share all arcs but one. In Section 4, we discuss initialization issue and propose a new algorithm that can start from arbitrary pair of primal feasible and dual feasible spanning trees. In Section 5, we apply

the algorithm to the shortest-path problem which is a special case of the minimum-cost network-flow problem and prove the strongly polynomial time complexity of the algorithm.

Throughout the paper, we assume:

- $G$  is connected.
- all data are integral.
- $\sum_{i \in N} b_i = 0$ .
- $\langle P \rangle$  and  $\langle D \rangle$  are feasible.

Given  $\mathcal{T}$ , a spanning tree of  $G$ , we define the primal basic solution:

**Definition 1 Primal Basic Solution**  $x(\mathcal{T})$ : Set  $x_{ij} = 0$  if  $(i, j) \notin \mathcal{T}$ , and otherwise so that  $x_{ij}$  satisfies the equality conditions of (1.1).

The primal basic solution is well-defined. In fact, to decide  $x_{ij}$  on the tree  $\mathcal{T}$ , we choose a leaf  $(i, j)$  where the node  $i$  or  $j$  does not share any arc of  $\mathcal{T}$  other than  $(i, j)$ . In case where  $j$  does not share, we put  $x_{ij} = b_j$  removing  $(i, j)$  from  $\mathcal{T}$  and changing  $b_i$  to  $b_i - b_j$ . This procedure determines  $x_{ij}$  and decreases  $|\mathcal{T}|$  by 1. It is easy to see that performing this procedure recursively, we have  $x_{ij} : (i, j) \in \mathcal{A}$  satisfying the equality condition of  $\langle P \rangle$ . See [1] for more detail.

Similarly, we define the dual basic solution as follows.

**Definition 2 Dual Basic Solution**  $(y(\mathcal{T}), z(\mathcal{T}))$ : Set  $z_{ij} = 0$  if  $(i, j) \in \mathcal{T}$  and otherwise so that the equality conditions of (1.2) hold.

We sometimes call  $y(\mathcal{T})$  a node potential vector and  $z(\mathcal{T})$  a dual slack corresponding to  $\mathcal{T}$ , respectively. To be precise, the node potential vector will not be determined uniquely by this definition; we can add a constant to all  $y_i$  without changing  $z_{ij}$ , thus satisfying the definition. Without loss of generality, we can assume that there is a node for which the node potential is always zero. Once we fix such a node, we can calculate  $y_i : i \in N$  by visiting each node through  $\mathcal{T}$  from that node using the relation  $c_{ij} - y_i + y_j = 0$  for  $(i, j) \in \mathcal{T}$ . Then the values of  $z_{ij}$  for  $(i, j) \notin \mathcal{T}$  are automatically determined by  $c_{ij} - y_i + y_j$ .

We call that a spanning tree  $\mathcal{T}$  of  $G$  is primal (or dual) feasible for  $\langle P \rangle$  (or  $\langle D \rangle$ ), if the corresponding primal (or dual) basic solution is feasible for  $\langle P \rangle$  (or  $\langle D \rangle$ ), respectively. By the duality theorem, if  $\mathcal{T}$  is primal and dual feasible, then  $x(\mathcal{T})$  and  $(y(\mathcal{T}), z(\mathcal{T}))$  are optimal solutions for  $\langle P \rangle$  and  $\langle D \rangle$ , respectively, in which case we say that  $\mathcal{T}$  is an optimal spanning tree.

## 2. The Primal-Dual Symmetric Pivoting Rule

Let  $\mathcal{T}_P$  and  $\mathcal{T}_D$  be primal and dual feasible spanning trees, respectively, and  $x^P$  and  $(y^D, z^D)$  be the corresponding primal and dual basic solutions. Assume also that  $\mathcal{T}_P$  is different from  $\mathcal{T}_D$  by only one arc, that is,

$$\mathcal{T}_D = \mathcal{T}_P - \{(u, v)\} + \{(r, s)\}, \quad (2.1)$$

where  $(u, v)$  and  $(r, s)$  are the difference arcs. By definition, we have

$$\begin{aligned} x_{ij}^P &= 0 \text{ if } (i, j) \notin \mathcal{T}_P, \\ x_{ij}^P &\geq 0 \text{ if } (i, j) \in \mathcal{T}_P, \end{aligned}$$

and

$$\begin{aligned} z_{ij}^D &\geq 0 \text{ if } (i, j) \notin \mathcal{T}_D, \\ z_{ij}^D &= 0 \text{ if } (i, j) \in \mathcal{T}_D, \end{aligned}$$

thus

$$\sum_{(i,j) \in \mathcal{A}} x_{ij}^P z_{ij}^D = \sum_{(i,j) \in \mathcal{T}_P} x_{ij}^P z_{ij}^D = x_{uv}^P z_{uv}^D. \quad (2.2)$$

This relation means that if  $x_{uv}^P = 0$  or  $z_{uv}^D = 0$ , then  $x^P$  and  $z^D$  are optimal for  $\langle P \rangle$  and  $\langle D \rangle$ , respectively.

Let

$$U = \left\{ i \in N \mid \begin{array}{l} \text{there exists an undirected path} \\ \text{between } i \text{ and } u \text{ on } \mathcal{T}_P - \{(u, v)\}. \end{array} \right\}, \quad (2.3)$$

$$V = N - U, \quad (2.4)$$

and  $\mathcal{T}_P/U = \{(i, j) \in \mathcal{T}_P \mid (i, j) \in U \times U\}$ . Since there is no cycle on  $\mathcal{T}_P/U$  and  $\mathcal{T}_P/U$  connects all the nodes in  $U$  to  $u$ ,  $\mathcal{T}_P/U$  is a spanning tree of  $U$ . Similarly, we can see that  $\mathcal{T}_P/V$  is a spanning tree of  $V$ . We also have the following lemma.

**Lemma 1** 1.  $\mathcal{T}_D/U = \mathcal{T}_P/U$  and  $\mathcal{T}_D/V = \mathcal{T}_P/V$ .

2. Either  $(r, s) \in U \times V$  or  $(r, s) \in V \times U$ .

*Proof*: Since  $\mathcal{T}_P/U$  does not contain  $(u, v)$ ,  $\mathcal{T}_P/U$  is included by  $\mathcal{T}_D$ , from which it follows that  $\mathcal{T}_D/U = \mathcal{T}_P/U$ . The relation  $\mathcal{T}_D/V = \mathcal{T}_P/V$  can be shown similarly. Now  $(r, s)$  must connect  $U$  and  $V$  since if it does not, then there is no path from  $U$  to  $V$  on  $\mathcal{T}_D$ , which contradicts that  $\mathcal{T}_D$  is a spanning tree.  $\square$

If  $(r, s) \in U \times V$ , we call that  $(r, s)$  has the *same* direction as  $(u, v)$  with respect to the partition  $(U, V)$  and *opposite* in the other case.

If  $(r, s)$  has the same direction as  $(u, v)$ , then we let  $\mathcal{W}$  be the cycle which consists of  $(r, s)$  and  $\mathcal{T}_P$ . We define the orientation of  $\mathcal{W}$  so that  $\mathcal{W}$  contains  $(r, s)$  as a forward-arc. Let

$$\delta_P = \min\{x_{ij}^P \mid (i, j) \in \mathcal{W} \text{ is a backward-arc of } \mathcal{W}\} \geq 0, \quad (2.5)$$

and send  $\delta_P$  on  $\mathcal{W}$  in its orientation. Since  $(u, v)$  is a backward-arc of  $\mathcal{W}$ , the constraint set of (2.5) is nonempty. As a result, at least one backward-arc, say  $(p, q)$ , of  $\mathcal{W}$  will have a flow 0, and  $x_{uv}$  will be reduced by  $\delta_P$ . The resulting  $x$  is still feasible for  $\langle P \rangle$ , where the corresponding spanning tree is  $\mathcal{T}_P - \{(r, s)\} + \{(p, q)\}$ . This procedure is called a *primal* pivot, and  $(r, s)$  and  $(p, q)$  are called entering and leaving arcs, respectively.

If  $(r, s)$  has the opposite direction as  $(u, v)$  with respect to  $(U, V)$ , then adding

$$\delta_D = \min\{z_{ij}^D \mid (i, j) \in U \times V\} \geq 0$$

to  $y_i$  for  $i \in U$ , we get a new dual feasible solution. In fact, after  $\delta_D$  is added to  $y_i$  for  $i \in U$ ,

- $\forall (i, j) \in \{(i', j') \in \mathcal{A} \mid (i', j') \in U \times U \text{ or } (i', j') \in V \times V\}$ ,  $z_{ij}^D$  is not changed.
- $\forall (i, j) \in \{(i', j') \in \mathcal{A} \mid (i', j') \in U \times V\}$ ,  $z_{ij}^D$  decreases by  $\delta_D$  and for at least one arc  $(p, q) \in U \times V$ ,  $z_{pq}^D$  will become 0.
- $\forall (i, j) \in \{(i', j') \in \mathcal{A} \mid (i', j') \in V \times U\}$ ,  $z_{ij}^D$  increases by  $\delta_D$ , thus remains nonnegative.

This procedure is called a *dual* pivot, and  $(r, s)$  and  $(p, q)$  are called leaving arc and entering arc, respectively. Note that since  $(u, v) \in U \times V$ ,  $z_{uv}$  is decreased by  $\delta_D$ .

**Lemma 2** 1. In case of primal pivot, the resulting tree  $\mathcal{T}_P^+$  is different from  $\mathcal{T}_D$  by at most one arc. If  $(u, v)$  leaves out, then  $\mathcal{T}_P^+ = \mathcal{T}_D$ , in which case  $\mathcal{T}_P^+$  is an optimal spanning tree. If  $(u, v)$  does not leave out, then the partition  $(U, V)$  will be changed to  $(U^+, V^+)$  and the new  $(r^+, s^+) \in \mathcal{T}_D \setminus \mathcal{T}_P^+$  has the opposite direction as  $(u, v)$ .

2. In case of dual pivot, the resulting tree  $\mathcal{T}_D^+$  is different from  $\mathcal{T}_P$  by at most one arc. If  $(u, v)$  enters in, then  $\mathcal{T}_P = \mathcal{T}_D^+$ , in which case  $\mathcal{T}_D^+$  is an optimal spanning tree. If  $(u, v)$  does not enter in, the partition  $(U, V)$  does not change due to that dual pivot, and the new  $(r^+, s^+) \in \mathcal{T}_D^+ \setminus \mathcal{T}_P$  has the same direction as  $(u, v)$ .

*Proof*: Let  $(p, q)$  be the leaving arc from  $\mathcal{T}_P$  in a primal pivot. The new spanning tree is

$$\mathcal{T}_P + \{(r, s)\} - \{(p, q)\} = \mathcal{T}_D + \{(u, v)\} - \{(p, q)\}, \quad (\text{Use (2.1)})$$

thus the difference is at most one, and if  $(p, q) = (u, v)$ , then  $\mathcal{T}_P^+ = \mathcal{T}_D$ . If  $(p, q) \neq (u, v)$ , then the new  $(r^+, s^+)$  is  $(p, q)$ . Since  $(p, q)$  and  $(u, v)$  are backward-arcs of  $\mathcal{W}$ , there is an undirected path from  $q$  to  $u$  in  $\mathcal{T}_P^+$ . This implies that  $s^+ \in U^+$  and by Lemma 1,  $(r^+, s^+)$  has the opposite direction as  $(u, v)$  with respect to  $(U^+, V^+)$ . This proves the first assertion.

The first and second parts of the second assertion follow by a similar argument as above. We omit the detail. The last part is obvious since the definition of  $U$  depends only on  $(u, v)$  and  $\mathcal{T}_P$ .  $\square$

This is the application of the primal-dual symmetric pivoting rule proposed in [5] to the minimum-cost network-flow problem. The whole algorithm is described in the following.

**procedure** *primal-dual-pivot* $((x^P, \mathcal{T}_P), (y^D, z^D, \mathcal{T}_D))$

**begin**

Set  $(u, v)$  and  $(r, s)$  such that (2.1) holds.

Calculate  $U$  and  $V$ .

**while**  $x_{uv}^P z_{uv}^D > 0$  **do**

**begin**

**if**  $r \in U$  and  $s \in V$  **then**

*primal\_pivot* $(x^P, \mathcal{T}_P, (r, s))$ ;

**else**

*dual\_pivot* $(y^D, z^D, \mathcal{T}_D, (r, s))$ ;

**endif**

Update  $U$  and  $V$ .

**end**

**end**

**procedure** *primal-pivot* $(x^P, \mathcal{T}_P, (r, s))$

**begin**

Let  $\mathcal{W}$  be the cycle which consists of  $(r, s)$  and  $\mathcal{T}_P$ ;

$\delta_P := \min\{x_{ij}^P \mid (i, j) \text{ is a backward-arc of } \mathcal{W}\}$

Let  $(p, q)$  be a minimum achieving arc;

**for**  $(i, j) \in \mathcal{W}$  **do**

**if**  $(i, j)$  is a forwarding-arc of  $\mathcal{W}$  **then**

$x_{ij}^P := x_{ij}^P + \delta_P$ ;

**else**

$x_{ij}^P := x_{ij}^P - \delta_P$ ;

**endif**

$\mathcal{T}_P := \mathcal{T}_P - \{(p, q)\} + \{(r, s)\}$ ;

**end**

**procedure** *dual-pivot*( $y^D, z^D, \mathcal{T}_D, (r, s)$ )

**begin**

Let  $(U, V)$  be the partition of  $N$  defined by  $(r, s)$  and  $\mathcal{T}_D$ .

$\mathcal{C} := \{(i, j) \in \mathcal{A} \mid i \in U, j \in V\}$ ;

$\delta_D := \min\{z_{ij}^D \mid (i, j) \in \mathcal{C}\}$  and  $(p, q)$  be a minimum achieving arc;

**for**  $i \in U$  **do**  $y_i^D := y_i^D + \delta_D$ ;

**for**  $(i, j) \in \mathcal{A}$  **do**  $z_{ij}^D := c_{ij} - y_i^D + y_j^D$ ;

$\mathcal{T}_D := \mathcal{T}_D - \{(r, s)\} + \{(p, q)\}$ ;

**end**

By Lemma 2, we immediately obtain the following properties on the procedure *primal-dual-pivot*.

**Corollary 3** 1. *The primal and dual pivots are executed alternately.*

2. *The difference arc  $(u, v)$  does not change through the iterations.*

### 3. Convergence Analysis

In this section, we prove global convergence of the procedure *primal-dual-pivot* described in Section 2. Note that a pivot does not necessarily decrease the objective function value. Such a pivot is called a *degenerate pivot*. If we encounter a degenerate pivot, then we have a possibility that the algorithm fails to find an optimal solution. To avoid this situation, we use a notion of strongly feasible spanning tree developed by Cunningham [6] and Barr, Glover, and Klingman [4].

**Definition 3 Strongly Feasible Spanning Tree** *A primal feasible spanning tree is called strongly feasible if there is a node  $t$  such that any arc of the spanning tree whose flow is zero has the direction towards  $t$ .*

We call  $t$  the *root* of the primal strongly feasible spanning tree.

Suppose that we are making a primal pivot. If there is only one edge achieving the minimum of (2.5), then that edge is the leaving arc. In this case, the resulting spanning tree is also strongly feasible with root  $t$ .

In the case where we have several candidates for leaving arc, we need more involved argument to maintain strong feasibility of the spanning tree. We call such candidate arcs *blocking arcs*. The *apex* of the cycle  $\mathcal{W}$  is the vertex  $w$  satisfying that:

$$w = \operatorname{argmin}_{w' \in \mathcal{W}} \{\text{length of the undirected path along } \mathcal{T}_P \text{ connecting } w' \text{ and } t.\}$$

We will use the following rule to choose the leaving arc:

**Leaving Arc Selection Rule:** The leaving arc is the last blocking arc encountered in traversing the cycle  $\mathcal{W}$  along its orientation starting from  $w$ .

The following lemma is well-known, and we show it without proof. See, for example, page 421 of [1] for a proof.

**Lemma 4** *If we use the Leaving Arc Selection Rule described above, then the spanning tree after the primal pivot is strongly feasible with root  $t$ .*

We assume that the initial spanning tree is strongly feasible for root  $t$  in this section. Lemma 4 implies that  $\mathcal{T}_P$  is always strongly feasible through iterations.

Now we check the number of primal pivots. We remark that the same number of dual pivots are needed in *primal-dual-pivot*.

**Lemma 5** *There are no  $2n$  consecutive degenerate primal pivots.*

*Proof*: Since we consider only primal pivots, we assume that  $(r, s) \in U \times V$  in this proof. We also assume that the leaving arc is not  $(u, v)$ .

First, we consider the case where a degenerate pivot occurs when  $t \in U$ . Since the distance from  $t$  to  $u$  is no less than that to the apex  $w$ ,  $w$  is contained in  $U$ . We claim that  $w$  is different from  $r$ . In fact, if  $w = r$ , we can send a positive amount of flow from node  $s$  to  $w$  along  $\bar{\mathcal{T}}_P$ , thus on  $\mathcal{W}$  as well. This means that the pivot is not degenerate, which is a contradiction.

Since we can send a positive amount of flow from  $s$  to  $w$ , the leaving arc must be contained between  $w$  and  $r$ . This implies that  $t$  remains in  $U$  after the pivoting and that  $|U|$  is decreased at least by 1. Therefore, it is impossible to have  $n - 1$  consecutive degenerate primal pivots in this case.

Next we consider the case where  $t \in V$ . In this case,  $w$  is contained in  $V$ . Since we can send a positive amount of flow from  $s$  to  $t$  through  $w$ , no arc between  $s$  and  $w$  will leave and the leaving arc must be between  $v$  and  $w$ , or  $r$  and  $u$ . (Recall that we ignore the case  $(u, v)$  is leaving, in which case we obtain an optimal spanning tree.)

If the leaving arc is between  $v$  and  $w$ , then  $t$  will be contained in  $U$  after the pivot, and it is impossible to have  $n - 1$  more consecutive degenerate primal pivots due to the previous argument.

If the leaving arc is between  $r$  and  $u$ , then the same argument as above shows that  $t$  remains in  $V$  and  $|U|$  is decreased at least by 1. Therefore, within  $n - 1$  consecutive degenerate primal pivots,  $U$  have no arcs in it, and in the next degenerate primal pivot an arc between  $v$  and  $t$  leaves. After this pivot,  $t$  is contained in  $U$ , and there are no  $n - 1$  consecutive degenerate primal pivots. This is the longest case where the total number of consecutive degenerate pivots is still less than  $2n$ .  $\square$

Now we prove the convergence of our algorithm.

**Theorem 6** *The algorithm terminates in at most  $2n^2 \max_{i \in N} b_i$  primal pivots.*

*Proof*: Due to Lemma 5, a nondegenerate primal pivot occurs within  $2n$  primal pivots, reducing  $x_{uv}^P$  at least by 1. Therefore, the number of primal pivot is bounded by  $2n\hat{x}_{uv}^P$  where  $\hat{x}^P$  is the initial flow. The initial flow is bounded by

$$\sum_{i \in N} |b_i|/2 = \sum_{i \in N: b_i > 0} b_i \leq n \max_{i \in N} b_i.$$

$\square$

#### 4. Initialization Issue

It is not easy in general to find a pair of primal strongly feasible and dual feasible spanning trees which share all arcs but one. The initialization scheme shown in [5] introduces a large artificial number into the coefficient matrix, and cannot be used in the framework of network simplex method. We resolve this issue in this section. Specifically, we assume that we have a primal strongly feasible spanning tree  $\bar{\mathcal{T}}_P$  and a dual feasible spanning tree  $\bar{\mathcal{T}}_D$ , and that  $\bar{\mathcal{T}}_P$  is different from  $\bar{\mathcal{T}}_D$  by more than one arc. This assumption is relatively mild since one can easily obtain a strongly feasible spanning tree and a dual feasible spanning tree (See [1]). Under this condition, we show how to obtain an optimal solution of  $\langle P \rangle$  and  $\langle D \rangle$  by using the procedure *primal-dual-pivot*.

Let

$$\mathcal{R} = \left\{ (i, j) \in \bar{\mathcal{T}}_P \mid (i, j) \notin \bar{\mathcal{T}}_D \right\}.$$

By assumption, we have  $R := |\mathcal{R}| > 1$ . For  $k \in \{0, \dots, R\}$ , we define an edge set  $\mathcal{R}_k$  as follows:

- $\mathcal{R}_0 = \mathcal{R}$ .
- $\mathcal{R}_{k+1}$  is obtained by removing an (arbitrary) edge from  $\mathcal{R}_k$ .

It is obvious that  $|\mathcal{R}_{k+1}| = |\mathcal{R}_k| - 1$  and that  $\mathcal{R}_R = \phi$ .

For  $k \in \{0, \dots, R\}$ , we consider the following primal-dual pair of the problems:

$$\langle \bar{P}_k \rangle \begin{cases} \text{minimize} & \sum_{(i,j) \in \mathcal{R}_k} \bar{c}_{ij} x_{ij} + \sum_{(i,j) \in \mathcal{A} - \mathcal{R}_k} c_{ij} x_{ij} \\ \text{subject to} & \sum_{l: (i,l) \in \mathcal{A}} x_{il} - \sum_{l: (l,i) \in \mathcal{A}} x_{li} = b_i, \quad i \in N \\ & x_{ij} \geq 0, \quad (i,j) \in \mathcal{A}. \end{cases} \quad (4.1)$$

$$\langle \bar{D}_k \rangle \begin{cases} \text{maximize} & \sum_{i \in N} b_i y_i \\ \text{subject to} & z_{ij} = \bar{c}_{ij} - y_i + y_j, \quad (i,j) \in \mathcal{R}_k \\ & z_{ij} = c_{ij} - y_i + y_j, \quad (i,j) \in \mathcal{A} - \mathcal{R}_k \\ & z_{ij} \geq 0, \quad (i,j) \in \mathcal{A}, \end{cases} \quad (4.2)$$

where  $\bar{c}_{ij}$  for  $(i,j) \in \mathcal{R}$  will be defined later. We denote the optimal spanning tree of  $\langle \bar{P}_k \rangle$  and  $\langle \bar{D}_k \rangle$  by  $\mathcal{T}^k$ . Since  $\langle \bar{P}_R \rangle$  is  $\langle P \rangle$  and  $\langle \bar{D}_R \rangle$  is  $\langle D \rangle$ ,  $\mathcal{T}^R$  is optimal for  $\langle P \rangle$  and  $\langle D \rangle$ .

Before going into the details of the algorithm, we give an overview. We will define  $\bar{c}_{ij}$  for  $(i,j) \in \mathcal{R}$  so that  $\bar{c}_{ij} \leq c_{ij}$  holds and that  $\bar{\mathcal{T}}_P$  is an optimal spanning tree of  $\langle \bar{P}_0 \rangle$  and  $\langle \bar{D}_0 \rangle$ . Observe that, for  $k \in \{0, \dots, R\}$ , the feasible region of  $\langle \bar{P}_k \rangle$  is identical to that of  $\langle P \rangle$ . Therefore,  $\mathcal{T}^k$ , the optimal spanning tree of  $\langle \bar{P}_k \rangle$ , is feasible for  $\langle \bar{P}_{k+1} \rangle$ . Then we will show that a dual feasible spanning tree  $\mathcal{T}_D^{k+1}$  for  $\langle \bar{D}_{k+1} \rangle$  can be obtained by pivoting one arc from  $\mathcal{T}^k$ . We can start the procedure *primal-dual-pivot* using  $\mathcal{T}^k$  and  $\mathcal{T}_D^{k+1}$  to obtain an optimal spanning tree  $\mathcal{T}^{k+1}$  of  $\langle \bar{P}_{k+1} \rangle$ . Starting with the pair  $\langle \bar{P}_0 \rangle$  and  $\langle \bar{D}_0 \rangle$ , and their obvious optimal spanning tree, we obtain an optimal spanning tree of  $\langle \bar{P}_R \rangle = \langle P \rangle$  and  $\langle \bar{D}_R \rangle = \langle D \rangle$  after at most  $R$  executions of *primal-dual-pivot*. This is the whole sketch of the algorithm.

To define  $\bar{c}_{ij}$  for  $(i,j) \in \mathcal{R}$ , we let  $(\bar{y}, \bar{z})$  be the dual basic solution for  $\langle D \rangle$  corresponding to the initial dual feasible spanning tree  $\bar{\mathcal{T}}_D$ . We define

$$\bar{c}_{ij} = \bar{y}_i - \bar{y}_j$$

for  $(i,j) \in \mathcal{R}$ .

**Lemma 7** *We have*

1.  $c_{ij} \geq \bar{c}_{ij}$ .
2.  $\bar{\mathcal{T}}_P$  is optimal for  $\langle \bar{P}_0 \rangle$  and  $\langle \bar{D}_0 \rangle$ .

*Proof:* The first assertion follows from

$$c_{ij} - \bar{c}_{ij} = c_{ij} - \bar{y}_i + \bar{y}_j = \bar{z}_{ij} \geq 0$$

for  $(i,j) \in \mathcal{R}$ .

Since  $\mathcal{R}$  does not contain any edge in  $\bar{\mathcal{T}}_D$ ,  $\bar{y}$  is still a node potential vector of  $\langle \bar{D}_0 \rangle$  corresponding to  $\bar{\mathcal{T}}_D$ . This and the relation

$$\bar{c}_{ij} - \bar{y}_i + \bar{y}_j = 0 \text{ for } (i,j) \in \mathcal{R}$$

imply that  $\bar{y}$  is a node potential vector of  $\langle \bar{D}_0 \rangle$  corresponding to  $\bar{\mathcal{T}}_P$ .



Now we claim that  $\bar{\mathcal{T}}_P$  is dual feasible for  $\langle \bar{D}_0 \rangle$ , which proves the second assertion. In fact, for  $(i, j) \notin \bar{\mathcal{T}}_P$ , since  $(i, j) \notin \mathcal{R}$  for such  $(i, j)$ , the dual slack is

$$c_{ij} - \bar{y}_i + \bar{y}_j = \bar{z}_{ij} \geq 0.$$

This completes the proof.  $\square$

Next we show how to obtain a dual feasible spanning tree  $\mathcal{T}_D^{k+1}$  from  $\mathcal{T}^k$ . To show this, we consider a *general* minimum-cost network-flow problem  $\langle P \rangle$  and  $\langle D \rangle$ , and consider another problem  $\langle P' \rangle$  which is the same as  $\langle P \rangle$  except a unit cost on an arc. The next lemma shows how to obtain a dual feasible spanning tree for the dual of  $\langle P' \rangle$  from an optimal spanning tree of  $\langle P \rangle$  and  $\langle D \rangle$ .

**Lemma 8** *Let  $\tilde{\mathcal{T}}$  be an optimal spanning tree of  $\langle P \rangle$  and  $\langle D \rangle$ , and  $(\tilde{y}, \tilde{z})$  be the corresponding optimal solution of  $\langle D \rangle$ . We define another minimum-cost network-flow problem  $\langle P' \rangle$  which has the same feasible region as  $\langle P \rangle$  and the cost function*

$$\sum_{(i,j) \in \mathcal{A}} \hat{c}_{ij} x_{ij}$$

where

$$\hat{c}_{ij} = \begin{cases} c_{ij} & \text{if } (i, j) \neq (u, v) \\ \text{a number greater than } c_{ij} & \text{if } (i, j) = (u, v). \end{cases}$$

Namely, we assume that the cost function of  $\langle P' \rangle$  is different from that of  $\langle P \rangle$  only on an arc  $(u, v)$ . Let  $\langle D' \rangle$  be the dual of  $\langle P' \rangle$ .

1. If  $(u, v) \notin \tilde{\mathcal{T}}$ , then  $\tilde{\mathcal{T}}$  is dual feasible for  $\langle D' \rangle$ , thus optimal for  $\langle P' \rangle$  and  $\langle D' \rangle$ .
2. If  $(u, v) \in \tilde{\mathcal{T}}$ , let  $U$  and  $V$  be as in (2.3) and (2.4),

$$\delta := \min \{ \tilde{z}_{ij} + \hat{c}_{ij} - c_{ij} \mid (i, j) \in U \times V \}, \tag{4.3}$$

and  $(r, s)$  be a minimum achieving arc. The spanning tree

$$\tilde{\mathcal{T}}' := \tilde{\mathcal{T}} - \{(u, v)\} + \{(r, s)\}$$

is dual feasible for  $\langle D' \rangle$ .

*Proof:* Note that  $\langle D' \rangle$  is written as

$$\langle D' \rangle \begin{cases} \text{maximize} & \sum_{i \in N} b_i y_i \\ \text{subject to} & z_{ij} = \hat{c}_{ij} - y_i + y_j, \quad (i, j) \in \mathcal{A} \\ & z_{ij} \geq 0, \quad (i, j) \in \mathcal{A}. \end{cases}$$

The assumption  $(u, v) \notin \tilde{\mathcal{T}}$  implies that  $\tilde{y}$  is a node potential vector of  $\langle D' \rangle$  corresponding to  $\tilde{\mathcal{T}}$ . Therefore, the dual slack is nonnegative since

$$\hat{c}_{ij} - \tilde{y}_i + \tilde{y}_j \geq c_{ij} - \tilde{y}_i + \tilde{y}_j = \tilde{z}_{ij} \geq 0$$

holds for  $(i, j) \in \mathcal{A}$ . This proves the first assertion.

In case where  $(u, v) \in \tilde{\mathcal{T}}$ , we put

$$\begin{aligned} \tilde{y}'_i &:= \tilde{y}_i + \delta & \text{if } i \in U, \\ \tilde{y}'_i &:= \tilde{y}_i & \text{if } i \in V, \end{aligned}$$

and show that  $\tilde{y}'$  is a node potential vector of  $\langle D' \rangle$  corresponding to  $\tilde{T}'$ . Let  $\tilde{z}'_{ij} := \hat{c}_{ij} - \tilde{y}'_i + \tilde{y}'_j$  for  $(i, j) \in \mathcal{A}$ . Obviously, for  $(i, j) \in U \times U$  or  $V \times V$ ,  $\tilde{z}'_{ij} = \tilde{z}_{ij}$  holds, thus if, furthermore,  $(i, j) \in \tilde{T}$ , then  $\tilde{z}'_{ij} = \tilde{z}_{ij} = 0$ . Since  $(r, s)$  achieves the minimum of (4.3), we have

$$\begin{aligned} \tilde{z}'_{rs} &= \hat{c}_{rs} - \tilde{y}'_r + \tilde{y}'_s \\ &= \hat{c}_{rs} - c_{rs} + c_{rs} - \tilde{y}_r - \delta + \tilde{y}_s \\ &= \hat{c}_{rs} - c_{rs} + \tilde{z}_{rs} - \delta \\ &= 0. \end{aligned} \tag{4.4}$$

Therefore,  $\tilde{y}'$  is a node potential vector of  $\langle D' \rangle$  corresponding to  $\tilde{T}'$ .

Finally, we show that  $\tilde{z}'$  is nonnegative. For  $(i, j) \in U \times U$ ,  $V \times V$ , and  $V \times U$ , it is obvious that  $\tilde{z}'_{ij} \geq \tilde{z}_{ij} \geq 0$ . For  $(i, j) \in U \times V$  we have

$$\begin{aligned} \tilde{z}'_{ij} &= \hat{c}_{ij} - \tilde{y}'_i + \tilde{y}'_j \\ &= \hat{c}_{ij} - c_{ij} + c_{ij} - \tilde{y}_i - \delta + \tilde{y}_j \\ &= \hat{c}_{ij} - c_{ij} + \tilde{z}_{ij} - \delta \\ &\geq 0 \end{aligned} \tag{4.5}$$

by definition of  $\delta$ . This completes the proof.  $\square$

The whole algorithm is as follows.

**procedure** *solve-network* $((x, \mathcal{T}_P), (y, z, \mathcal{T}_D))$

**begin**

Set  $\mathcal{R} = \mathcal{T}_P \setminus \mathcal{T}_D$  and  $\mathcal{R}_k$  for  $k = 1, \dots, |\mathcal{R}|$  as in the above.

Set  $\bar{c}_{ij} = y_i - y_j$  if  $(i, j) \in \mathcal{R}$  and  $\bar{c}_{ij} = c_{ij}$  otherwise.

**for**  $k = 0, \dots, |\mathcal{R}| - 1$  **do**

**begin**

Set  $(u, v) \in \mathcal{R}_{k+1} - \mathcal{R}_k$ ;

Set  $U$  and  $V$  as in (2.3) and (2.4);

Set  $\delta = \min\{z_{ij} + \bar{c}_{ij} - c_{ij} \mid (i, j) \in U \times V\}$  and  $(r, s)$  be a minimum achieving arc;

**if**  $(u, v) \neq (r, s)$  **then**

**begin**

$\mathcal{T}_D = \mathcal{T}_P - \{(u, v)\} + \{(r, s)\}$ ;

*primal-dual-pivot* $((x, \mathcal{T}_P), (y, z, \mathcal{T}_D))$ ;

**end**

**end**

**end**

Now we establish the following complexity result.

**Theorem 9** *The algorithm solve-network terminates within  $2n^3 \max_{i \in N} b_i$  primal pivots.*

*Proof:* Each call for *primal-dual-pivot* may have  $2n^2 \max_{i \in N} b_i$  primal pivots due to Theorem 9. The number of call is bounded by  $|\mathcal{R}|$ , thus  $n$ , from which the theorem follows.  $\square$

### 5. Application to the Shortest-Path Problems

In this section, we apply the simplex method using the primal-dual symmetric pivoting rule to the shortest-path problem to find shortest paths from a source node  $a$  to the other nodes:

$$\langle SP \rangle \left\{ \begin{array}{l} \text{minimize} \quad \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} \\ \text{subject to} \quad \sum_{l:(i,l) \in \mathcal{A}} x_{il} - \sum_{l:(l,i) \in \mathcal{A}} x_{li} = -1, \quad i \in N - \{a\} \\ \sum_{l:(a,l) \in \mathcal{A}} x_{al} - \sum_{l:(l,a) \in \mathcal{A}} x_{la} = n - 1, \\ x_{ij} \geq 0 \quad (i, j) \in \mathcal{A}, \end{array} \right. \quad (5.1)$$

where  $c_{ij} \geq 0$  for  $(i, j) \in \mathcal{A}$  stands for the distance between nodes  $i$  and  $j$ . The dual of (5.1) is the following:

$$\langle DSP \rangle \left\{ \begin{array}{l} \text{maximize} \quad (n - 1)y_a - \sum_{i \neq a} y_i \\ \text{subject to} \quad z_{ij} = c_{ij} - y_i + y_j \quad (i, j) \in \mathcal{A} \\ z_{ij} \geq 0 \quad (i, j) \in \mathcal{A}. \end{array} \right. \quad (5.2)$$

It is well-known that any feasible spanning tree of  $\langle SP \rangle$  is nondegenerate. Therefore, we have the following theorem.

**Theorem 10** *If we have a pair of primal feasible and dual feasible spanning trees sharing all arcs but one, then primal-dual-pivot solves  $\langle SP \rangle$  and  $\langle DSP \rangle$  within  $n - 1$  primal pivots.*  
*Proof:* An initial flow on the difference arc is bounded by  $n - 1$ , and since the spanning tree is nondegenerate, each primal pivot decrease the flow at least by 1. Thus the theorem readily follows.  $\square$

Next we assume that arbitrary initial primal and dual feasible spanning trees are given. Notice that in the algorithm described in the previous section, we have a freedom to select  $\mathcal{R}_k$ . To obtain a better complexity result for the shortest-path problems, we specify how to select  $\mathcal{R}_k$ . Namely, we determine  $\mathcal{R}_{k+1}$  from  $\mathcal{R}_k$  dynamically at the end of each *primal-dual-pivot*. Let  $x$  be the primal basic solution corresponding to the optimal spanning tree for  $\langle P_k \rangle$ ,

$$(u, v) = \operatorname{argmin} \{ x_{ij} \mid (i, j) \in \mathcal{R}_k \},$$

and  $\mathcal{R}_{k+1} = \mathcal{R}_k - \{(u, v)\}$ . Using this selection, we have the following complexity result.

**Theorem 11** *If we have a pair of arbitrary primal feasible and dual feasible spanning trees, then the algorithm solve-network where  $\mathcal{R}_k$  is selected as above solves  $\langle SP \rangle$  and  $\langle DSP \rangle$  within  $n(n - 1)/2$  primal pivots.*

**Lemma 12** *Let  $\mathcal{E}$  be a set of  $d$  edges contained in a primal feasible spanning tree of  $\langle SP \rangle$  whose basic solution is  $x$ . There exists an edge  $(i, j)$  in  $\mathcal{E}$  such that  $x_{ij} \leq n - d$ .*

*Proof:* We first show the case where  $\mathcal{E}$  is a tree containing the source  $a$ . Note that a flow of an edge is the number of nodes which is *under* that edge. (We assume that the source  $a$  is put at the top of the tree; an edge of a primal feasible spanning tree of shortest-path problem always points out from the source.) The number of nodes which is not contained in the tree is  $n - d - 1$ , thus there exists an edge in the tree whose flow is less than or equals to  $n - d$ .

If  $\mathcal{E}$  is not a tree containing  $a$ , we can make a tree  $\mathcal{S}$  containing  $a$  by adding some edges of the primal feasible spanning tree to  $\mathcal{E}$  such that any leaf edge is contained in  $\mathcal{E}$ . The lemma is now obvious since  $n - |\mathcal{S}| \leq n - d$ .  $\square$

*Proof of Theorem 11* : Since any feasible spanning tree is nondegenerate, one primal pivot of the *primal-dual-pivot* decreases the flow  $x_{uv}$  at least by 1. Suppose that we have finished the  $k$ -th execution of *primal-dual-pivot*. If  $\mathcal{R}_k \not\subseteq \mathcal{T}_P$ , then we have an edge  $(u, v) \in \mathcal{R}_k$  such that  $x_{uv} = 0$ , which means that  $\mathcal{T}_P$  is optimal for  $\langle \bar{P}_{k+1} \rangle$  and  $\langle \bar{D}_{k+1} \rangle$  with  $\mathcal{R}_{k+1} = \mathcal{R}_k - \{(u, v)\}$ . If  $\mathcal{R}_k \subseteq \mathcal{T}_P$ , then by Lemma 12, there exists an edge  $(u, v)$  in  $\mathcal{R}_k \subseteq \mathcal{T}_P$  such that  $x_{uv} \leq n - |\mathcal{R}_k|$ . Therefore, the total number of primal pivots is bounded by

$$\sum_{k=1}^{|\mathcal{R}|} (n - k) \leq \sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}.$$

This completes the proof.  $\square$

Since the algorithm described above requires initial feasible solutions, one might think that the result is less important. One advantage of this algorithm however, appears in the case where the cost vector is dynamically changing. Suppose that we once solve a shortest-path problem to get an optimal spanning tree and then cost of an edge is changed. Using the same procedure in Section 4 we can obtain a spanning tree feasible for dual of the changed shortest-path problem, having one different arc from the optimal spanning tree which is primal feasible for the modified problem. Now we can apply the algorithm *primal-dual-pivot* to obtain an optimal spanning tree of the new shortest-path problem within  $n - 1$  iterations (See Theorem 10).

The best bound for the network simplex method for the shortest-path problem is  $(n - 1)(n - 2)/2$  due to Goldfarb, Hao, and Kai [10]. Our result is almost the same as their bound; the coefficient of  $n^2$  is the same. In the Goldfarb, Hao, and Kai's method, the initial primal feasible spanning tree must have the *star* shape, namely, the source node is connected to the other nodes directly. Compared to their method, our method is more flexible since an arbitrary spanning tree can be used for initial spanning tree. On the other hand, we have a disadvantage that the same number of dual pivots should be performed as that of the primal pivots. Therefore, one iteration of our method will cost more.

## 6. Concluding Remarks

We have analyzed convergence property of the network simplex version of the Chen, Pardalos, and Saunders's simplex method. We obtained the following results:

1. The procedure *primal-dual-pivot* solves minimum-cost network-flow problems within pseudopolynomial time if a pair of primal strongly feasible and dual feasible spanning tree which share all arcs but one is given.
2. For the case where we have a pair of primal strongly feasible and dual feasible spanning trees which are different from each other by more than one arcs, the procedure *solve-network* solves the problem within pseudopolynomial time.
3. The procedure *solve-network* also solves the shortest-path problems within  $n(n - 1)/2$  primal pivots if initial primal feasible and dual feasible trees are given.

One can apply the same strategy as described in Section 5 to choose  $\mathcal{R}_k$  for reducing the coefficient of the bound for the number of primal pivots for minimum-cost network-flow problems, although we did not describe it in this paper. The reason is because we want to keep the description of the algorithm as simple as possible, and, since the bound is pseudopolynomial, the coefficient is not so important. Polynomial or strongly polynomial time complexity of *primal-dual-pivot* and *solve-network* is the subject for further research.

An efficient implementation of one iteration of *primal-dual-pivot* is another interesting subject. Note that we do not need to hold two spanning trees; we only need one spanning tree and a difference arc.

Finally, numerical experience will show whether this algorithm is really efficient or not in practice.

### Acknowledgements

A part of this research was done when the author stayed at Princeton University in the United States. The author thanks to Professor R. J. Vanderbei of Princeton University for inviting him and for stimulating discussions. The author also thanks to two anonymous referees for their valuable comments which lead him to improve the presentation of the paper considerably.

### References

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin: *Network Flows* (Prentice Hall, 1993).
- [2] R. K. Ahuja and J. B. Orlin: The scaling network simplex algorithm. *Operations Research*, **40**, Supplement 1 (1992) S5–S13.
- [3] R. Armstrong and Z. Jin: A strongly polynomial dual network simplex algorithm. *Mathematical Programming*, **78** (1997) 131–148.
- [4] R. S. Barr, F. Glove, and D. Klingman: The alternating path basis algorithm for the assignment problem. *Mathematical Programming Study*, **13** (1977) 1–13.
- [5] H.-D. Chen, P. M. Pardalos, and M. A. Saunders: The simplex algorithm with a new primal and dual pivot rule. *Operations Research Letters*, **16** (1994) 121–127.
- [6] W. H. Cunningham: A network simplex method. *Mathematical Programming*, **11** (1976) 105–116.
- [7] G. B. Dantzig: Application of the simplex method to a transportation problem. In T. C. Koopmans (eds.): *Activity Analysis and Production and Allocation* (Wiley, New York, 1951), 209–213.
- [8] G. B. Dantzig: On the shortest route through a network. *Management Science*, **6** (1960) 187–190.
- [9] K. Dorios, and K. Paparrizos: Resolution of the problem of degeneracy in a primal and dual simplex algorithm. *Operations Research Letters*, **20** (1997) 45–50.
- [10] D. Goldfarb, J. Hao, and S. Kai: Efficient shortest path simplex algorithms. *Operations Research*, **38** (1990) 624–628.
- [11] E. L. Johnson: Networks and basic solutions. *Operations Research*, **14** (1966) 619–624.
- [12] J. B. Orlin, S. A. Plotkin, and E. Tardos: Polynomial dual network simplex algorithms. *Mathematical Programming*, **60** (1993) 255–276.
- [13] J. B. Orlin: A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, **78** (1997) 109–129.
- [14] S. J. Wright: *Primal-dual Interior-point Methods* (SIAM Philadelphia, 1997).

Masakazu Muramatsu  
Department of Mechanical Engineering  
Sophia University  
Chiyoda-ku, Tokyo 102-8554, Japan  
E-mail: muramatu@me.sophia.ac.jp