PAPER

# An Optimal File Transfer on Networks with Plural Original Files

Yoshihiro KANEKO[†a)] *and* Shoji SHINODA[††], *Regular Members*

**SUMMARY** A problem of obtaining an optimal file transfer of a file transmission net $N$ is to consider how to transmit, with the minimum total cost, copies of a certain file of information from some vertices, called sources, to other vertices of $N$ by the respective vertices' copy demand numbers. This problem is *NP-hard* for a general file transmission net $N$. Some classes of $N$, on each of which a polynomial time algorithm for obtaining an optimal file transfer can be designed, are known. In the characterization, we assumed that file given originally to the source remains at the source without being transmitted. In this paper, we relax the assumption to the one that a sufficient number of copies of the file are given to the source and those copies can be transmitted from the source to other vertices on $N$. Under this new assumption, we characterize a class of file transmission nets, on each of which a polynomial time algorithm for obtaining an optimal file transfer can be designed. A minimum spanning tree with degree constraints plays a key role in the algorithm.

***key words:*** *file transfer, original files, single source, minimum spanning tree with degree constraints*

## 1. Introduction

As a network theoretical problem, it is fundamental to consider how to distribute, with the minimum total cost, a certain information file within a network system. Our file transfer problem is to define a network model $N$ called a file transmission net and to determine how the copies of a certain file, denoted by $J$, are to be transmitted as well as duplicated so as to meet the demands within $N$.

The data transfer problem [1], the scheduling file transfer problem [2], the file allocation problem [3] and the file assignment problem [4], which have already been considered by other authors, seem to be similar to ours. However, each of these is also quite different from ours because they do not take account of the cost of making copies of a file at vertices in the total cost of file distribution. Indeed, the idea of making copies of a file at vertices is one of the features of our problem formulation. Our problem or its variations, including its generalization, might be essential in situations where the cost of making copies of a certain file of information data preserved by copyright must be taken into consideration in comparison with the cost of file transmission through arcs on computer networks. For instance, in the coming paperless society, the cost of duplicating information on the Internet might not be negligible from the viewpoint of copyrights.

In [5], we proved that our problem is *NP-hard*, unless a class of file transmission nets to which $N$ belongs is restricted. In [6]–[8], also, we characterized a class of file transmission nets, on each of which a polynomial time algorithm for obtaining an optimal file transfer can be designed. In the characterization, we assumed that file $J$ is given to the source from outside $N$ and that its copies are distributed from the source after a necessary number of copies are made there.

In addition to that, we have so far assumed that file $J$ given originally to the source always remains at that source without being transmitted [5]–[8]. In this paper, we relax the assumption to the one that a sufficient number of copies of $J$ are given to the source, which can be transmitted from the source to other vertices on $N$. Under the new assumption, we characterize a class of file transmission nets, on each of which a polynomial time algorithm for obtaining an optimal file transfer can be designed. A minimum spanning tree with degree constraints plays a key role in the algorithm.

In the rest of the paper, we proceed as follows. In Sect. 2, we give preliminaries for this study. In Sect. 3, we introduce the superimposition net and tree-type file transfers, and show the optimality of such file transfers. In Sect. 4, we propose how to synthesize an optimal file transfer. Finally, we give concluding remarks and discuss future tasks.

## 2. Preliminaries

In this section, we define the fundamental terms required to formulate our problem. For basic terms such as vertex, arc, walk, cycle, complete graph, and minimum spanning tree, on graph theory, refer to [9]. In the following, unless otherwise stated, an edge, an arc, a walk, a cycle and a graph indicate an undirected edge, a directed arc, a directed walk, a directed cycle and a directed graph, respectively. A graph $G$ with a vertex set $V$ and an arc set $A$ is denoted by $G=(V,A)$ and sometimes the arc set $A$ is denoted by $A(G)$. The set of arcs incident from vertex $v$ is denoted by $\delta_+(v)$ and the set of arcs incident to $v$ is denoted by $\delta_-(v)$. The

Manuscript received July 5, 2001.
Manuscript revised February 8, 2002.
Final manuscript received June 24, 2002.
[†]The author is with the Faculty of Engineering, Gifu University, Gifu-shi, 501-1193 Japan.
[††]The author is with the Faculty of Science and Engineering, Chuo Univeristy, Tokyo, 112-8551 Japan.
a) E-mail: kaneko@info.gifu-u.ac.jp

numbers $|\delta_+(v)|$ and $|\delta_-(v)|$ are called the out-degree and the in-degree of $v$. A vertex with no out-degree is called a leaf of $G$.

We consider a network $N$, called a file transmission net, such that 1) the topology of $N$ is a finite connected simple graph with vertex set $V$ and arc set $A$, 2) copies of file $J$ of information data are given to specified vertices called sources and those copies of $J$ are sometimes called the original files, 3) we can make copies of $J$ at any vertex, transmit those copies through arcs along their directions and take out those copies from each vertex of $N$ by the number needed at the vertex, 4) each vertex $v$ has three parameters $s_V$, $c_V$, and $d$, called original number, copying cost, and copy demand of $v$, respectively, and each arc $e$ has one parameter $c_A$ called transmission cost of $e$. In $N$, a copy of a copy of $J$ is nothing but a copy of $J$, and there is no difference between the original file $J$ and its copy. We call a copy of $J$ a $J$-copy. For each vertex $v$, $s_V(v)$ indicates the number of $J$-copies given to $v$ from outside $N$. Let $S = \{v \in V \mid s_V(v) > 0\}$. Each vertex in $S$ is called a *source*. For each vertex $v$, $c_V(v)$ indicates the unit cost of making one $J$-copy at $v$ and $d(v)$ indicates the number of $J$-copies needed at $v$. Let $U = \{v \in V \mid d(v) > 0\}$. For each arc $e$, $c_A(e)$ indicates the unit cost of transmitting one $J$-copy through $e$. The above file transmission net $N$ is denoted by $N = (V, A, s_V, c_V, d, c_A)$.

In this paper, we assume that $S = \{v_1\}$ and $s_V\colon V \to Z_0^+$, $c_V\colon V \to Z^+$, $d\colon V \to \{0,1\}$, and $c_A\colon A \to Z^+$, where $Z^+$ is the set of all positive integers and $Z_0^+ = Z^+ \cup \{0\}$. In addition, if $(x,y) \in A$, then $(y,x) \in A$ and $c_A(x,y) = c_A(y,x)$.

For a walk $p$, let $s(p)$ and $t(p)$ be the initial vertex and the terminal vertex of $p$, respectively, that is, $p$ begins at $s(p)$ and ends at $t(p)$. If a walk $p$ satisfies $s(p) = x$ and $t(p) = y$, then $p$ is called an $x - y$ walk. For a walk $p$, let us denote by $c_A(p)$ the total arc cost on $p$. For two vertices $x$ and $y$, if an $x - y$ walk $p$ satisfies $c_A(p) \le c_A(p')$ for every other $x - y$ walk $p'$, then $p$ is called a minimum cost $x - y$ path and its cost is denoted by $c_{x,y}$. For source $v_1$, we define $v_1 - v_1$ path $p_0$ composed of a single vertex satisfying $s(p_0) = t(p_0) = v_1$ and $c_{v_1,v_1} = 0$. For a mapping $f$ on $A$, if a walk $W$ satisfies $f(e) > 0$ for all arcs $e$ on $W$, then we say that $W$ is *f-connected*.

**Definition 1:** A pair of mappings $\varphi\colon V \to Z_0^+$ and $f\colon A \to Z_0^+$, denoted by $D = (\varphi, f)$, of $N$ is called a file transfer of $N$ if the following rules are satisfied.

(C1) The conservation of $J$-copies at a vertex on $N$: For any vertex $v \in V$, the following holds:

$$s_V(v_1) + \varphi(v_1) + \sum_{e \in \delta_-(v_1)} f(e) \ge \sum_{e \in \delta_+(v_1)} f(e) + d(v_1),$$

$$s_V(v) + \varphi(v) + \sum_{e \in \delta_-(v)} f(e) = \sum_{e \in \delta_+(v)} f(e) + d(v) \qquad (v \in V - \{v_1\}).$$

(C2) The $f$-connectivity of $J$-copies through arcs on $N$: For any vertex $v$ such that $\varphi(v) > 0$, there exists an $f$-connected walk from $v_1$ to $v$.

For a file transfer $D = (\varphi, f)$,

$$C(D) = \sum_{v \in V} \varphi(v) c_V(v) + \sum_{e \in A} f(e) c_A(e),$$

is called the cost of $D$. A file transfer $D$ is called an *optimal* file transfer of $N$ if $C(D) \le C(D')$ for every other file transfer $D'$. For a file transfer $D = (\varphi, f)$, also,

$$z(D) = s_V(v_1) + \varphi(v_1) + \sum_{e \in \delta_-(v_1)} f(e) - \sum_{e \in \delta_+(v_1)} f(e) - d(v_1),$$

is called the residue of $J$-copies at $v_1$. □

In a file transfer $D$, $z(D)$ $J$-copies remain at $v_1$ without being transmitted. It is of great concern to us in this paper to obtain an optimal file transfer of $N$. A subset $M$ of $V$ defined by

$$M = \{m \in V | c_V(m) < c_V(x) + c_{x,m} \text{ for } \forall x \in V - \{m\}\}$$

is very useful for obtaining an optimal file transfer of $N$.

As an example, we consider the file transmission net $N$ shown in Fig. 1, where each undirected line represents a pair of arcs with opposite directions. The number attached to each vertex indicates its copying cost per $J$-copy, and the number attached to each undirected line indicates its transmission cost per $J$-copy. Suppose that $U = V$ and assume that the source $v_1$ satisfies $s_V(v_1) = 3$. Then, by easy calculation, $M = \{v_2, v_4, v_6\}$. The cost of file transfer $D = (\varphi, f)$ shown in Fig. 2 is

$$\begin{aligned} C(D) &= 1 \cdot 4 + 2 \cdot 1 \quad \text{(copying cost)} \\ &\quad + 1 \cdot 1 + 2 \cdot 2 + 1 \cdot 2 + 2 \cdot 1 + 1 \cdot 2 \quad \text{(transmission cost)} \\ &= 17. \end{aligned}$$

In Fig. 2, the number attached to vertex $v$ represents $\varphi(v)$, and the number attached to arc $e$ represents $f(e)$. Note that any arc $e$ satisfying $f(e) = 0$ is not drawn in Fig. 2.

As was shown in [5], generally, the problem of obtaining an optimal file transfer is $NP$-hard. The relaxed problem in this paper is also $NP$-hard since it

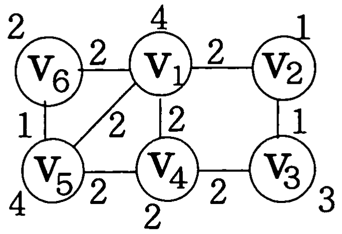


**Fig. 1** A file transimission net $N$.

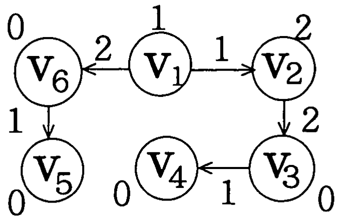


**Fig. 2** A file transfer $D = (\varphi, f)$.

can be reduced to the problem in [5] by restricting $s_V(v_1) = 1$. As was shown in [6]–[8], there is a class of file transmission nets to each of which a polynomial time algorithm for obtaining an optimal file transfer can be designed. A sufficient condition for the problem to be solved in polynomial time in the case of $s_V(v_1) = 1$ is that $M \cup \{v_1\} \subseteq U$ holds.

Suppose that the relaxed problem also satisfies this parametric condition. Then we have the following lemma, if we set $S = \{v_1\}$ for Prop.2 given in [7].

**Lemma 1**[7]: Suppose that a file transfer $D = (\varphi,\ f)$ of $N$ satisfies $\varphi(u) > 0$ for a vertex $u \in V - M$. Then, $N$ has another file transfer $D' = (\varphi',\ f')$ such that $C(D) \geq C(D')$ and $\varphi'(v) = 0$ for any vertex $v \in V - M$. □

By this lemma, in order to obtain an optimal file transfer of $N$, we have only to consider file transfers such that $M$ contains all vertices at each of which $J$-copies are made. Therefore, we assume that every file transfer $(\varphi,\ f)$ described below satisfies the following property:

Property($\psi$1): If $\varphi(v) > 0$, then $v \in M$.

In addition, taking account of the definition of optimality, we can assume that every file transfer $D = (\varphi,\ f)$ described below satisfies the following property:

Property($\psi$2): If $\varphi(v_1) > 0$ , then $z(D) = 0$.

Also, every file transfer satisfies the following lemma given in [7], which plays a key role in proving the propositions described below.

**Lemma 2**[7]: For any vertex $u \in U - \{v_1\}$, every file transfer $(\varphi,\ f)$ of $N$ contains an $f$-connected walk from $v_1$ to $u$. □

## 3. Optimality of Tree-Type File Transfers

### 3.1 Superimposition Net of Walks

In the following, a net of $N$, denoted by $N(\alpha, \beta)$, means a network whose structure is identical to that of $N$ and with each vertex $v$ and each arc $e$, $\alpha(v)$ and $\beta(e)$ are associated, respectively. Note that file transfer itself is a net of $N$. In the following, we say a walk $p$ is a cycle if $s(p) = t(p)$ and otherwise $p$ is a path. For a walk set, the superimposition net is as follows.

**Definition 2:** With respect to a walk set $W$ on $N$, the superimposition net $N(W) = N(\alpha, \beta)$ of $N$ is defined to be a network such that the following are satisfied.

(N1) The vertex set and the arc set of $N(W)$ are $V$ and $A$ of $N$, respectively.

(N2) For each vertex $v$, $\alpha(v)$ is (the number of all paths in $W$ that have $v$ as the initial vertex in common)—(the number of all paths in $W$ that have $v$ as the terminal vertex in common). When a vertex $v$ is neither the initial vertex nor the terminal vertex of any path in $W$, we define $\alpha(v) = 0$.

(N3) For each arc $e$, $\beta(e)$ is the total number of all walks that contain $e$. When an arc $e$ is not contained in any walk in $W$, we define $\beta(e) = 0$. □

The above defined mappings $\alpha$ and $\beta$ satisfy $\alpha$: $V \to Z$ and $\beta$: $A \to Z_0^+$, where $Z$ is the set of all integers. For the partition of a walk set $W$ into path set $P$ and cycle set $C$, the superimposition net $N(W)$ is sometimes represented by $N(P \cup C)$ in place of $N(W)$. If $P = \phi$, then $N(P \cup C) = N(C)$ and if $C = \phi$, then $N(P \cup C) = N(P)$. Thus, in the following, when we deal with $N(P \cup C)$, it may happen that $P = \phi$ or $C = \phi$. A path set $P$ and a cycle set $C$ on $N$ are said to be *disjoint* if no path in $P$ and no cycle in $C$ have a common vertex. As shown in [8], when we deal with $N(P \cup C)$, we assume that the path set $P$ and the cycle set $C$ is disjoint or $C = \phi$.

**Lemma 3:** For a path set $P$ and a cycle set $C$ on $N$, and for the above defined mappings $\alpha$ on $V$ and $\beta$ on $A$, we have

$$\sum_{v \in V} c_V(v) \cdot \alpha(v) + \sum_{e \in A} c_A(e) \cdot \beta(e)$$
$$= \sum_{p \in P} \{c_V(s(p)) + c_A(p) - c_V(t(p))\}$$
$$+ \sum_{l \in C} c_A(l)\,.$$ □

This lemma is trivial.

### 3.2 Tree-Type File Transfer

In this section, it is shown that in order to obtain an optimal file transfer, we have only to consider tree-type file transfers defined below. In the following, let $s_1 = s_V(v_1)$ for simplicity.

An arborescence is a directed tree where each vertex in-degree is at most 1 and a vertex with in-degree 0 is called a root. It is easy to see that an arborescence has a single root and any vertex has a unique path from the root. We define a special net based on an arborescence.

**Definition 3:** Let $U'$ be a vertex set with $v_1$ twice and each vertex in $U \setminus \{v_1\}$ once. Then, let an arborescence $T$ satisfy the following.

(CT) The vertex set is $U'$ such that one $v_1$ is the root and another $v_1$ is a leaf.

For each arc $(x,y)$ in $T$, let $P$ be a path set with one minimum cost $x-y$ path. For $N(\alpha,\beta) = N(P)$, a function $\alpha'$ on $V$ is defined to be

$$\alpha'(v) = \begin{cases} \max\{0,\ \alpha(v_1) + d(v_1) - s_1\}\ (v = v_1), \\ \alpha(v) + d(v) \quad (v \in V \setminus \{v_1\}). \end{cases} \tag{1}$$

□

Concerning the above, we have the next lemma.

**Lemma 4:** Suppose an arborescence $T$ satisfy (CT). Then, the net $N(\alpha',\beta)$ of Definition 3 is a file transfer of $N$.

(Proof) It is easy to see that $\alpha'\colon V \to Z$ and $\beta\colon A \to Z_0^+$.

First, we show $\alpha'\colon V \to Z_0^+$. It follows from Eq. (1) that $\alpha'(v_1) \geq 0$. Since $T$ is an arborescence, each vertex $u \in U \setminus \{v_1\}$ satisfies $\alpha(u) \geq -1$, which implies from Eq. (1) that $\alpha'(u) \geq 0$. In addition, since $T$ does not contain any vertex $v \in V \setminus U$, we have $\alpha(v) = 0$, which implies from Eq. (1) that $\alpha'(v) = 0$. Thus, $\alpha'\colon V \to Z_0^+$ holds.

Next we show that $\alpha'$ and $\beta$ satisfy (C1). In $N(\alpha,\beta)$, each vertex $v \in V$ satisfies

$$\alpha(v) + \sum_{e \in \delta_-(v)} \beta(e) = \sum_{e \in \delta_+(v)} \beta(e). \tag{2}$$

In relation to $v_1$, Eq. (1) gives $\alpha'(v_1) \geq \alpha(v_1) + d(v_1) - s_1$, which implies

$$s_1 + \alpha'(v_1) + \sum_{e \in \delta_-(v_1)} \beta(e) \geq \sum_{e \in \delta_+(v_1)} \beta(e) + d(v_1).$$

Thus, $\alpha'$ and $\beta$ satisfy (C1) for $v_1$. It follows from Eqs. (1) and (2), that $\alpha'$ and $\beta$ satisfy (C1) for each vertex in $V \setminus \{v_1\}$.

It remains to show that $\alpha'$ and $\beta$ satisfy (C2). As mentioned above, each vertex $v \in V \setminus U$ satisfies $\alpha'(v) = 0$. This means that $u \in U$ if $\alpha'(u) > 0$. It is clear from (CT) that there exists a $\beta$−connected walk from $v_1$ to each vertex in $U$. Thus, $\alpha'$ and $\beta$ satisfy (C2). Hence, we have this lemma. □

From now on, we call the net $N(\alpha',\beta)$ of Definition 3 a tree-type file transfer based on $T$.

In the following, as in the figures of file transfers, the notation of net $N(\alpha,\beta)$ means that the number attached to a vertex $v$ indicates $\alpha(v)$, the number attached to an arc $e$ indicates $\beta(e)$, and any arc $e$ such that $\beta(e) = 0$ is omitted. With the use of $N$ of Fig. 1, we illustrate the above definitions. Since $N$ satisfies $U = V$, $U' = \{v_1, v_1, v_2, v_3, v_4, v_5, v_6\}$ holds. Let $T$ be the arborescence on $U'$ shown in Fig. 3. Then, by Definition 3, we have the net $N(\alpha,\beta)$ shown in Fig. 4 and a tree-type file transfer $N(\alpha',\beta)$ based on $T$ shown in Fig. 5. The cost of such a file transfer is as follows.

**Lemma 5:** For an arborescence $T$ satisfying (CT), let $D$ be a tree-type file transfer based on $T$. Then we have

$$C(D) = \sum_{(x,y) \in A(T)} \{c_V(x) + c_{x,y}\} - \min\{k, s_1\} \cdot c_V(v_1),$$

where $k$ is the out-degree of the root $v_1$ in $T$.

(Proof) Since there holds $N(\alpha,\beta) = N(P)$, we can see, from Lemma 3, that

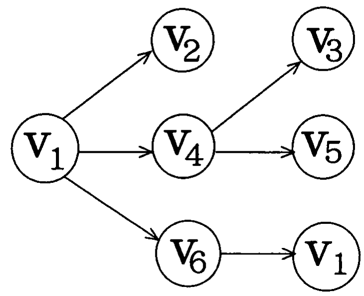


**Fig. 3** An arborescence $T$.

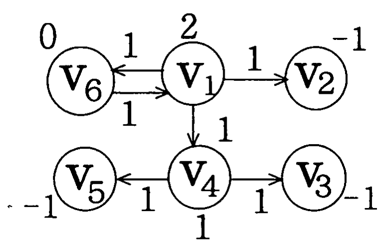


**Fig. 4** The net $N(\alpha,\beta)$.

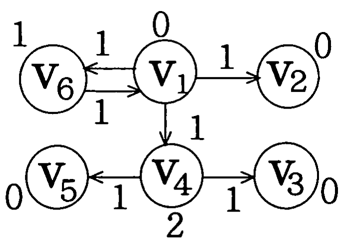


**Fig. 5** A tree-type file transfer $N(\alpha',\beta)$.

$$\sum_{v\in V} c_V(v)\cdot\alpha(v)+\sum_{e\in A} c_A(e)\cdot\beta(e) = \sum_{(x,y)\in A(T)} \{c_V(x)+c_{x,y}-c_V(y)\}.$$

It follows from (CT) that $\alpha(v_1)=k-1$, which implies, from $d(v_1)=1$ and Eq. (1), that

$$\alpha'(v_1)=\alpha(v_1)+1+max\{-k,\ -s_1\}.$$

In addition, each vertex $v\in V\backslash\{v_1\}$ satisfies

$$\sum_{v\in V\backslash\{v_1\}} c_V(v)\cdot\alpha'(v) = \sum_{v\in V\backslash\{v_1\}} c_V(v)\cdot\alpha'(v)+\sum_{v\in U\backslash\{v_1\}} c_V(v),$$

because of Eq. (1). We can see from (CT) that

$$\{y\in V|(x,y)\in A(T)\}=U.$$

Hence, we have this lemma from the above equations. □

### 3.3 Optimality

First we give some necessary definitions to prove the optimality of tree-type file transfers.

**Definition 4:** Let $v_0 \notin V$. For a path set $P$, let $G(P)$ be a directed graph with vertex set $\{v_0\}\cup\{s(p),t(p)\mid p\in P\}$ and arc set $\{(v_0,v_1)\}\cup\{(s(p),t(p))\mid p\in P\}$. In addition, let $R(P)=\{v\in V\mid G(P)$ contains a $v_0-v$ walk$\}$. □

If we let $S=\{v_1\}$ for Lemmas 6–7, and Proposition 3 in [8], then the following lemma holds.

**Lemma 6**[8]: For a file transfer $D=(\varphi,\ f)$ of $N$ satisfying $M\cup\{v_1\}\subseteq U$ and $M\backslash\{v_1\}\neq\phi$, let a function $\alpha$ on $V$ satisfy

$$\alpha(v)=\begin{cases} s_1-z(D)+\varphi(v_1)-d(v_1) & (v=v_1),\\ \varphi(v)-d(v) & (v\in V\backslash\{v_1\}).\end{cases} \tag{3}$$

Then, we have a path set $P$ satisfying the following.

(P1) For a cycle set $C$, $N(P\cup C)$ is identical to $N(\alpha,\ f)$ and each path $p\in P$ satisfies $s(p)\in\{v_1\}\cup M$ and $t(p)\in U$.

(P2) $P$ has a subset $P'$ such that $G(P')$ is an arborescence with vertex set $\{v_0,v_1\}\cup M$ and root $v_0$. □

It may happen that a path set satisfying (P1) has no path which begins at $v_1$, e.g., if $\alpha(v_1)<0$. From this lemma, we have the next proposition which supports the optimality of tree-type file transfers.

**Proposition 1:** For any file transfer $D$ of $N$ such that $M\cup\{v_1\}\subseteq U$, there exists a tree-type file transfer $D'$, satisfying $C(D)\geq C(D')$, based on $T$ which is an arborescence satisfying (CT).

(Proof) In order to prove the proposition, we suppose that any file transfer $D=(\varphi,\ f)$ below satisfies Properties ($\psi1$) and ($\psi2$). For $\alpha$ defined as Eq. (3), Lemma 6 says that we have a path set $P$ and a cycle set $C$ such that $N(P\cup C)$ is identical to $N(\alpha,\ f)$. In the following, we consider the case of $M\backslash\{v_1\}\neq\phi$. Using the path set $P_M\ (\subseteq P)$ satisfying (P2), we divide the above $P$ into four subsets: $P_M$,

$$\begin{aligned} P_{M'} &= \{p\in P\backslash P_M \mid t(p)\in M\backslash\{v_1\}\},\\ P_S &= \{p\in P\backslash P_M \mid t(p)=v_1\},\ \text{and}\\ P_U &= \{p\in P\backslash P_M \mid t(p)\in U\backslash(M\cup\{v_1\})\}.\end{aligned}$$

Suppose that $P_{M'}$ contains a path $p$ which begins at $v_1$. Then, since $\alpha(t(p))\geq -1$ by Eq. (3) and $P_M$ also contains a path which ends at $t(p)$, the original $P$ has at least one path which begins at $t(p)$. Let $p'$ be such a path and let $p''$ be a $v_1-t(p')$ path composed of $p$ and $p'$. Then, $N(P\backslash\{p,p'\}\cup\{p''\})$ is identical to $N(P)$ as well as $P\backslash\{p,p'\}\cup\{p''\}$ satisfying (P1) and (P2). Next, we consider $P\backslash\{p,p'\}\cup\{p''\}$ instead of $P$.

We can continue this process until we obtain another $P$ satisfying (P1) and (P2) such that no path in $P_{M'}$ begins at $v_1$. In the following, let $k_{out}$ and $k_{in}$ be the out-degree and the in-degree of $v_1$ in $N(P)$, respectively. Then for such $P$, we have $k_{out}=|\{p\in P_M\cup P_U|\ s(p)=v_1\}|$ and $k_{in}=|P_S|$.

Here, we rearrange $P$ to prove this proposition. First, we consider the case of $k_{out}>k_{in}\geq 1$. This implies $P_S\neq\phi$. Let $p\in P_S$. Note here from $P\cap C=\phi$ that $s(p)\neq v_1$. We can see from $k_{out}>1$ that $P_M\cup P_U$ contains a path $p'$ such that $s(p')=v_1$ and $t(p')\neq s(p)$. Let $p''$ be an $s(p)-t(p')$ path composed of $p$ and $p'$, and let $P'_S=P_S\backslash\{p\}$. If $p'\in P_M$, then let $P'_M=P_M\backslash\{p'\}\cup\{p''\}$ and $P'_U=P_U$. Otherwise, let $P'_M=P_M$ and $P'_U=P_U\backslash\{p'\}\cup\{p''\}$. Moreover, let $P'=P'_M\cup P'_U\cup P'_S$. Then we can see that $N(P')$ is identical to $N(P)$ as well as $P'$ satisfying (P1) and (P2). For this $P'$, let $k'_{out}$ and $k'_{in}$ be the out-degree and the in-degree of $v_1$ in $N(P')$, respectively. Then, we have $k'_{out}>k'_{in}$ because $k'_{out}=k_{out}-1$ and $k'_{in}=k_{in}-1$. Next, we consider $P'$ instead of $P$.

We can continue this process until $k'_{in}=0$, that is, we get another $P$ satisfying $P_S=\phi$ as well as (P1) and (P2). Hence, in the following, if $P$ satisfies $k_{out}>k_{in}$, then $k_{in}=0$, i.e., $P_S=\phi$.

In a similar way, suppose that $P$ satisfies $k_{out}=1$ if $k_{out}\leq k_{in}$. Note here that $k_{out}>0$ because $M\backslash\{v_1\}\neq\phi$ and $P_M\neq\phi$.

In addition, if $P_{M'}\neq\phi$, then for any $p\in P_{M'}$

$$c_V(s(p))+c_A(p)>c_V(t(p)),$$

because $t(p)\in M$. Thus, by Definition 1, Lemma 3 and

Eq. (3),

$$C(D) \geq \sum_{p \in P_M \cup P_U \cup P_S} \{c_V(s(p)) + c_A(p) - c_V(t(p))\} - c_V(v_1) \cdot \{s_1 - z(D)\} + \sum_{v \in V} c_V(v) \cdot d(v). \quad (4)$$

We can see from Eq. (3), ($\psi$1), and (P1) that for any $u \in U \backslash (M \cup \{v_1\})$, $P_U$ contains just one path ending at $u$. Since $G(P_M \cup P_U)$ is an arborescence with vertex set $\{v_0\} \cup U$ and root $v_0$ because of (P2), we have

$$\sum_{p \in P_M \cup P_U} c_V(t(P)) = \sum_{v \in V \backslash \{v_1\}} c_V(v) \cdot d(v). \quad (5)$$

In the following, we consider classifying $P_S$.

(Case1) The case of $P_S = \phi$. This implies $k_{in} = 0$ and $\alpha(v_1) = k_{out}$. Let $T_1$ be an arborescence obtained from $G(P_M \cup P_U)$ by adding an arc $(v_1, v_1)$ which is not a self-loop. Then, $T_1$ satisfies (CT). Let $k'_{out}$ and $k'_{in}$ be the out-degree and in-degree of $v_1$ in $T_1$, respectively. Then we have $k'_{out} = k_{out} + 1$ and $k'_{in} = 1$. We consider classifying $\varphi(v_1)$.

(1-1) The case of $\varphi(v_1) > 0$. This implies, from ($\psi$2), that $z(D) = 0$. Therefore, it follows from Eq. (3) that $k'_{out} = \alpha(v_1) + 1 = s_1 + \varphi(v_1) > s_1$, which implies, from Eqs. (4), (5) and Lemma 6, that we have

$$\begin{aligned} C(D) &\geq \sum_{(x,y) \in A(T_1)} \{c_V(x) + c_{x,y}\} - c_V(v_1) \cdot s_1 \\ &= \sum_{(x,y) \in A(T_1)} \{c_V(x) + c_{x,y}\} - c_V(v_1) \cdot \min\{k'_{out}, s_1\}. \end{aligned}$$

Note here that $(v_1, v_1) \in A(T_1)$ and $c_A(p_0) = 0$.

(1-2) The case of $\varphi(v_1) = 0$. It follows from Eq. (3) that $k'_{out} = k_{out} + 1 = s_1 - z(D) \leq s_1$. Similarly to the case above, for the arborescence $T_1$ with vertex set $U'$, we have

$$\begin{aligned} C(D) &\geq \sum_{(x,y) \in A(T_1)} \{c_V(x) + c_{x,y}\} - c_V(v_1) \cdot k'_{out} \\ &= \sum_{(x,y) \in A(T_1)} \{c_V(x) + c_{x,y}\} - c_V(v_1) \cdot \min\{k'_{out}, s_1\}. \end{aligned}$$

(Case2) The case of $P_S \neq \phi$. This implies $k_{in} \geq k_{out} = 1$ and $\alpha(v_1) \leq 0$. Suppose that $\varphi(v_1) > 0$. Then we can see from ($\psi$2) that $z(D) = 0$, which implies, from Eq. (3), that $\alpha(v_1) = s_1 + \varphi(v_1) - d(v_1) \geq s_1 > 0$. This is a contradiction. Therefore, $\varphi(v_1) = 0$. Let $p_1 \in P_S$ and let $T_2$ be an arborescence obtained from $G(P_M \cup P_U)$ by adding an arc $(s(p_1), v_1)$. Then, $T_2$ satisfies (CT). Let $k'_{out}$ and $k'_{in}$ be the out-degree and in-degree of $v_1$ in $T_2$, respectively. Then, $k'_{out} = k_{out}$ and $k'_{in} = 1$. We can see, from Eq. (3), that $k_{out} - k_{in} = s_1 - z(D) - 1$, that is, $-k_{in} + 1 - s_1 + z(D) = -k_{out} = -k'_{out}$. Since $s_1 \geq 1 = k'_{out}$, similarly to the case above, we have

$$\begin{aligned} C(D) &\geq \sum_{p \in P_M \cup P_U \cup \{p_1\}} \{c_V(s(p)) + c_A(p) - c_V(t(p))\} \\ &\quad + \sum_{p \in P_S \backslash \{p_1\}} \{c_V(s(p)) + c_A(p) - c_V(t(p))\} \\ &\quad - c_V(v_1) \cdot \{s_1 - z(D)\} + \sum_{v \in V} c_V(v) \cdot d(v) \\ &\geq \sum_{(x,y) \in A(T_2)} \{c_V(x) + c_{x,y}\} \\ &\quad + \sum_{p \in P_S \backslash \{p_1\}} \{c_V(s(p)) + c_A(p)\} \\ &\quad - \{k_{in} - 1\} c_V(v_1) - c_V(v_1) \cdot \{s_1 - z(D)\} \\ &\geq \sum_{(x,y) \in A(T_2)} \{c_V(x) + c_{x,y}\} - c_V(v_1) \cdot \min\{k'_{out}, s_1\}. \end{aligned}$$

The remaining case is $M \backslash \{v_1\} = \phi$, that is, $M = \{v_1\}$. It is clear that $P_M = P_{M'} = \phi$. In addition, $P_S = \phi$ because $P \cap C = \phi$. Hence, $P = P_U$ holds. It is easy to see that the proposition holds for this case. □

Let us demonstrate the above Case2. Suppose that for $N$ in Fig. 1, we have a file transfer $D = (\varphi, f)$ shown in Fig. 6, where $z(D) = 4$. Let a mapping $\alpha$ on $V$ be obtained by substituting $\varphi$ into Eq. (3). Then the net $N(\alpha, f)$ is shown in Fig. 7. Also, we have a path set $P$ such that $N(P)$ is identical to $N(\alpha, f)$, where $P = \{v_1 - v_2$ path $p_1$, $v_2 - v_3$ path $p_2$, $v_2 - v_4$ path $p_3$, $v_4 - v_1$ path $p_4$, $v_4 - v_5$ path $p_5$, $v_5 - v_6$ path $p_6$, $v_5 - v_1$ path $p_7$, $v_6 - v_1$ path $p_8\}$. Since $M = \{v_2, v_4, v_6\}$, we have $P_M = \{p_1, p_3, p_6\}$, $P_U = \{p_2, p_5\}$ and $P_S = \{p_4, p_7, p_8\}$. Therefore, we have $G(P_M \cup P_U \cup \{p_4\})$ as the arborescence $T$ shown in Fig. 8 and a tree-type file transfer $D' = (\varphi', f')$ based on $T$ shown in Fig. 9,

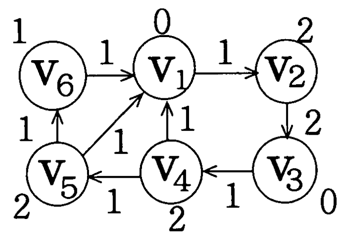


**Fig. 6** A file transfer $D = (\varphi, f)$.

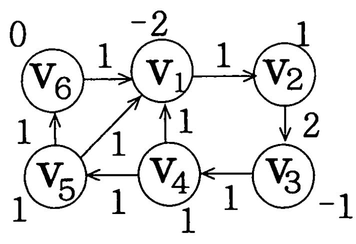


**Fig. 7** The net $N(\alpha, f)$.

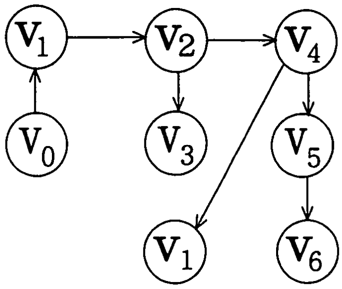


**Fig. 8** The arborescence $T$.

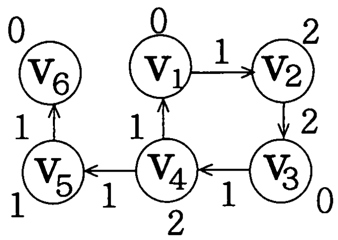


**Fig. 9** A tree-type file transfer $D' = (\varphi', f')$.

where $z(D') = 2$. It is easy to see that $C(D) > C(D')$ because of $p_7$ and $p_8$.

## 4. Synthesis of an Optimal File Transfer

By Prop.1, in order to obtain an optimal file transfer, it suffices to consider tree-type file transfers with the minimum cost, which is discussed in this section.

Let $E(G)$ be the set of all edges in an undirected graph $G$. Similarly to the directed case, the number of edges incident to a vertex $v$ is called the degree of $v$.

**Definition 5:** An undirected complete network with vertex set $U'$ is called the associated net, denoted by $AN$, where each edge $e = (x, y)$ has the weight $w(e) = c_V(x) + c_{x,y} + c_V(y)$. For an undirected tree $T$ on $AN$, let $w(T) = \sum_{e \in E(T)} w(e)$. An undirected network with the same structure as $AN$, where each edge $e = (x, y)$ has the weight $w_0(e)$ defined as

$$w_0(e) = \begin{cases} w(x,y) - c_V(v_1) & (v_1 \notin \{x, y\}), \\ c_V(v_1) + c_{v_1,y} & (\text{otherwise}, \ x = v_1), \end{cases}$$

is called the revised associated net, denoted by $AN_0$. For an undirected tree $T$ on $AN_0$, let $w_0(T) = \sum_{e \in E(T)} w_0(e)$. □

Note here that $w(v_1, v_1) = 2c_V(v_1)$ and $w_0(v_1, v_1) = c_V(v_1)$. The above $AN$ is a slight modification of the associated net in [6].

**Proposition 2:** Let $T$ be an undirected tree on $AN$ and let $T'$ be an arborescence obtained from $T$ by giving direction so that one $v_1$ is the root with degree $k$. Let $D$ be a tree-type file transfer based on $T'$. Then, we have

$$C(D) = w(T) - \min\{k, s_1\} \cdot c_V(v_1) - \sum_{u \in U} c_V(u), \tag{6}$$

$$C(D) = w_0(T) + \max\{0, k - s_1\} \cdot c_V(v_1) - \sum_{u \in U} c_V(u). \tag{7}$$

(Proof) Since the above arborescence $T'$ satisfies $\{y \in V | (x, y) \in T'\} = U$, $w(T) = \sum_{(x,y) \in A(T')} \{c_V(x) + c_{x,y}\} + \sum_{y \in U} c_V(y)$ holds. Therefore, by Lemma 5, we have Eq. (6). It is easy to prove Eq. (7) because $w(T) = w_0(T) + k \cdot c_V(v_1)$. □

Since the third term is constant in both Eqs. (6) and (7), the cost of the above $D$ is minimum if and only if the sum of the first two terms is minimum. Therefore, we first consider a minimum spanning tree with a degree constraint.

For such spanning trees on $AN$ and $AN_0$, we have the next lemma.

**Lemma 7:** Let *opt* and *opt'* be the degrees of root $v_1$ in a minimum spanning tree on $AN$ and $AN_0$, respectively. Then, we have $opt \leq opt'$.

(Proof) Suppose that $opt' < opt$. Let $T$ and $T'$ be minimum spanning trees on $AN$ and $AN_0$, respectively. Then

$$w_0(T) + c_V(v_1) \cdot opt = w(T),$$
$$w_0(T') + c_V(v_1) \cdot opt' = w(T').$$

Since $T'$ is optimal on $AN_0$, $w_0(T) \geq w_0(T')$ holds, which implies from the above inequalities that

$$w(T) = w_0(T) + c_V(v_1) \cdot opt > w_0(T') + c_V(v_1) \cdot opt' = w(T').$$

This contradicts that $T$ is optimal on $AN$. □

For a degree constraint minimum spanning tree, the next lemma holds

**Lemma 8**[10]**:** The weight of minimum spanning trees with one vertex degree constraint is a concave function of its degree number. The time complexity of obtaining such a tree is $O(n^2)$, where $n$ is the number of vertices. □

From this lemma, it is easy to see that the next lemma holds.

**Lemma 9:** Let $T_k$ be a minimum spanning tree on $AN$ such that the degree of root $v_1$ is $k$. Let *opt* be the degree of such $v_1$ in a minimum spanning tree on $AN$. Then, we have

$$w(T_1) \geq w(T_2) \geq \cdots \geq w(T_{opt}),$$
$$w(T_{opt}) \leq w(T_{opt+1}) \leq \cdots \leq w(T_{|U|}).$$

If we replace $AN$ and $w$ by $AN_0$ and $w_0$, respectively, then we have similar inequalities for the revised associated net. □

In the following, let $T_k$ be a minimum spanning tree such that the degree of root $v_1$ is $k$ on $AN$ or $AN_0$ and let $D_k$ be a tree-type file transfer obtained from $T_k$ by Definition 3.

Using the above lemma, we have the next proposition.

**Proposition 3:** Let $T$ and $T'$ be minimum spanning trees in $AN$ and $AN_0$, respectively, and let *opt* and *opt'* be the degrees of root $v_1$ in $T$ and $T'$, respectively. Consequently, if $s_1 \leq opt$, then $D_{opt}$ is an optimal file transfer and if $s_1 \geq opt'$, then $D_{opt'}$ is an optimal file transfer.

(Proof) First, we prove the case of $s_1 \leq opt$. For $k \in Z^+$, with $k \leq s_1$, we can see from Eq. (6) that

$$C(D_k) = w(T_k) - k \cdot c_V(v_1) - \sum_{u \in U} c_V(u).$$

Therefore, we can see from Lemma 9 that

$$C(D_1) > C(D_2) > \cdots > C(D_{s_1}) \tag{8}$$

because $k \leq s_1 \leq opt$. Next, for $k' \in Z^+$, with $s_1 \leq k' \leq |U|$, we can see from Eq. (6) that

$$C(D_{k'}) = w(T_k) - s_1 \cdot c_V(v_1) - \sum_{u \in U} c_V(u),$$

which implies from Lemma 9 that

$$C(D_{opt}) \leq C(D_{k'}) \quad (s_1 \leq k' \leq |U|)\,. \tag{9}$$

We can see from Eqs. (8) and (9) that $D_{opt}$ is an optimal file transfer.

Similarly, in the case of $s_1 \geq opt'$, we can prove that $D_{opt'}$ is an optimal file transfer from Eq. (7) and Lemma 9. □

In addition, we have the next proposition.

**Proposition 4:** Let *opt* and *opt'* be the degrees of root $v_1$ in a minimum spanning tree on $AN$ and $AN_0$, respectively. If $opt < s_1 < opt'$, then $Ds_1$ is an optimal file transfer.

(Proof) It should be noted from Lemma 7 that $opt \leq opt'$. For $k \in Z^+$, with $k \leq s_1$, it is easy to see from Eq. (7) that we have

$$C(D_k) = w_0(T_k) - \sum_{u \in U} c_V(u).$$

It follows from $k \leq s_1 < opt'$ and Lemma 9 that $w_0(T_k) \geq w_0(T_{s_1})$, which implies

$$C(D_k) \geq C(Ds_1) \qquad (1 \leq k \leq s_1)\,.$$

Next, for $k' \in Z^+$ with $s_1 \leq k' \leq |U|$, we can see from Eq. (6) that

$$C(D_k') = w(T_{k'}) - s_1 \cdot c_V(v_1) - \sum_{u \in U} c_V(u).$$

Since $opt < s_1 \leq k'$ and Lemma 9, we have

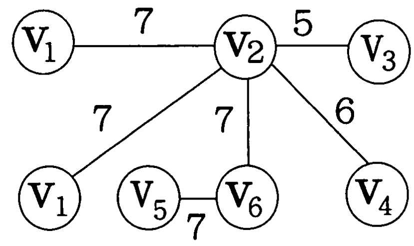


**Fig. 10** A minimum spanning tree on $AN$.

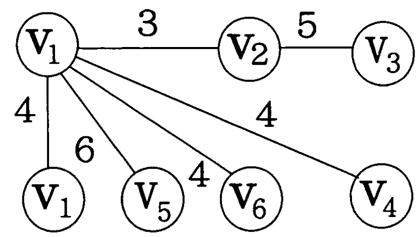


**Fig. 11** A minimum spanning tree on $AN_0$.

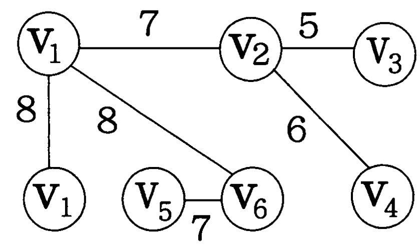


**Fig. 12** A minimum spanning tree with degree constraint.

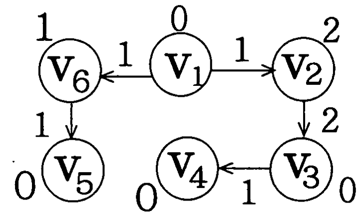


**Fig. 13** An optimal file transfer $D_5 = (\varphi_5, f_5)$.

$w(Ts_1) \leq w(T_{k'})$. Thus, we have

$$C(D_{s_1}) \leq C(D_{k'}) \qquad (s_1 \leq k' \leq |U|)\,.$$

Hence we prove Proposition 4. □

Here is an example of the above propositions. Minimum spanning trees on nets $AN$ and $AN_0$ of $N$ in Fig. 1 are shown in Fig. 10 and Fig. 11, respectively. The numbers attached to arc $e$ in Fig. 10 (resp., Fig. 11) indicates its weight in $AN$ (resp., $AN_0$). Since $opt = 1 < s_1 = 3 < 5 = opt'$ for the degrees of root $v_1$ in those trees, we can apply Proposition 4. Therefore, based on a minimum spanning tree whose degree of root $v_1$ is 3 on $AN$ shown in Fig. 12, we have an optimal file transfer $D_5 = (\varphi_5,\ f_5)$ of $N$ shown in Fig. 13 with its cost

$$\begin{aligned} C(D_5) &= 2 \cdot 1 + 1 \cdot 2 + 2 \cdot 1 + 1 \cdot 2 + 2 \cdot 1 + 2 \cdot 1 + 1 \cdot 1 \\ &= 13. \end{aligned}$$

To summarize the above consideration, we present an algorithm for synthesizing an optimal file transfer of $N$ as follows.

*Algorithm*
(Input) A file transmission net $N$
(Output) An optimal file transfer of $N$

Step 1. Find a minimum cost path and its cost among all vertices in $U$.

Step 2. Make the associated net $AN$. Find a minimum spanning tree $T$ on $AN$ and let *opt* be the degree of root $v_1$ on $T$.

Step 3. Make the revised associated net $AN_0$. Find a minimum spanning tree $T'$ on $AN_0$ and let *opt'* be the degree of root $v_1$ on $T'$.

Step 4. If $s_1 \leq opt$, then output a tree-type file transfer based on $T$ and terminate. Otherwise if $s_1 \geq opt'$, then output a tree-type file transfer based on $T'$ and terminate.

Step 5. Find a minimum spanning tree $T''$ whose degree of root $v_1$ is $s_1$ in $AN$. Output a tree-type file transfer based on $T''$ and terminate. □

Let $n = |V|$ and we analyze the above algorithm. It is easy to see that the worst case for the algorithm is that $N$ has a complete graph structure satisfying $U = V$. In Step 1, we solve all pair shortest-path problems, which take $O(n^3)$. In Step 2 and 3, we solve minimum spanning tree problems for dense $AN$ and $AN_0$, which take $O(n^2)$. Considering outputting a file transfer based on an arborescence, Step 4 takes $O(n^2)$. Finally Step 5 takes $O(n^2)$ from Lemma 8 including such output. Hence, the total time complexity of the algorithm is $O(n^3)$, where Step 1 is crucial. In addition, the space of the algorithm is $O(n^2)$.

## 5. Conclusion

We have shown that we can obtain, in polynomial time, an optimal file transfer of a file transmission net $N$ with at most one copy demand at each vertex and with a single source to which a sufficient number of copies of some information file are given.

A future task is to relax this condition for $N$. For instance, for $N$ with a vertex whose copy demand is more than one, we consider how to obtain an optimal file transfer in polynomial time. We also are interested in approaching graph-structure conditions, instead of the present condition of $M \cup \{v_1\} \subseteq U$ for parameters $s_V$, $c_V$, $d$, and $c_A$.

## Acknowledgements

The authors would like to thank the reviewers for their helpful and constructive comments.

## References

[1] H. Choi and S. Hakimi, "Data transfers in networks with transceivers," Networks, vol.17, pp.393–421, 1987.
[2] I. Rivera-Vega, R. Varadarajan, and S.B. Navathe, "Scheduling file transfers in fully connected networks," Networks, vol.22, pp.563–588, 1992.
[3] R. Casey, "Allocation of copies of file in an information network," AFIPS Conf. Proc, 1972 Spring Joint Comput, Conf. 40, pp.617–625, 1972.
[4] A. Segall, "Dynamic file assignment in a computer network," IEEE Trans. Automat. Control., vol.AC-21, no.2, pp.161–173, April 1976.
[5] Y. Kaneko and S. Shinoda, "The complexity of an optimal file transfer problem," IEICE Trans. Fundamentals, vol.E82-A. no.2, pp.394–397, Feb. 1999.
[6] Y. Kaneko, S. Shinoda, and K. Horiuchi, "A synthesis of an optimal file transfer on a file transmission net," IEICE Trans. Fundamentals, vol.E76-A. no.3, pp.377–386, March 1993.
[7] Y. Kaneko, K. Suzuki, S. Shinoda, and K. Horiuchi, "A synthesis of a forest-type optimal file transfer on a file transmission net with source vertices," IEICE Trans. Fundamentals, vol.E78-A, no.6, pp.671–679, June 1995.
[8] Y. Kaneko and S. Shinoda, "On the optimality of forest-type file transfers on a file transmission net," IEICE Trans. Fundamentals, vol.E83-A, no.7, pp.1411–1419, July 2000.
[9] J. Gross and J. Yellen, Graph Theory and Its Applications, CRC Press, 1998.
[10] H. Gabow, "A good algorithm for smallest spanning trees with a degree constraint," Networks, vol.8, pp.201–208, 1978.
[11] M.R. Garey and D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, W.H. Freeman, New York, 1979.
[12] M. Iri, I. Shirakawa, Y. Kajitani, and S. Shinoda, Graph Theory with Exercise, Corona Co. Ltd., 1983.

**Yoshihiro Kaneko** was born on June 23, 1964, in Nagoya, Japan. He received B.E., M.E., and D.E. degrees, all in electronics and communication engineering, from Waseda University, Tokyo, Japan, in 1988, 1990 and 1993, respectively. From 1992 to 1994, he was a Research Assistant at the School of Science and Engineering, Waseda University. Since 1994, he has been an Assistant professor at the Department of Electronics and Computer Engineering, Gifu University, Gifu, Japan. From 2000 to 2001, he was a visiting faculty of Florida Atlantic University. His research interest is in graph theory and its application. Dr. Kaneko is a member of Information Processing Society of Japan, Operations Research Society of Japan, and the Institute of Combinatorics and its Applications.

**Shoji Shinoda** received B.E., M.E., and D.E. degrees, all in electrical engineering, from Chuo University, Tokyo, Japan in 1964, 1966 and 1973, respectively. He is currently a Professor of electrical, electronic and communication engineering at Chuo University, Tokyo, Japan. His current research interests are various problems especially related to the topological aspects of circuits, networks, and systems including mobile communication systems. He is the author of Introduction of Circuit Theory (vols.(1) and (2);1996), and a coauthor of the books: Fundamentals of Circuit Theory (revised cd.,1997), Linear Algebra (1997), and Graph Theory with Exercises (1983), which were all published from Corona Publishing Company, Tokyo, Japan. He has published many papers in the areas of circuits, networks, and systems including mobile communication systems. Prof. Shinoda holds two Japanese patents. He is also the recipient of several awards and prizes, including the Best Paper Awards in 1992, 1997 and 1998 from the IEICE, the Best Paper Awards in 1996 from the Conference Committee of the IEEE International Conference on Neural Networks and Signal Processing (Nanjing, China), and Chuo University Prize Awards for Distinguished Research Promotion in 1997, 1998, 1999 and 2002. He served as the General Secretary of the 1984 Conference on Industrial Electronics, Control and Instrumentation (IECON'84), the Publicity Chair of the 1985 International Symposium on Circuits and Systems (ISCAS'85), the Vice General Chair and Technical program Chair of the 25th SICE Conference in 1986, Editor of Transactions of the IEICE (Japanese version, A), from May 1987 to May 1989, the Chair of the IEICE TG on Circuits and systems from May 1989 to May 1990, the Editor of the IEICE Transactions on fundamentals of Electronics, Communications and Computer Sciences from May 1990 to Dec. 1992, the Chair of the Student Activities Committee of the IEEE Region 10 (Asia·Pacific Region) from 1991 to 1992, the Chair of the Member Development Committee of the IEEE Region 10 (Asia·Pacific Region) from 1993 to 1994, the President of the IEICE Research Group on Engineering Sciences from May 1993 to May 1994, a Deputy President of the IEICE Society on Engineering Sciences from May 1995 to May 1996, the Chair of the IEICE TG on Multi-dimensional Mobile Information Networks (MuMIN) from May 1995 to July 1996, the Chair of the Research Society on Multi-dimensional Mobile Information Networks (MuMIN) from July 1996 to May 1997, the Chair of the Editorial Board of IEICE Engineering Sciences Society from May 1996 to May 1998, the General Chair of the 1999 International Symposium on Nonlinear Theory and Its Applications (NOLTA'99, Big Island, Hawaii), and the General Chair of the 2002 International Conference on Fundamentals of Electronics, Communications and Computer Sciences (ICFS 2002). Also, he served as a member of the IEICE Council from Currently, he is a member of the IEICE Board of Directors, the Chairperson of the IEICE Editorial Board, a Chair of the IEICE Committee on Technology and History, the Deputy Chair of the IEICE Committee on Accreditation Plicy Council, a member of the IEICE Strategic Planning Committee, and the President of the Japan Society for Simulation Technology. He is a Fellow of IEEE and a member of IEEK.