PAPER *Special Section of Selected Papers from the 10th Karuizawa Workshop on Circuits and Systems*

Covering Problems in the p-Collection Problems

Kaoru WATANABE[†], **Masakazu SENGOKU**[††], **Hiroshi TAMURA**[†††], ***and*** **Shoji SHINODA**[††††], ***Members***

SUMMARY The lower-bounded p-collection problem is the problem where to locate p sinks in a flow network with lower bounds such that the value of a maximum flow is maximum. This paper discusses the cover problems corresponding to the lower bounded p-collection problem. We consider the complexity of the cover problem, and we show polynomial time algorithms for its subproblems in a network with tree structure.

key words: *location problem, network flows, NP-complete, optimization problem*

1. Introduction

Location problems are an important in a lot of fields, applied mathematics, computer science, operations research, management science and industrial engineering (see the papers [1], [2], or the book [3]). The p-collection problem is the problem where to locate p sinks in a flow network such that the value of a maximum flow is maximum [6]. This problem is an important location problem in a flow network because one can apply to locating p resources (e.g. data bases, file-servers, etc.) in a computer network such that as many terminals (clients) can utilize these resources as possible. The paper [4] deal with cover problems in flow networks. This paper discusses new cover problems, which correspond to the p-collection problems.

Let $D = (V, A)$ be the digraph with a vertex set V and an arc set A such that $(v, u) \notin A$ for any arc (u, v) of A. Let b^- and b^+ be functions: $V \to Z$ (the set of integers) such that $b^-(v) \leqq b^+(v)$ for any v of V, let c^- and c^+ be functions: $A \to Z$ such that $c^-(a) \leqq c^+(a)$ for any a of A, and let d^- and d^+ be functions: $V \to Z \cup \{\infty\}$ such that $d^-(v) \leqq d^+(v)$ for any v of V. The functions $b^+(v)$ and $b^-(v)$ imply respectively an upper and an lower bounds of flows from the source to v. The functions $c^+(a)$ and $c^-(a)$ correspond to respectively the capacity and the lower bound of a. The functions $d^+(v)$ and $d^-(v)$ mean respectively an upper and an lower bounds of flows from v to the sink. We call a 7-tuple:

$$N = (D, b^-, b^+, c^-, c^+, d^-, d^+)$$

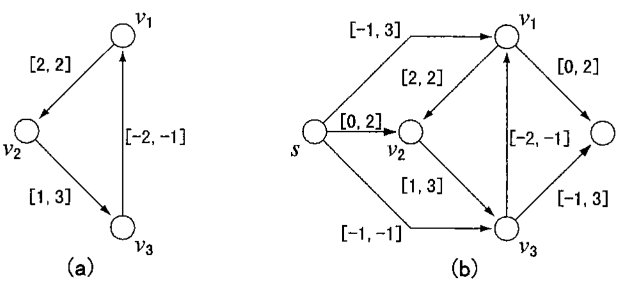
a *network with lower bounds* (usually, described as $N = (D, b^\pm, c^\pm, d^\pm)$, and called a network, for convenience). Figure 1 (a) illustrates an example of a network with lower bounds. Let s and t be new specified vertices called the source and the sink, respectively. We represent V^*, A^s and A^t as the vertex set $V \cup \{s, t\}$, the arc sets $\{(s, v) : v \in V\}$ and $\{(v, t) : v \in V\}$ respectively. Let X be an arbitrary subset of V, let $A^t_X = \{(x, t) : x \in X\}$, and let $A^*_X = A \cup A^s \cup A^t_X$. We define functions e^- and e^+: $A^*_V \to Z \cup \{\infty\}$ as

$$e^\pm(u, v) = \begin{cases} b^\pm(v) & \text{if } u = s \\ d^\pm(u) & \text{if } v = t \\ c^\pm(u, v) & \text{otherwise.} \end{cases}$$

(In this paper, double signs in an equation are in same order.) And we define the adjoint network N_X of N with respect to a subset X of V as the s-t flow network with lower bounds:

$$N_X = (D_X, s, t, e^+|A^*_X, e^-|A^*_X),$$

where $D_X = (V^*, A^*_X)$, and where $e^-|A^*_X$ (respectively, $e^+|A^*_X$) denotes the restrictions of e^- (e^+ respectively) on A^*_X. The adjoint network N_X is *feasible* if there exists a flow f in A_X meeting the following conditions.
Capacity Constrain: For any $a \in A^*_X$,



$b^-(v_1) = -1,\quad b^+(v_1) = 3,\quad d^-(v_1) = 0,\quad d^+(v_1) = 2$
$b^-(v_2) = 0,\quad b^+(v_2) = 2,\quad d^-(v_2) = 1,\quad d^+(v_2) = 3$
$b^-(v_3) = -1,\quad b^+(v_3) = -1,\quad d^-(v_3) = -1,\quad d^+(v_3) = 3.$

Fig. 1 (a) A network N, (b) the adjoint network of N with respect to $X = \{v_1, v_3\}$.

Manuscript received June 30, 1997.
Manuscript revised September 12, 1997.
[†]The author is with Osaka Electro-Communication University, Neyagawa-shi, 572–8530 Japan.
[††]The author is with Niigata University, Niigata-shi, 950–2181 Japan.
[†††]The author is with Niigata Institute of Technology, Kashiwazaki-shi, 945–1195 Japan.
[††††]The author is with Chuo University, Tokyo, 112–8551 Japan.

$$e^-(a) \leqq f(a) \leqq e^+(a).$$

Flow Conservation: The flow f is conserved in any vertex v of V in N_X. That is,

$$\sum_{u \in \mathrm{ad}(v)} f(v, u) = 0,$$

where $\mathrm{ad}(v)$ is the set of vertices adjacent to-or-from v in N_X, and where $f(v, u) = -f(u, v)$ for any (u, v) of A_X^*.

A flow has skew symmetry, which is already used in the condition of Flow Conservation. That is, we can assume that, if (u, v) is an arc of A_X^*, then $f(v, u) = -f(u, v)$. For any (u, v) of A_X^*, we can also suppose that $c^\pm(v, u) = -c^\mp(u, v)$, and $e^\pm(v, u) = -e^\mp(u, v)$, since the capacity constrain holds for the pair (v, u).

If N_X is feasible, the value of f is $\mathrm{val}(f) = \sum_{a \in A_X^t} f(a)$ for any flow f in N_X. Given a network N, we define $h_N(X)$ as

$$h_N(X) = \begin{cases} \max_f \mathrm{val}(f) & \text{if } N_X \text{ is feasible} \\ -\infty & \text{otherwise.} \end{cases}$$

(Notice that $\max_f \mathrm{val}(f)$ is the value of a maximum flow in N_X.) If p is a positive integer with $p \leqq |V|$, let $H_N(p) = \max\{h_N(X) : |X| = p\}$. (We usually omit the subscripts N of h_N and H_N when N is clear from context.) A subset X^* of V with $|X^*| = p$ is a *maximum p-collection set* of N if $h(X^*) = H(p)$. We call the optimization problem of searching a maximum p-collection set of a network the *lower-bounded p-collection problem*, and we write LBC as this problem. For example let's solve LBC for the network illustrated in Fig. 1 (a). We have

$$h(v_1, v_2) = -\infty,\ h(v_2, v_3) = 4,\ h(v_3, v_1) = 3.$$

Thus $H(2) = 4$, and $\{v_2, v_3\}$ is the maximum 2-collection set.

The p-collection problem discussed in [6] is equivalent to the subproblem of LBC such that

$$\begin{array}{ll} b^-(v) = 0 & \text{for any } v \text{ of } V, \\ c^-(a) \leqq 0 \leqq c^+(a) & \text{for any } a \text{ of } A, \\ d^-(v) = 0 \text{ and } d^+(v) = \infty & \text{for any } v \text{ of } V. \end{array}$$

In this paper we describe this problem to PRC, the primary p-collection problem.

The sink-capacitated p-collection problem discussed in [7] is equivalent to the subproblem of LBC such that

$$\begin{array}{ll} b^-(v) = 0 & \text{for any } v \text{ of } V, \\ c^-(a) \leqq 0 \leqq c^+(a) & \text{for any } a \text{ of } A, \\ d^-(v) = 0 & \text{for any } v \text{ of } V. \end{array}$$

In this paper we describe this problem to SCC.

We introduce description $\alpha/\beta/\gamma$ for subproblems of the p-collection problem. The first item α denotes is the value of p. If $\alpha = p$, suppose that p is arbitrarily fixed. The second item β indicates a network topology. This paper deals with two topology types: general network ($\beta = N$) and network with tree structure ($\beta = T$) where a network with tree structure is a network whose underlying graph is a tree. The last item γ means a subproblem (PRC, SCC or LBC) on which $b^\pm$, $c^\pm$ and $d^\pm$ are restricted. For example, the problem $p/T/$LBC denotes the lower-bounded p-collection problem in a network with tree structure.

A subset X of V is a *cover* of N if $h(X) = \sum_{v \in V} b^+(v)$. (Notice that there is not always a cover.) The cover problem corresponding to LBC is the problem of finding a minimum cover of N. We refer C$/\beta/\gamma$ as the cover problem corresponding to $p/\beta/\gamma$. In the network illustrated in Fig. 1 (a), the subset $\{v_2, v_3\}$ is a minimum cover, since each subset with cardinality 1 is not a cover.

In the paper [6] the authors presented an $O(n)$ algorithm for $1/T/$PRC, and an $O(p^2n^3C^2)$ algorithm for $p/T/$PRC, where $n = |V|$ and C denotes the maximum of weight and capacity. Tsukiyama [5] proposed an $O(p^2n^2C^2)$ algorithm for $p/T/$PRC. The paper [8] contains some complexity results that $p/N/$PRC is strongly NP-hard, and that $p/T/$PRC is weakly NP-hard. The paper [9] proves the weak NP-hardness of $p/T/$LBC, and presents an $O(n)$ algorithm for $1/T/$LBC and an $O(p^2n^2C^2)$ algorithm for $p/T/$LBC.

This paper discusses the cover problems. In Sect. 2 we obtain complexity results for cover problems. Section 3 presents an $O(n)$ algorithm for C$/T/$PRC, and Sect. 4 presents an $O(n^3)$ algorithm for C$/T/$SCC with dynamic programming type. Hence we know that, although $p/T/$PRC and $p/T/$SCC are NP-hard, the corresponding cover problems are solvable in polynomial time.

2. Complexity Results

In this section we shall obtain complexity results for cover problems. The paper [8] shows strong NP-hardness for $p/N/$PRC by polynomial reduction from the vertex cover problem to this problem. The reduced instance in there is also the instance of C$/N/$PRC. Hence C$/N/$PRC is strong NP-hard. Hence we obtain the following theorem.

Theorem 1: The problem C$/N/$PRC, C$/N/$SCC and C$/N/$LBC are strongly NP-hard.

We can know weak NP-hardness for C$/T/$LBC in much the same way as the proof for $p/T/$LBC in the paper [9]. Thus we obtain the following theorem.

Theorem 2: The problem C$/T/$LBC is weakly NP-hard.

In Sects. 3 and 4, we shall design polynomial time algorithms for C$/T/$PRC and C$/T/$SCC, respectively. Thus, although $p/T/$PRC and $p/T/$SCC are NP-hard, the following theorem holds for the corresponding cover problems.

Table 1 Complexity results for cover problems.

Topology type	PRC	SCC	LBC
Network	Strongly NPH	Strongly NPH	Strongly NPH
Tree	P	P	Weakly NPH

Theorem 3: The covering problems C/T/PRC and C/T/SCC belong to the class P.

The obtained results so far are brought together in Table 1.

3. The Problem C/T/PRC

Let $N = (T, b^{\pm}, c^{\pm}, d^{\pm})$ be a network with tree structure, where $T = (V, A)$. For a cover problem, the function $b^{-}(\cdot)$ has few meaning. For convenience, we assign $b^{-}(v)$ to $-\infty$ for any v of V. We present Algorithm 1 with complexity $O(n)$ for C/T/PRC, and we prove Theorem 4.

Theorem 4: Algorithm 1 solves C/T/PRC exactly.

Proof: Suppose that the algorithm returns a vertex set X for a network $N = (T, b^{\pm}, c^{\pm}, d^{\pm})$ where $T = (V, A)$. Let v_0 denote the vertex indicated by u in Step 1.20, let $B(u)$ denote the value of $b^{+}(u)$ in Step 1.6, and let $B(v_0)$ denote the value of $b^{+}(u)$ in Step 1.21. If v is a vertex of $V - \{v_0\}$, then $n(v)$ denotes the vertex adjacent to v and nearest to v_0 in N, and then $U(v)$ denotes the set of vertices adjacent to-or-from v excluding $n(v)$. And let $U(v_0)$ denote the set of vertices adjacent to-or-from v_0. To begin with, we show that X is a cover of N. We define the function f on A^{*}_{X} as follows. For any (s, v) of A^{s}, let $f(s, v) = b^{+}(v)$. For any u of $V - \{v_0\}$, let $f(u, v)$ be the value of D in Step 1.17. Then f satisfies the capacity constrain in $A^{s} \cup A$. We correct f to be flow-conserved in each vertex of $V - X$.

Consider the case where $v_0 \notin X$. Then $B(v_0) = \sum_{u \in U(v_0)} f(u, v_0) + b^{+}(v_0) \leqq 0$. Since $b^{+}(v_0) \geqq 0$, we can increase some $f(u, v_0)$s with negative value without exceeding 0 for f to be flow-conserved on v_0. If $v_0 \in X$, then we do not need to have above the consideration. However, if x is a vertex of X, then let $f(u, x) = 0$ for any u of $U(x)$ such that $f(u, x) < 0$. We call $f(u, v)$ corrected if $v \in X \cup \{v_0\}$, and if $u \in U(v)$. Let v be a vertex of $V - \{v_0\}$. If $f(v, n(v))$ is corrected, and if $f(u, v)$ is not corrected for any u of $U(v)$, then we perform the following operation, hoping for f to be flow-conserved on v.

Step 1. If $B(v) < 0$, then increase some $f(u, v)$s with negative value without exceeding 0 to meet the flow conservation on v.

Step 2. Let $f(u, v)$ be corrected for any u of $U(v)$. □

If $B(v) \geqq 0$, then f is flow-conserved on v. Hence Step 1 corrects f if $B(v) < 0$. Thus

$$B(v) = \sum_{u \in U(v)} f(u, v) + b^{+}(v) < 0.$$

Algorithm 1

```
1.1  begin
1.2    X ← ∅;
1.3    while |V(N)| > 1 do
1.4      u ← an arbitrary leaf of N;
1.5      v ← the vertex of N adjacent to-or-from u;
1.6      if b+(u) ≧ 0 then
1.7        if b+(u) ≦ c+(u,v) then
1.8          D ← b+(u);
1.9        else
1.10         b+(u) ← −∞;
1.11         X ← X ∪ {u};
1.12       fi;
1.13     fi;
1.14     if b+(u) < 0 then
1.15       D ← max{b+(u), c−(u,v)};
1.16     fi;
1.17     b+(v) ← b+(v) + D;
1.18     remove u from N;
1.19   od;
1.20   u ← the vertex of N;
1.21   if b+(u) > 0 then X ← X ∪ {u} fi;
1.22   return(X);
1.23 end.
```

Since $b^{+}(v) \geqq 0$, we can always perform Step 1. Hence this operation changes f to be flow-conserved on v. We repeat the preceding operation, until $f(u, v)$ is corrected for any (u, v) of A. After the iteration completes, the function f meets the capacity constrain in $A^{s} \cup A$ and the flow conservation in $V - X$. We finally define

$$f(x, t) = \sum_{v \in \mathrm{ad}(x)} f(v, x) + b^{+}(x),$$

and the function f has changed to a flow in N_X such that $f(s, v) = b^{+}(v)$ for any (s, v) of A^{s}. Hence X covers N.

There remains to verify whether X is minimum. If we remove the arc $(v, n(v))$ from T for any v of $X - \{v_0\}$, then we obtain the disconnected digraph. This disconnected digraph consists of the connected components each with either v_0 or a vertex of X. Let x be a vertex of $X \cup \{v_0\}$, let T_x be the component with x, and let $V_x = V(T_x)$. Now let W be a subset of $V - V_x$, and assume that W is a cover of N. Then there exists a maximum flow g in N_W such that $g(s, v) = b^{+}(v)$ for any v of V. For any v of V_x, the following inequality must hold.

$$B(v) \leqq \sum_{u \in U(v)} g(u, v) + g(s, v).$$

Consider the case where $x = v_0$, and where $x \in X$. Then

$$\sum_{u \in U(x)} g(u,x) + g(s,x) > 0.$$

Since $x \notin W$, the flow g is not conserved in x. This result contradicts that g is a flow. Hence W is not cover. Now consider the case where $x \in X - \{v_0\}$. Since $c^+(x, n(x)) < B(x)$, we obtain

$$c^+(x, n(x)) < g(x, n(x)).$$

This inequality contradicts the capacity constrain. Hence W is not a cover of N. By the above two considerations, a cover of N must contain a vertex of V_x for any x of X. Hence $|X| \leqq |X^*|$ where X^* denotes a minimum cover of N. We know X is a cover of N, and so we have $|X^*| = |X|$. Consequently X is a minimum cover. □

4. The Problem $C/T/$SCC

In this section we present an algorithm with dynamic programming type for C/T/SCC. Given a network, let v_1 be an arbitrary vertex of it. For convenience, we add to the given network a new vertex v_0 and a new arc (v_1, v_0) such that $c^\pm(v_1, v_0) = \pm\infty$. Let $N = (T, b^\pm, c^\pm, d^\pm)$ denote the new obtained network. We suppose that v_0 is the root, and regard N as rooted. For any vertex v of N, we define the level lev(v) of v as the number of all the arcs in a v-v_0 path between v and v_0. Let $L(l) = \{v \in V : \mathrm{lev}(v) = l\}$ for any nonnegative l, and let l^* denote the maximum integer such that $L(l^*) \neq \emptyset$. If v is a vertex of $V - \{v_0\}$, then $p(v)$ denotes the parent of v, the vertex adjacent to v and nearest to v_0 in N, and then Ch(v) denotes the set of all the children, all the vertices adjacent to-or-from v excluding $p(v)$. Let v be a vertex of T, and let U' be a subset of Ch(v). If we remove the arcs of $\{(v, p(v))\} \cup \{(u, v) : u \in \mathrm{Ch}(v) - U'\}$ from T, then we obtain the subtree $T^v_{U'}$ involving v. Fixing U', let $W = V(T^v_{U'}) - \{v\}$, and let $N^v_{U'}$ denotes the restriction of N on $T^v_{U'}$. Let k be an integer such that $0 \leqq k \leqq |W|$, let X be a subset of W with $|X| = k$, and let $(N^v_{U'})_X$ be the restriction of the adjoint network N_X on the digraph obtained by adding to $T^v_{U'}$ the source s, the sink t, the arc set $\{(s, w) : w \in W\}$, and $\{(x, t) : x \in X\}$. The subnetwork $N^v_{U'}$ is k-covered if there exists a function f_X on the arc set of $(N^v_{U'})_X$ for some X, satisfying the capacity constrain for any arc of $(N^v_{U'})_X$, and the flow conservation for any vertex of W, such that $f_X(s, w) = b^+(s, w)$ for any w of W. (Notice that flows does not always conserved on v.) We define

$$F^v_{U'}(k) = \begin{cases} \min\limits_{|X|=k} \min\limits_{f_X} \sum\limits_{u \in U'} f_X(u, v) & \text{if } N' \text{ is } k\text{-covered} \\ \infty & \text{otherwise.} \end{cases}$$

The value of $F^v_{U'}(k)$ means the minimum value of a flow on the cut from U' to v that covers all the vertices of W.

Let's examine the meaning of $F^{v_0}_{\{v_1\}}(k)$. There exist a subset X of W with $|X| = k$ and f_X on $(N^{v_0}_{\{v_1\}})_X$ such that $f_X(v_0, v_1) = 0$ if and only if there exists a cover with size k of the given network. From the definition of $F^v_{U'}(k)$, if $F^{v_0}_{\{v_1\}}(k) > 0$, then there is not such X and f_X; otherwise however there exist. Hence $F^{v_0}_{\{v_1\}}(k) \leqq 0$ if and only if there exists a cover with size k. Thus we shall find the minimum of k such that $F^{v_0}_{\{v_1\}}(k) \leqq 0$. We shall obtain an algorithm C/T/SCC by simplifying the algorithm for p/T/SCC in the paper [7]. That algorithm stores all feasible values of flows on each cut. In this case the considered algorithm stores only the maximum value of feasible flows. Thus C/T/SCC is solvable in polynomial time. Now study the properties of $F^v_{U'}(k)$ separating three cases.

Case 1: $|U'| = 1$, and $u \in U'$ is a leaf.
It is easy to see the following equation for $k = 0$.

$$F^v_{U'}(0) = \begin{cases} b^+(u) & \text{if } b^+(u) \leqq c^+(u, v) \\ \infty & \text{otherwise.} \end{cases}$$

And for $k = 1$,

$$F^v_{U'}(1) = \begin{cases} b^+(u) - d^+(u) & \text{if } b^+(u) - d^+(u) \leqq c^+(u, v) \\ \infty & \text{otherwise.} \end{cases}$$

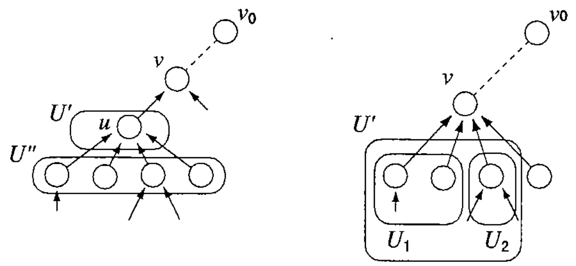
Case 2: $|U'| = 1$, and $u \in U'$ is not a leaf.
Let $U'' = \mathrm{Ch}(u)$, and furthermore consider two cases. Fig. 2 (a) shows state in this case.
Case 2.1: u is not adjacent to t, and $k < |W|$.
Let $F'_{U'}(k)$ denote the value of $F^v_{U'}(k)$ in this case. If $F^u_{U''}(k) + b^+(u) \leqq c^+(u, v)$, then we obtain

$$F'_{U'}(k) = \min\{F^u_{U''}(k) + b^+(u), c^-(u, v)\};$$

otherwise $F'_{U'}(k) = \infty$.
Case 2.2: u is adjacent to t, and $k \geqq 1$.
Let $F''_{U'}(k)$ denote the value of $F^v_{U'}(k)$ in this case. If $F^u_{U''}(k-1) + b^+(u) - d^+(u) \leqq c^+(u, v)$, then we obtain

$$F''_{U'}(k) = \min\{F^u_{U''}(k-1) + b^+(u) - d^+(u), c^-(u, v)\};$$

otherwise $F''_{U'}(k) = \infty$.



(a) (b)

Fig. 2 (a) Case 2, (b) Case 3.

Algorithm 2

2.1 begin
2.2 $\quad i \leftarrow l^* - 1$;
2.3 $\quad$ while $i \geqq 0$ do
2.4 $\quad\quad$ for all $v \in L(i)$ do
2.5 $\quad\quad\quad$ for all $u \in \mathrm{Ch}(v)$ do
2.6 $\quad\quad\quad\quad U' \leftarrow \{u\}$;
2.7 $\quad\quad\quad\quad$ if u is a leaf then
2.8 $\quad\quad\quad\quad\quad$ calculate $F^v_{U'}(0)$ and $F^v_{U'}(1)$; {Case 1}
2.9 $\quad\quad\quad\quad\quad X_{U'}(0) \leftarrow \emptyset$, and $X_{U'}(1) \leftarrow \{u\}$;
2.10 $\quad\quad\quad\quad$ else
2.11 $\quad\quad\quad\quad\quad m \leftarrow |V(T^v_{U'})| - 1$;
2.12 $\quad\quad\quad\quad\quad$ for all $0 \leqq k < m$ do calculate $F'_{U'}(k)$ od; {Case 2.1}
2.13 $\quad\quad\quad\quad\quad$ for all $1 \leqq k \leqq m$ do calculate $F''_{U'}(k)$ od; {Case 2.2}
2.14 $\quad\quad\quad\quad\quad F^v_{U'}(0) \leftarrow F'_{U'}(0)$, $X_{U'}(0) \leftarrow X_{\mathrm{Ch}(u)}(0)$, $F^v_{U'}(m) \leftarrow F''_{U'}(m)$, and $X_{U'}(m) \leftarrow X^u_{\mathrm{Ch}(u)}(k)$;
2.15 $\quad\quad\quad\quad\quad$ for all $1 \leqq k < m$ do
2.16 $\quad\quad\quad\quad\quad\quad$ if $F'_{U'}(k) \leqq F''_{U'}(k)$ then $F^v_{U'}(k) \leftarrow F'_{U'}(k)$, and $X_{U'}(k) \leftarrow X_{\mathrm{Ch}(u)}(k)$;
2.17 $\quad\quad\quad\quad\quad\quad$ else $F^v_{U'}(k) \leftarrow F''_{U'}(k)$, and $X_{U'}(k) \leftarrow X_{\mathrm{Ch}(u)}(k-1) \cup \{u\}$
2.18 $\quad\quad\quad\quad\quad\quad$ fi;
2.19 $\quad\quad\quad\quad\quad$ od;
2.20 $\quad\quad\quad\quad$ fi;
2.21 $\quad\quad\quad$ od;
2.22 $\quad\quad\quad U_1 \leftarrow$ the set of a vertex of $\mathrm{Ch}(v)$;
2.23 $\quad\quad\quad$ while $U_1 \neq \mathrm{Ch}(v)$ do
2.24 $\quad\quad\quad\quad U_2 \leftarrow$ the set of a vertex of $\mathrm{Ch}(v) - U_1$, and $U' \leftarrow U_1 \cup U_2$;
2.25 $\quad\quad\quad\quad$ for all $0 \leqq k < |V(T^v_{U'})|$ do
2.26 $\quad\quad\quad\quad\quad$ calculate $F^v_{U'}(k)$; {Case 3}
2.27 $\quad\quad\quad\quad\quad X_{U'}(k) \leftarrow X_{U_1}(k_1) + X_{U_2}(k_2)$; {where k_1, k_2 are integers such that $F^v_{U'}(k) = F^v_{U_1}(k_1) + F^v_{U_2}(k_2)$}
2.28 $\quad\quad\quad\quad$ od;
2.29 $\quad\quad\quad\quad U_1 \leftarrow U'$;
2.30 $\quad\quad\quad$ od;
2.31 $\quad\quad$ od;
2.32 $\quad\quad i \leftarrow i - 1$;
2.33 $\quad$ od;
2.34 $\quad k^* \leftarrow$ the minimum integer such that $F^{v_0}_{\{v_1\}}(k) \leqq 0$;
2.35 $\quad$ return$(X_{v_1}(k^*))$;
2.36 end.

Using the results in Case 2.1 and 2.2, we can obtain

$$F^v_{U'}(k) = \begin{cases} F'_{U'}(k) & \text{if } k = 0 \\ \min\{F'_{U'}(k), F''_{U'}(k)\} & \text{if } 0 < k < |W| \\ F''_{U'}(k) & \text{if } k = |W|. \end{cases}$$

We can compute the value of $F^v_{U'}(k)$ if we know the values of $F^u_{U''}(k)$ for all children u of v.

Case 3: $|U'| \geqq 2$.

Let U_1 and U_2 be non-empty subsets of U' such that $U_1 \cup U_2 = U'$, and such that $U_1 \cap U_2 = \emptyset$. Figure 2 (b) shows state in this case. Then

$$F^v_{U'}(k) = \min_{k_1,k_2}\{F^v_{U_1}(k_1) + F^v_{U_2}(k_2)\},$$

where k_1 and k_2 are integers such that $k_1 + k_2 = k$, such that $0 \leqq k_1 \leqq \min\{k, |V(T_{U_1})| - 1\}$, and such that $0 \leqq k_2 \leqq \min\{k, |V(T_{U_2})| - 1\}$. We can estimate the value of $F^v_{U'}(k)$ if we know the values of $F^v_{U_1}(k_1)$ and $F^v_{U_2}(k_2)$ for all k_1 and k_2. (In this algorithm we assume that $|U_2| = 1$.)

From the above results we obtain the recurring formulas (Case 2 and 3) and the boundary condition (Case 1) for dynamic programming. Suppose that we know the value of $F^u_{U''}(k)$ for any k and for any u of $\mathrm{Ch}(v)$. Then we can estimate the value of $F^v_{\{u\}}(k)$ for any k and for any u (Fig. 2 (a)). Next we can obtain the value of $F^v_{\mathrm{Ch}(v)}(k)$ for any k (Fig. 2 (b)). We can design Algorithm 2. First the algorithm calculate $F^v_{\mathrm{Ch}(v)}(k)$ for any $v \in L(l^*-1)$, next for any $v \in L(l^*-2)$, $\cdots$, and last for the root v_0.

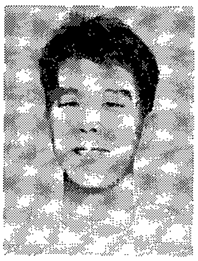
In algorithm 2, the step 2.5 is executed $O(n)$ times. It costs $O(n)$ time to calculate each $F'_{U'}(k)$, $F''_{U'}(k)$ and $F^v_{U'}(k)$. Since $0 \leqq k \leqq n$, the complexity of the algorithm is $O(n^3)$.

5. Conclusion

This paper deals with the cover problems corresponding to PRC, SCC and LBC. In Sect. 2, we obtain the complexity results for the covering problems. Section 3 presents an $O(n)$ algorithm for $C/T/$PRC, and Sect. 4 presents an $O(n^3)$ algorithm with dynamic programming type for $C/T/$SCC. Hence, although $p/T/$PRC and $p/T/$SCC are NP-hard, the corresponding cover problems, $C/T/PRC$ and $C/T/SCC$, are solvable in polynomial time. We hope to improve Algorithm 2.

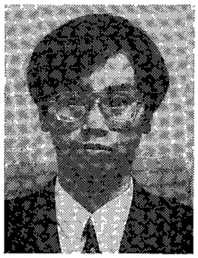
References

[1] O. Kariv and S.L. Hakimi, "An algorithmic approach to network location problems. I: The p-centers," SIAM J., APPL., MATH., vol.37, no.3, pp.515–538, Dec. 1979.
[2] O. Kariv and S.L. Hakimi, "An algorithmic approach to network location problems. II: The p-medians," SIAM J., APPL., MATH., vol.37, no.3, pp.539–560, Dec. 1979.
[3] P.B. Mirchandani and R.L. Francis, ed., "Discrete location theory," John Wiley & Sons, Inc., 1990.
[4] H. Tamura, M. Sengoku, S. Shinoda, and T. Abe, "Some covering problems in location theory on flow networks," IEICE Trans., Fundamentals, vol.E75-A, no.6, pp.678–684, June 1992.
[5] S. Tsukiyama, "A new algorithm for p-collection problem on a tree-type flow network," Proc. of ITC-CSCC, vol.2, pp.721–724, 1996.
[6] K. Watanabe, H. Tamura, and M. Sengoku, "Problems of where to locate p-sinks in a flow network," IEICE Trans., vol.J78-A, no.8, pp.938–946, 1995.
[7] K. Watanabe, H. Tamura, and M. Sengoku, "Problems of where to locate capacitated p sinks in a flow network," Proc. of JTC-CSCC, pp.758–761, 1995.
[8] K. Watanabe, H. Tamura, and M. Sengoku, "The problems of where to locate p-sinks in a flow network: Complexity approach," IEICE Trans. Fundamentals, vol.E79-A, no.9, pp.1495–1503, Sept. 1996.
[9] K. Watanabe, H. Tamura, and M. Sengoku, "The p-collection problem in a flow network with lower bounds," IEICE Trans., Fundamentals, vol.E80-A, no.4, pp.651–657, April 1997.



Kaoru Watanabe received the B.E., M.E. and Ph.D. degrees from Niigata University in 1990, 1992 and 1996, respectively. He is presently in Osaka Electro-Communication University. He received the Paper Award from IEICE in 1996. He is interested in graph theory and algorithm theory.

Masakazu Sengoku received the B.E. degree in electrical engineering from Niigata University, Niigata, Japan, 1967 and the M.E. and Ph.D. degrees from Hokkaido University in 1969 and 1972, respectively. In 1972, he joined the staff at Department of Electronic Engineering, Hokkaido University as a Research Associate. In 1978, he was an Associate Professor at Department of Information Engineering, Niigata University, where he is presently a Professor. His research interests include network theory, graph theory, transmission of information and mobile communications. He received the Paper Awards from IEICE in 1992, 1996 and 1997, and IEEE ICNNSP Best Paper Award in 1996. He was 1995 Chairman of IEICE Technical Group on Circuits and Systems (TG-CAS). He is a senior member of IEEE, a member of the Information Processing Society of Japan and Japan Society for Industrial and Applied Mathematics.



Hiroshi Tamura received the B.Educ., M.S. and Ph.D. degrees from Niigata University in 1982, 1986 and 1990, respectively. In 1990, he joined the staff at the Graduate School of Science and Technology, Niigata University as a Research Associate. He is presently an Associate Professor at Niigata Institute of Technology. His research interests are in graph theory and its application. He received the Paper Awards from IEICE in 1992 and 1996. He is a member of IEEE, IPS of Japan and the Mathematical Society of Japan.

Shoji Shinoda received the B.E., M.E. and D.E. degrees, all in electrical engineering, from Chuo University, Tokyo, Japan in 1964, 1966 and 1973, respectively. Since April 1965, he has been with the Faculty of Science and Engineering, Chuo University, Tokyo Japan where he is currently professor of the Department of Electrical and Electronics Engineering and the Dean of the Graduate School of Science and Engineering, Chuo University. He research interest has been in diagnosis, analysis and design of circuits, networks and discrete systems. He has published more than one hundred technical papers and has co-authored seven books and two handbooks. He received the IEICE Best Paper Awards in 1992 and 1997, and IEEE ICNNSP Best Paper Award in 1996. He is a member of IEEE, SICE (Society of Instrument and Control Engineering), KITE (Korean Institute of Telematics and Electronics) and JSST (Japan Society of Simulation Technology). He was 1989 Chairman of IEICE Technical Group on Circuits and Systems (TG-CAS), the 1991–1992 Editor of the IEICE Trans. on Fundamentals, and the 1993 President of the IEICE Engineering Science Group (ESG) consisting of 13 Technical Groups such as TG-CAS, TG-NLP, TG-VLD and TG-CST. Also, he was the 1991–1992 SAC Chairman and the 1993–1994 MDC Chairman of the IEEE Region 10 (Asia-Pacific Region). He is now the Chairman of the Editorial Board of the IEICE Engineering Science Society.