PAPER Special Section on Discrete Mathematics and Its Applications

The p-Collection Problem in a Flow Network with Lower Bounds

Kaoru WATANABE†, Hiroshi TAMURA††, Keisuke NAKANO†, *and* Masakazu SENGOKU†, *Members*

SUMMARY In this paper we extend the p-collection problem to a flow network with lower bounds, and call the extended problem the lower-bounded p-collection problem. First we discuss the complexity of this problem to show NP-hardness for a network with path structure. Next we present a linear time algorithm for the lower-bounded 1-collection problem in a network with tree structure, and a pseudo-polynomial time algorithm with dynamic programming type for the lower-bounded p-collection problem in a network with tree structure. Using the pseudo-polynomial time algorithm, we show an exponential algorithm, which is efficient in a connected network with few cycles, for the lower-bounded p-collection problem.

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

1. Introduction

Recently the authors discussed the problem where to locate p sinks in a flow network such that the value of a maximum flow is maximum, and called it the p-collection problem [4], [5]. It 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 these resources can be used by as many terminals (clients) as possible. In this paper we extend the p-collection problem to a flow network with lower bounds.

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 $b^-(v) \leq b^+(v)$ for any v of V, let c^- and c^+ be functions : $A \to Z$ such that $c^-(a) \leq c^+(a)$ for any a of A, and let d^- and d^+ be functions : $V \to Z \cup \{\infty\}$ such that $d^-(v) \leq d^+(v)$ for any v of V. We call the 7-tuple $N = (D, b^-, b^+, c^-, c^+, d^-, d^+)$ a *network with lower bounds* (described as $N = (D, b^\pm, c^\pm, d^\pm)$, and called a network, for simplicity). 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 $e^+ : A^*_X \to Z \cup \{\infty\}$ as

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

And we define the adjoint network N_X of N with respect to a subset X of V as the s-t flow network $N_X = (D_X, s, t, e^+, e^-)$ with lower bounds, where $D_X = (V^*, A^*_X)$. Figure 1 (b) shows an adjoint network of the network illustrated in (a). The adjoint network N_X is feasible if there exists a flow f in A_X meeting the following conditions.

Capacity Constrain: For any $a \in A^*_X$,

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

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

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

where $\mathrm{ad}^+(v)$ (respectively, $\mathrm{ad}^-(v)$) denotes the set of vertices adjacent from (to respectively) v in N_X.

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

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

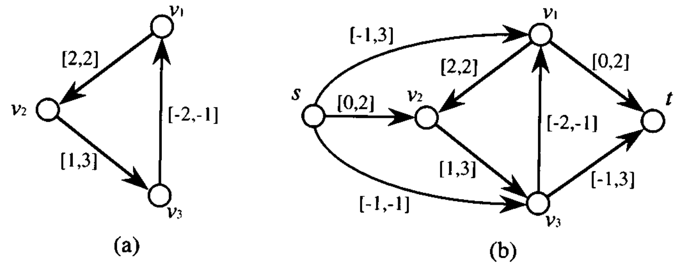
(Notice that $\max_f \mathrm{val}(f)$ is the value of a maximum flow in N_X.) If p is a positive integer with $p \leq |V|$, let $H_N(p) = \max\{h_N(X) : |X| = p\}$. (We usually omit the subscripts N of h_N and H_N when N is clear from context.) A subset X^* of V with $|X^*| = p$ is a *maximum p-collection 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.

Manuscript received September 6, 1996.
Manuscript revised November 1, 1996.

†The authors are with Graduate School of Science and Technology, Niigata University, Niigata-shi, 950–21 Japan.
††The author is with Niigata Institute of Technology, Kashiwazaki-shi, 945–11 Japan.



$$
\begin{array}{llll}
b^-(v_1) = -1, & b^+(v_1) = 3, & d^-(v_1) = 0, & d^+(v_1) = 2 \\
b^-(v_2) = 0, & b^+(v_2) = 2, & d^-(v_2) = 1, & d^+(v_2) = 3 \\
b^-(v_3) = -1, & b^+(v_3) = -1, & d^-(v_3) = -1, & d^+(v_3) = 3
\end{array}
$$

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

The p-collection problem discussed in [4] 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 call this problem PRC, the primary p-collection problem. The sink-capacitated p-collection problem discussed in [6] is equivalent to the subproblem of LBC such that

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

In this paper we call this problem 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. In this paper we deal with three topology types: general network ($\beta = N$), tree ($\beta = T$) and path ($\beta = P$). 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.

In the paper [4] 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 proposed an $O(p^2n^2C^2)$ algorithm for $p/T/$PRC [2]. The report [5] contains some complexity results; the problem $p/N/$PRC is strongly NP-hard, and $p/T/$PRC is weakly NP-hard. The report [3] proposes an $O(pn^2)$ algorithm for $p/P/$PRC.

In Sect. 2 we discuss the complexity of this problem to show NP-hardness of $p/P/$LBC. We present an $O(n)$ algorithm for $1/T/$LBC in Sect. 3, and a pseudo-polynomial time algorithm with dynamic programming type for $p/T/$LBC in Sect. 4. Using the pseudo-polynomial time algorithm, we show an exponential algorithm, which is efficient in a connected network with few cycles, for $p/N/$LBC in Sect. 5.

2. Complexity Results

It is known that $p/N/$PRC is strongly NP-hard [5]. Hence $p/N/$SCC and $p/N/$LBC are strongly NP-hard. We know that $p/T/$PRC is weakly NP-hard [5]. We will show a pseudo-polynomial time algorithm for $p/T/$LBC in Sect. 4. Thus $p/T/$SCC and $p/T/$LBC are weakly NP-hard. There is an $O(pn^2)$ algorithm for $p/P/$PRC [3]. Hence $p/P/$PRC belongs to the class P. In this section we mainly consider the complexity of $p/P/$LBC. We prove the following theorem transforming the partition problem to $p/P/$LBC.

Theorem 1: The problem $p/P/$LBC is weakly NP-hard.

proof: Let $k_1, \ldots, k_n$ be positive integers. The partition problem is that of deciding whether there exists the subset I' of I such that $\sum_{i \in I'} k_i = \sum_{i \in I - I'} k_i$, where $I = \{1, \ldots, n\}$. This problem is known to be NP-complete [1] if n is even, and if we require that $|I'| = n/2$. In the case where $\sum_{i \in I} k_i$ is odd, it is easy to see that there is not a solution. Hence we can suppose that $\sum_{i \in I} k_i$ is even. Let $K = (1/2) \sum_{i \in I} k_i$. We construct an instance of $p/P/$LBC from the partition problem. Let V be the set of n vertices $v_1, \ldots, v_n$, let $A = \{(v_i, v_{i+1}) : 1 \leqq i < n\}$, and let $D = (V, A)$. For any $1 \leqq i \leqq n$, we define b^+ and b^- as

$$
b^\pm(v_i) = \begin{cases} K & \text{if } i = 1 \\ 0 & \text{otherwise,} \end{cases}
$$

and d^+ and d^- as

$$
d^\pm(v_i) = k_i.
$$

For any arc a of A, we associate K with $c^+(a)$, and 0 with $c^-(a)$. We can construct the network $N = (D, b^\pm, c^\pm, d^\pm)$ in polynomial time.

Now we show that, there exists a solution I' of the partition problem with $|I'| = n/2$ if and only if $H(n/2) \geqq 0$. Suppose that there exists a subset I' of I with $|I'| = n/2$ such that $\sum_{i \in I'} k_i = \sum_{i \in I - I'} k_i$. Let $X = \{v_i : i \in I'\}$. Then $h(X) = K$, and so $H(n/2) \geqq 0$.

Conversely assume that $H(n/2) \geqq 0$. Then there exists a subset X of V with $|X| = n/2$ such that $h(X) = H(n/2)$. Let $I' = \{i \in I : v_i \in X\}$. Let f be a flow in the adjoint network N_X. Then $f(s, v_1) = K$, and

$$f(v_i, t) = \begin{cases} k_i & \text{if} \quad i \in I' \\ 0 & \text{otherwise.} \end{cases}$$

Hence we obtain $\sum_{i \in I'} k_i = \sum_{i \in I'} f(v_i, t) = K$. Thus we have $\sum_{i \in I'} k_i = \sum_{i \in I - I'} k_i = K$ with $|I'| = n/2$.

However the algorithm in Sect. 4 solves $p/P/$LBC in pseudo-polynomial time. Consequently this problem is weakly NP-hard. □

The topology type, path, is not essential because the lower-bounded p-collection problem is (weakly or strongly) NP-hard for any kind of topology types with connected structure. The obtained results so far are brought together in Table 1. The complexity of $p/P/$SCC remains open.

3. The Problem $1/T/$LBC

In this section we present a linear time algorithm for $1/T/$LBC. Let $N = (T, b^{\pm}, c^{\pm}, d^{\pm})$ be a network with tree structure, where $T = (V, A)$. Let v_0 be an arbitrary vertex of N, and call v_0 the root of N. For any vertex v of N, we define the level $\text{lev}(v)$ of v as the number of all the arcs in a v-v_0 path between v and v_0. Let $L(l) = \{v \in V : \text{lev}(v) = l\}$ for any non-negative l, and let l^* denote the maximum integer such that $L(l^*) \neq \emptyset$. For convenience, we reverse the direction of any arc (u, v) with $\text{lev}(u) < \text{lev}(v)$. Then we can assume that $c^{\pm}(v, u) = -c^{\mp}(u, v)$. If T' is a subdigraph of T, then $N|T'$ denotes the restriction of N on T'. Let u and v be adjacent vertices of T. If we remove all the arcs adjacent to-or-from v from T without (u, v) or (v, u), then we obtain the subtree T_u^v involving v. Let N_u^v be the s-t flow network obtained by replacing $e^{\pm}(s, v)$ with 0 and $e^{\pm}(v, t)$ with $\pm\infty$ in the adjoint network of $N|T_u^v$ with respect to v. We define $F^+(u, v)$ ($F^-(u, v)$ respectively) as the value of a maximum (respectively, minimum) flow in N_u^v if the obtained network is feasible; otherwise $F^+(u, v) = -\infty$ (respectively, $F^-(u, v) = +\infty$). Let

$$g^{\pm}(u, v) = \sum_{w \in \text{ad}(u) - \{v\}} F^{\pm}(w, u) + b^{\pm}(u),$$

and let

$$G^{\pm}(v) = \sum_{w \in \text{ad}(v)} F^{\pm}(w, v) + b^{\pm}(v),$$

where $\text{ad}(x)$ denotes the set of vertices adjacent to-or-from x. Then we can prove the following lemmas.

Lemma 1: Let $N = (T, b^{\pm}, c^{\pm}, d^{\pm})$ be a network with tree structure. Let u and v be adjacent vertices of T. If $\max\{c^-(u, v), g^-(u, v)\} \leqq \min\{c^+(u, v), g^+(u, v)\}$, then

$$F^+(u, v) = \min\{c^+(u, v), g^+(u, v)\}, \tag{1}$$

and

$$F^-(u, v) = \max\{c^-(u, v), g^-(u, v)\}; \tag{2}$$

otherwise $F^{\pm}(u, v) = \mp\infty$.

Proof: If u is a leaf, then this lemma is trivial. Hence we consider the other case hereafter. Suppose that the lemma holds on $F^{\pm}(w, u)$ for any $w \in \text{ad}(u) - \{v\}$. If there exists a vertex $w \neq v$ incident to-or-from u such that $\max\{c^-(w, u), g^-(w, u)\} > \min\{c^+(w, u), g^+(w, u)\}$, then $F^{\pm}(w, u) = \mp\infty$, and so the lemma holds clearly. Assume that there is not such a vertex w. Then $F^-(w, u) \leqq F^+(w, u)$ for any w of $\text{ad}(u) - \{v\}$. Hence $g^-(u, v) \leqq g^+(u, v)$. If $g^+(u, v) < c^-(u, v)$ or $c^+(u, v) < g^-(u, v)$, i.e., $\min\{c^+(u, v), g^+(u, v)\} < \max\{c^-(u, v), g^-(u, v)\}$, then N_u^v is not feasible, and so $F^{\pm}(u, v) = \mp\infty$. If $\max\{c^-(u, v), g^-(u, v)\} \leqq \min\{c^+(u, v), g^+(u, v)\}$, then Eq. (1) and (2) hold from the capacity constrain. □

Lemma 2: Let $N = (T, b^{\pm}, c^{\pm}, d^{\pm})$ be a network with tree structure, and let v be a vertex of T. If $\max\{d^-(v), G^-(v)\} \leqq \min\{d^+(v), G^+(v)\}$, then

$$h(v) = \min\{d^+(v), G^+(v)\};$$

otherwise $h(v) = -\infty$.

Proof: If there exists a vertex w adjacent to-or-from v such that $F^+(w, v) = -\infty$, then this lemma holds clearly. Assume that there is not such a vertex w. Then $F^-(w, v) \leqq F^+(w, v)$ for any w. Hence $G^-(v) \leqq G^+(v)$. If $G^+(v) < d^-(v)$ or $d^+(v) < G^-(v)$, i.e., $\min\{d^+(v), G^+(v)\} < \max\{d^-(v), G^-(v)\}$, then N_v is not feasible, and so $h(v) = -\infty$. If $\max\{d^-(v), G^-(v)\} \leqq \min\{d^+(v), G^+(v)\}$, then $h(v) = \min\{d^+(v), G^+(v)\}$ holds from the capacity constrain. □

Lemma 3: If there exist different vertices w_1 and w_2 adjacent to-or-from some vertex v in a network with tree structure such that $F^+(w_1, v) = -\infty$ and $F^+(w_2, v) = -\infty$, then $H(1) = -\infty$.

Proof: Let V be the vertex set of this network. Assume that there exist such vertices w_1 and w_2. By Lemma 1, we have $h(v) = -\infty$, and we obtain $h(v_1) = -\infty$ for any vertex v_1 of $V - V(T_{w_1}^v)$, and $h(v_2) = -\infty$ for any vertex v_2 of $V - V(T_{w_2}^v)$. Since $(V - V(T_{w_1}^v)) \cup (V - V(T_{w_2}^v)) \cup \{v\} = V$, we have $h(u) = -\infty$ for any vertex u of V. Hence $H(1) = -\infty$. □

Using Lemma 1, 2 and 3, we can design Algorithm 1 with complexity $O(|V|)$ for the 1-collection problem. Let

$$B = 2 \max \left\{ \left| \sum_{v \in V} b^+(v) \right|, \left| \sum_{v \in V} b^-(v) \right| \right\} + 2.$$

In this algorithm the value B substitutes for ∞.

Table 1 Complexity results.

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

Algorithm 1: (This algorithm returns the maximum 1-collection number.)

```
begin
  B ← 2 max{| Σ_{v∈V} b⁺(v)|, | Σ_{v∈V} b⁻(v)|} + 2;
  for all v ∈ V do G^±(v) ← b^±(v) od;
  for all (u,v) ∈ A do F^±(u,v) ← 0 and F^±(v,u) ← 0 od;
  i ← l*;
  while i > 0 do
    for all arcs (u,v) ∈ A such that lev(u) = i do
      if max{c⁻(u,v), G⁻(u)} ≦ min{c⁺(u,v), G⁺(u)} then
        F⁺(u,v) ← min{c⁺(u,v), G⁺(u)};
        F⁻(u,v) ← max{c⁻(u,v), G⁻(u)};
      else
        F^±(u,v) ← ∓B;
      fi;
      G^±(v) ← G^±(v) + F^±(u,v);
      if G⁺(v) ≦ −3B/2 then return(−∞) fi;
    od;
    i ← i − 1;
  od;
  i ← 1;
  while i ≦ l* do
    for all arcs (u,v) ∈ A such that lev(v) = i do
      g^±(v,u) ← G^±(v) − F^±(u,v);
      if max{c⁻(v,u), g⁻(v,u)} ≦ min{c⁺(v,u), g⁺(v,u)}
      then
        F⁺(v,u) ← min{c⁺(v,u), g⁺(v,u)};
        F⁻(v,u) ← max{c⁻(v,u), g⁻(v,u)};
      else
        F^±(v,u) ← ∓B;
      fi;
      G^±(u) ← G^±(u) + F^±(v,u);
      if G⁺(u) ≦ −3B/2 then return(−∞) fi;
    od;
    i ← i + 1;
  od;
  for all v ∈ V do h(v) ← min{d⁺(v), G⁺(v)} od;
  H(1) ← max{h(v) : v ∈ V};
  return(H(1));
end.
```

4. The Problem $p/T/\mathrm{LBC}$

In this section we present an $O(p^2n^2C^2)$ algorithm for $p/T/\mathrm{LBC}$ based on dynamic programming approach, improving the $O(p^2n^3C^2)$ algorithm in [4]. Given a network with tree structure, for convenience, we consider the network obtained by adding a new vertex v_0 and a new arc (v_1, v_0) with $c^\pm(v_1, v_0) = 0$ where v_1 is an arbitrary vertex of this network. Now let $N = (T, b^\pm, c^\pm, d^\pm)$ be such a network where $T = (V, A)$. In the same way as Sect. 3, we suppose that $\mathrm{lev}(u) > \mathrm{lev}(v)$ for any arc (u,v), and fixing the root on v_0 in N, we define $L(\cdot)$, $\mathrm{lev}(\cdot)$, l^* and T_u^v. Let v be a vertex of N, and let $U(v) = \{u \in \mathrm{ad}(v) : \mathrm{lev}(v) < \mathrm{lev}(u)\} \cup \{s\}$ (where s is the source, and where $\mathrm{ad}(v)$ denotes the set of vertices adjacent to-or-from v). We define D_u^v as the digraph obtained by adding the arcs (s,w) ($w \in V(T_u^v) - \{v\}$) to T_u^v for any vertex u of $U(v) - \{s\}$, and D_s^v as the digraph consisting of the arc (s,v). Let U' be a subset of $U(v)$, and let $D_{U'}^v = (\bigcup_{u\in U'} V(D_u^v), \bigcup_{u\in U'} A(D_u^v))$. For the cut $\{(u,v) : u \in U'\}$ in $D_{U'}^v$, we define $C^+(U')$ and $C^-(U')$ as

$$C^\pm(U') = \sum_{u\in U'} c^\pm(u,v).$$

Let $W = V(D_{U'}^v) - \{s,v\}$, and let $N' = N_\emptyset|D_{U'}^v$. Now we define $R_{U'}[y,k]$, the local optimal value in N'. The integer variables y and k run from $C^-(U')$ to $C^+(U')$, and from 0 to $\min\{p, |W|\}$, respectively. Let X be a subset of W with $|X| = k$, and let N'_X be the digraph obtained by adding the sink t and the arcs (x,t) ($x \in X$) to N'. Let $f_{X,y}$ be a function on the arc set of N'_X with $\sum_{u\in U'} f(u,v) = y$ satisfying the capacity constrain for any arc of N'_X, and the flow conservation for any vertex of W (usually dropping the subscript X, y of $f_{X,y}$ when X and y are clear from context). And we say N' to be $[y,k]$-feasible if there exists such an f for some X with $|X| = k$. We define $R_{U'}[y,k]$ as

$$\begin{cases} \max\limits_X \max\limits_f \sum\limits_{x\in X} f(x,t) & \text{if } N' \text{ is } [y,k]\text{-feasible} \\ -\infty & \text{otherwise.} \end{cases}$$

If N' is $[y,k]$-feasible, then we call a subset X^* of W, a maximum $[y,k]$-collection set of N' such that $R_{U'}[y,k] = \max\{\mathrm{val}(f_{X^*,y}) : f_{X^*,y}\}$. That is, the value of $R_{U'}[y,k]$ and a maximum $[y,k]$-collection set mean the local optimal value and a local optimal solution in N' respectively, when y and k are fixed. From the definition of $R_{U'}[y,k]$, we have $H(p) = R_{v_1}[0,p]$. So we should compute $R_{v_1}[0,p]$ for evaluating $H(p)$.

Now let's study properties of $R_{U'}[y,k]$ separating three cases.

Case 1: The vertex v is a leaf of T ($k = 0$).
Then $U' = \{s\}$, and N' consists of the arc (s,v). From the definition of $[y,k]$-feasibility, if there exists a function f on (s,v) with $f(s,v) = y$ that meets the capacity constrain on (s,v), then N' is $[y,k]$-feasible. Hence for any $C^-(U') \leqq y \leqq C^+(U')$, we have

$$R_{U'}[y,k] = 0.$$

From now on, we deal with cases where the vertex v is not a leaf of T.

Case 2: $|U'| = 1$.
Let u denote the vertex of U', let $U'' = U(u)$ and

$N'' = N_\emptyset|D^u_{U''}$. Furthermore we consider the following two cases.

Case 2.1: u is adjacent to the sink $(1 \leqq k \leqq |W|)$.

Let $q(y,k)$ denote the value of $R_{U'}[y,k]$ in this case. Let y_1, y_2 be integers with $y_1 - y_2 = y$ such that $C^-(U'') \leqq y_1 \leqq C^+(U'')$, and such that $d^-(u) \leqq y_2 \leqq d^+(u)$. In the much same way as the PRC version in [4], we can know that if $C^-(U'') - d^+(u) \leqq y \leqq C^+(U'') - d^-(u)$,

$$q(y,k) = \max_{y_1,y_2}\{R_{U''}[y_1, k-1] + y_2\};$$

otherwise $q(y,k) = -\infty$. For any y and k, let $y_1^*(y,k)$ denote a value of y_1 such that $q(y,k) = R_{U''}[y_1, k-1] + y_2$.

Case 2.2: u is not adjacent to the sink $(0 \leqq k < |W|)$.

Let $r(y,k)$ denote the value of $R_{U'}[y,k]$ in this case. In the much same way as the PRC version in [4], we can know that

$$r(y,k) = \begin{cases} R_{U''}[y,k] & \text{if } C^-(U'') \leqq y \leqq C^+(U'') \\ -\infty & \text{otherwise.} \end{cases}$$

Using values of $q(y,k)$ and $r(y,k)$, we have

$$R_{U'}[y,k] = \begin{cases} r(y,k) & \text{if } k = 0 \\ \max\{q(y,k), r(y,k)\} & \text{if } 1 \leqq k < |W| \\ q(y,k) & \text{if } k = |W|. \end{cases}$$

For any k $(1 \leqq k < |W|)$, if N' is $[y,k]$-feasible, and if $q(y,k) > r(y,k)$, then the union of $\{u\}$ and the maximum $[y_1^*(y,k), k-1]$-collection set is a maximum $[y,k]$-collection set of N'. If N' is $[y,k]$-feasible and if $q(y,k) \leqq r(y,k)$, then the maximum $[y,k]$-collection set of N'' is a maximum $[y,k]$-collection set of N'. It is easy to see that for $k = 0$ or $|W|$, if N' is $[y,k]$-feasible, then the maximum $[y,k]$-collection sets of N' are equal to $\emptyset$ or W, respectively.

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

Let U_1 and U_2 be nonempty subsets of U' such that $U_1 \cup U_2 = U'$ and $U_1 \cap U_2 = \emptyset$. Let $N_1 = N_\emptyset|D^v_{U_1}$ and $N_2 = N_\emptyset|D^v_{U_2}$. Let y_1 and y_2 be integers with $y_1 + y_2 = y$ such that $C^-(U_1) \leqq y_1 \leqq C^+(U_1)$ and $C^-(U_2) \leqq y_2 \leqq C^+(U_2)$. Let k_1, k_2 be integers with $k_1 + k_2 = k$ such that $0 \leqq k_1 \leqq \min\{p, |V(N_1)| - 2\}$ and $0 \leqq k_2 \leqq \min\{p, |V(N_2)| - 2\}$. We can evaluate $R_{U'}[y,k]$ using the values of $R_{U_1}[y_1, k_1]$ and $R_{U_2}[y_2, k_2]$. If N' is $[y,k]$-feasible, then the following equation holds.

$$R_{U'}[y,k] = \max_{y_1,y_2,k_1,k_2}\{R_{U_1}[y_1,k_1] + R_{U_2}[y_2,k_2]\} \quad (3)$$

where y_1, y_2, k_1, k_2 run meeting two conditions that N_1 is $[y_1,k_1]$-feasible, and that N_2 is $[y_2,k_2]$-feasible. If N_1 is not $[y_1,b_1]$-feasible, or N_2 is not $[y_2,b_2]$-feasible, then $R_{U_1}[y_1,k_1] + R_{U_2}[y_2,k_2] = -\infty$. Moreover, if N' is not $[y,k]$-feasible, then N_1 is not $[y_1,b_1]$-feasible, or N_2 is not $[y_1,b_1]$-feasible, for any y_1, y_2, b_1, b_2. Thus

Algorithm 2: (This algorithm returns the maximum p-collection number.)

```
begin
  i ← l*;
  while i ≧ 1 do
    for all v ∈ L(i) do
      for all u ∈ U(v) do
        if u is a leaf then
          calculate R_u; {Case 1}
        else
          calculate R_u using R_U(u); {Case 2}
        fi;
      od;
      U_1 ← the set of an arbitrary vertex of U(v);
      while U_1 ≠ U(v) do
        U_2 ← the set of an arbitrary vertex of U(v) − U_1;
        U' ← U_1 ∪ U_2;
        calculate R_U' using R_U1 and R_U2;{Case 3}
        U_1 ← U';
      od;
    od;
    i ← i − 1;
  od;
  calculate R_v1[0, p];
  return(R_v1[0, p]);
end.
```

Eq. (3) holds even if y_1 and k_1 are allowed to be a value such that N_1 are not $[y_1,k_1]$-feasible, and if y_2 and k_2 are allowed to be values such that N_2 are not $[y_2,k_2]$-feasible. If y_1', y_2', k_1', k_2' are integers such that $R_{U'}[y,k] = R_{U_1}[y_1',k_1'] + R_{U_2}[y_2',k_2']$, then the union of the maximum $[y_1',k_1']$-collection set of N_1 and the maximum $[y_2',k_2']$-collection set of N_2 is a maximum $[y,k]$-collection set of N'.

Using the above consequences, we obtain Algorithm 2 for the p-collection problem in a network with tree structure. Let $R_{U'}$ denote the array of $R_{U'}[y,k]$s for all y and k. If $|U'| = 1$, then this algorithm computes $R_{U'}$ using $R_{U''}$; otherwise using R_{U_1} and R_{U_2} where U_2 is the set of an arbitrary vertex of U', and where $U_1 = U' - U_2$. That is, first the algorithm computes R_u for any arcs (u,v) incident from each vertex of $L(l^*)$, next it calculates R_u for any arcs (u,v) incident from each vertex of $L(l^* - 1)$, and so on. Last it computes $R_{v_1}[0,p]$.

We examine the complexity of Algorithm 2. Let d and n be the maximum degree of vertices and the number of vertices in T, respectively. Let $C = \max\{|e^+(u,v)|, |e^-(u,v)| : (u,v) \in A^*\}$. If $|U'| = 1$ (corresponding to Case 2), then the complexity of evaluating the value of $R_{U'}[y,k]$ is $O(C)$, and the frequency of evaluating is $O(pnC)$. If $|U'| \geqq 2$ (corresponding to Case 3), then the complexity of estimating the value of $R_{U'}[y,k]$ is $O(kC)$, and the frequency of estimating is $O(pndC)$. As $k \leqq p$, the complexity of Algorithm 2 is $O(nd(pC)^2)$, i.e., $O(p^2n^2C^2)$.

5. The Problem $p/N/$LBC

In this section we present an algorithm in a connected

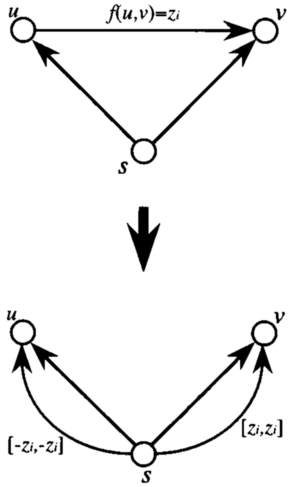


Fig. 2 Removing the arc (u, v) with flow z_i of a cotree.

network with cycles, applying Algorithm 2. This algorithm is efficient in a network with few cycles. Let $N = (D, b^{\pm}, c^{\pm}, d^{\pm})$ be a connected network with $D = (V, A)$. Let $T = (V, A')$ denote a spanning tree, let $\overline{A'} = A - A'$, and let $a_1, \ldots, a_m$ be $|\overline{A'}| (= m)$ arcs of $\overline{A'}$, and let z_i be an integer such that $c^-(a_i) \leqq z_i \leqq c^+(a_i)$ for any $1 \leqq i \leqq m$. The basic idea of this algorithm is that for any $a_i (= (u, v))$ with flow z_i, we remove a_i from an adjoint network, and add a new arc from s to u, which is different from the arc (s, u), such that $e^{\pm}$ of the new arc are equal to $-z_i$, and a new arc from s to v, which is different from the arc (s, v), such that $e^{\pm}$ of the new arc are equal to $+z_i$ (Fig. 2).

For any v of V we define two functions b'^{+} and b'^{-} on V as

$$b'^{\pm}(v) = b^{\pm}(v) + \sum_{i \in I} z_i - \sum_{j \in J} z_j,$$

where $I = \{1 \leqq i \leqq m : a_i \text{ is incident to } v\}$ and $J = \{1 \leqq j \leqq m : a_j \text{ is incident from } v\}$ for any v of V. Let $N(z_1, \ldots, z_m)$ be the network $(T, b'^{\pm}, c^{\pm}|A', d^{\pm})$. The value of $H_{N(z_1,\ldots,z_m)}(p)$ equals the maximum p-collection number of the network obtained by changing $c^{\pm}(a_i)$ to z_i for $1 \leqq i \leqq m$. To calculate $H_N(p)$, we should evaluate the value of $H_{N(z_1,\ldots,z_m)}(p)$ for any $c^-(a_1) \leqq z_1 \leqq c^+(a_1), \ldots, c^-(a_m) \leqq z_m \leqq c^+(a_m)$. So we obtain an algorithm for a connected network.

Algorithm 3:

Step 1: For all $z_1, \ldots, z_m$, calculate $H_{N(z_1,\ldots,z_m)}(p)$ applying $N(z_1, \ldots, z_m)$ to Algorithm 2.

Step 2: $H_N(p) \leftarrow \max_{z_1,\ldots,z_m} H_{N(z_1,\ldots,z_m)}(p)$.

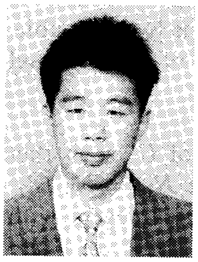
Since Algorithm 3 calls Algorithm 2 $O(C^m)$ times, the complexity of Algorithm 3 is $O(p^2 n^2 C^{m+2})$, that is, $O(p^2 n^2 C^{|A|-n+3})$.

6. Conclusion

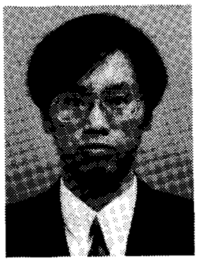
In this paper we extend the p-collection problem to a flow network with lower bounds. First we discuss complexity of this problem to show NP-complete results of the subproblems even for $p/P/$LBC. The complexity of $p/P/$SCC remains open. Next we show a linear time algorithm for $1/T/$LBC, and a pseudo-polynomial time algorithm with dynamic programming type for $p/T/$LBC. Using the pseudo-polynomial time algorithm, we show an exponential algorithm for $p/N/$LBC, which is efficient in a connected network with few cycles.

References

[1] M.R. Garey and D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-completeness," Freeman, San Francisco, 1979.
[2] S. Tsukiyama, "A new algorithm for p-collection problem on a tree-type flow network," Proc. ITC-CSCC, vol.2, pp.721–724, 1996.
[3] K. Watanabe, H. Tamura, and M. Sengoku, "Problems of where to locate p sinks in a flow network," Proc. IEICE 7th Karuizawa CAS Workshop, pp.339–344, 1994.
[4] 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.
[5] K. Watanabe, H. Tamura, and M. Sengoku, "Problems of where to locate p-sinks in a flow network: Complexity approach," Proc. IEICE 8th Karuizawa CAS Workshop, pp.413–418, 1995.
[6] K. Watanabe, H. Tamura, and M. Sengoku, "Problems of where to locate capacitated p sinks in a flow network," Proc. JTC-CSCC, pp.758–761, 1995.



Kaoru Watanabe was born in Niigata, Japan on April 13, 1967. He received the B.E. and M.E. degrees from Niigata University in 1990 and 1992, respectively. He is presently working toward the Ph. D degree at Niigata University. He received the Paper Award from IEICE in 1996. He is interested in graph theory and algorithm theory.



Hiroshi Tamura was born in Saitama prefecture, Japan, November 16, 1959. He 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. In 1991, he was an Associate Professor at Center for Cooperative Research, Niigata University. He is presently an Associate Professor at Niigata Institute of Technology. His research interests are in computational geometry, network theory and graph theory. He received the Paper Award from IEICE in 1992 and 1996. He is a member of IPS of Japan and the Mathematical Society of Japan.

Keisuke Nakano was born in Niigata, Japan, on April 22, 1966. He received the B.E. and M.E. degrees from Niigata University in 1989 and 1991, respectively. He received the Ph.D. degree from Niigata University in 1994. He was with Niigata College of Technology from 1994 to 1996. He is now a Research Associate of Graduate School of Science and Technology, Niigata University. His research interests include performance evaluation of mobile information networks.

Masakazu Sengoku was born in Nagano prefecture, Japan, on October 18, 1944. He 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 Award from IEICE in 1992 and 1996. He is a member of IEEE and IPS of Japan