論　文

フローネットワークの出口配置問題

正員　渡辺　郁[†]　　正員　田村　裕[††]　　正員　仙石　正和[†]

Problems of Where to Locate p-Sinks in a Flow Network

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

あらまし　フローネットワークのロケーション問題としては，p-センター問題やp-メジアン問題があり，それらは多項式時間で解けることが知られている．本論文では，フローネットワークの新たな一つのロケーション問題を提案する．それは一つの固定された入口とp個の（固定されていない）出口をもつフローネットワークNを考え，Nに最大フローが最大になるようにp個の出口をうまく配置する問題である（この問題をp-回収問題と呼ぶ）．まず木状ネットワークに対する1-回収問題を解く線形時間アルゴリズム，次に動的計画法に基づいた木状ネットワークのp-回収問題を解く疑多項式時間アルゴリズムを与える．また木状ネットワークに対する1-回収問題に対応する判定問題はNP-完全であることが知られているので，その判定問題が強NP-完全でないことがわかる．

キーワード　フローネットワーク，ロケーション問題，疑多項式時間アルゴリズム，動的計画法

1. まえがき

ネットワークを$|V|(=n)$個の点に非負の整数である重み，$|A|$個の弧に正の整数である容量が割り当てられた有向グラフ$D=(V,A)$とする．CをAの弧の容量と点の重みの中の最大の整数とする．都市をネットワークの頂点，都市の単位時間当り下水の排水量を点の重み，下水管を弧，下水管の単位時間当り流すことのできる量を弧の容量に対応させることで，下水道網はネットワークにモデル化することができる．そこで，最も効率良く都市に下水処理施設（出口）を設置するかを考える．このようなロケーション問題をp-回収問題と呼び，2.でその厳密な定義を行う．このような問題は，コストより通信トラヒックの方が重要な場合におけるコンピュータネットワーク上の複数資源の最適配置を求める問題などに応用でき（詳しくは2.で述べる），それは重要な問題と言える．

一つの入口と一つの出口をもつフローネットワークの最大フローを求める問題は長い間にわたって議論されてきた[(1)]．またネットワークのロケーション問題であるp-センター問題とp-メジアン問題はNP-困難であることが知られている[(2),(3)]．またフローネットワークにおいては，p-センター問題とp-メジアン問題は多項式時間で解けることが知られている[(4)]．筆者らはp-回収問題に対応する判定問題が強NP-完全であることと，木状ネットワークにおいてすらNP-完全であることを証明した[(5),(6)]．しかし木状ネットワークにおいて，強NP-完全かどうかはわかっていなかった．

本論文では主に木状ネットワークについて話を進める．3.では木状ネットワークの1-回収問題の$O(n)$時間アルゴリズムを与え，4.では動的計画法を用いた木状ネットワークのp-回収問題$O(p^2n^3C^2)$時間（疑多項式時間）アルゴリズムを与える．これは指数時間アルゴリズムではあるが，Cが小さい値のときは十分速く走るし，これから良い近似アルゴリズムをも構成できる[(8)]．また一般のネットワークに対するp-回収問題が強NP-完全であるので，それは疑多項式時間アルゴリズムはもち得ない．しかし木状ネットワークに対してはNP-完全であるが，疑多項式時間アルゴリズムが存在することがわかる．

2. 諸定義

ネットワーク$N=(D,c,d)$は，ネットワークを$|V|$個のそれぞれの点vに非負の整数である重み$d(v)$，$|A|$個のそれぞれの弧aに正の整数である容量$c(a)$が割り当てられた有向グラフ$D=(V,A)$とする．ある特別

[†] 新潟大学大学院自然科学研究科，新潟市
The Graduate School of Science and Technology, Niigata University, Niigata-shi, 950-21 Japan
[††] 新潟工科大学，柏崎市
Niigata Institute of Technology, Kashiwazaki-shi, 945 Japan

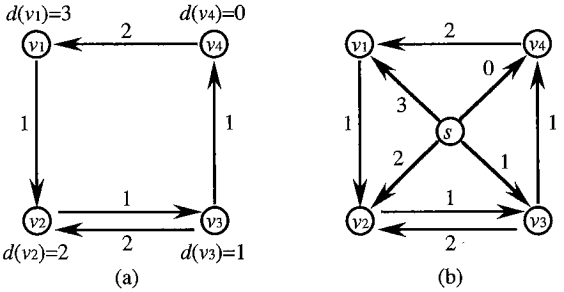

図1 (a) ネットワークの例，(b) その随伴ネットワーク
Fig.1 (a) An example of a network, (b) Its adjoint network.

な一つの点 $s \notin V$ を入口，A_s を弧集合 $\{(s,v)|v \in V\}$ とする．また $V^* = V \cup \{s\}$，$A^* = A \cup A_s$ とおき，$D^* = (V^*, A^*)$ を点集合 V^* と弧集合 A^* をもつ有向グラフとする．N の随伴ネットワーク $N^* = (D^*, s, c^*)$ を入口 s と次のように定義される辺容量：

$$c^*(a) = \begin{cases} c(a) & a \in A \\ d(v) & a(=(s,v)) \in A_s \end{cases}$$

をもつ有向グラフ $D^* = (V^*, A^*)$ とする．図1はネットワークとその随伴ネットワークの例を示している．

X を V の部分集合，フローネットワーク N_X^* を X の点に固定された $|X|$ 個の出口をもった随伴ネットワーク N^* とする（このような部分集合 X を出口集合と呼ぶ）．ネットワーク N が与えられたとき，V のそれぞれの部分集合 X に対して，X の回収数 $h_N(X)$ を，フローネットワーク N_X^* における s から X への最大フロー f の値 $\mathrm{val}(f)$ とする（添字の N が文脈から明らかな場合は省略する）．$H_N(p) = \max\{h(X) : |X| = p$ かつ $X \subseteq V\}$ としたとき，$H(p)$ を N の最大 p-回収数と呼ぼう．また $|X'| = p$ かつ $h(X') = H(p)$ となるような V の部分集合 X' を N の最大 p-回収集合と呼ぶ．ネットワーク N において最大 p-回収集合を求める最適化問題を p-回収問題と呼ぶ．例として，図1のネットワークの p-回収問題について考える．$p = 1$ のときは，$h(v_1) = 4$，$h(v_2) = 4$，$h(v_3) = 2$，$h(v_4) = 1$ であり，それ故 $H(1) = 4$ で，$\{v_1\}, \{v_2\}$ は最大1-回収集合である．$p = 2$ のときは，$h(v_1, v_2) = 6$，$h(v_1, v_3) = 5$，$h(v_1, v_4) = 4$，$h(v_2, v_3) = 4$，$h(v_2, v_4) = 4$，$h(v_3, v_4) = 2$ であり，それ故 $H(2) = 6$ で，$\{v_1, v_2\}$ は最大2-回収集合である．

p-回収問題に対して，N^* の p 個の点に出口を配置する ${}_{|V|}C_p$ 個の組合せに対して最大フローアルゴリズム[7]を適用する自明な $O(|V|^{p+1}|E| \log |V|^2/|E|)$ 時間アルゴリズムを構成することができる．またネットワークが木状のとき，それは $O(|V|^{p+1})$ 時間で解くことができる．

または無向ネットワークに対しても，グラフ $G = (V, E)$，点の重み関数 $d : V \to Z_0^+$（非負整数の集合）と辺の容量関数 $c : E \to Z^+$（正整数の集合）の三つ組を考えることで同様に定義できる．この場合，グラフ G のそれぞれの辺 e を容量 $c(e)$ をもつ互いに向きの反対の二つの弧に置き換えることで，一般性を失うことなく有向の場合に帰着できる．本論文で出てくる（有向ネットワークに対する）アルゴリズムはすべて無向の場合にも適用可能であるので，有向ネットワークだけを扱うことにする．

コストより通信トラヒックが重要なコンピュータネットワークに資源を配置する問題において，無向の場合は特に重要である．辺をネットワークの回線，点をネットワークに接続されたコンピュータ群を表すものとする．点の重みをネットワークに接続されているコンピュータ（クライアント）の個数，辺の重みを回線容量とする．データベース，大型コンピュータなど資源（サーバ）は p 個の点からなる点集合 X に配置され，クライアントから回線を通して利用されるものとする．ここでは考えやすいようにすべての辺 e を容量 $c(e)$ 本の多重辺に置き換えて得られた多重グラフ G' を考えよう．それぞれのクライアントからはどれか一つのサーバしか利用できないとすると（一つのサーバは複数のクライアントから同時に利用できる），そのとき使われている回線は，G' のクライアントがある点から利用されたサーバがある点への道で表現される．複数のクライアントからサーバを利用した場合，それ

らで使用される回線は，互いに辺素な道で表現される．従って $h(X)$ は同時にサーバを利用できるクライアントの数を示している．よって最大 p-回収数 $H(p)$ は p 箇所のサーバの配置で，同時にサーバを利用できるクライアントの最大数，最大 p-回収集合はそのときのクライアントの配置を示している．

3. 木状ネットワークの 1-回収問題

本章では木状ネットワークの 1-回収問題の線形時間アルゴリズムを与える．$N=(T,c,d)$ を木状ネットワークとする（$T=(V,A)$，$n=|V|$）．簡単のため，$(u,v)\notin A$ かつ $(v,u)\in A$ のときは容量 0 の弧 (u,v) を Tに加えておく．このようにして得られた有向グラフは対称で，その基礎グラフは木である．以下の章では，このような弧の容量に 0 をとることが許された対称な木状ネットワークを扱うものとする．

v_0 を Tの任意の点とする．すべての非負整数 x に対して，

$$L(x)=\left\{u\in V \;\middle|\; \begin{array}{l} v_0 \text{ から } u \text{ までの道に} \\ \text{含まれる辺の数が } x \end{array}\right\}$$

と定義する．そのとき，$v_0\in L(0)$ は N の根と呼ばれる．図 2 にこの根付き木の例を示す．l を $L(x)\neq\emptyset$ を満たす最大の整数 x とする．v が $v\in L(x)$ を満たすとき，x を v のレベル $\mathrm{lev}(v)$ と呼ぶ．T'が Tの部分グラフのとき，$N|T'$を Nの T'への制限と呼び，$N|T'=(T',c|T',d|T')$ である（但し $c|T'$, $d|T'$はそれぞれ c と d の $A(T')$ と $V(T')$ への制限である）．v と u を隣接し合う Tの点とする．弧 (u,v) と弧 (v,u) を Tから削除したとき，u を含む部分グラフ $T_{(u,v)}$ と v を含む Tの部分グラフ $T_{(v,u)}$は木である．$F(u,v)$ を出口 t と容量 $c(u,v)$ をもつ弧 (u,t) を $(N|T_{(u,v)})^*$ に加えて得られたフローネットワークの最大フローの値と定義する．そのときそれが次の命題を満たす．

[命題 1] x が Vの点のとき，Nにおいて $I(x)$ を x へ隣接する点の集合とする．そのとき

$$F(u,v)=\min\{c(u,v),\ \sum_{x\in I(u)-\{v\}} F(x,u)+d(u)\},$$

$$h(\{v\})=\sum_{x\in I(v)} F(x,v)+d(v). \qquad \square$$

命題 1 から，木状ネットワークの Nの最大 1-回収集合を求める $O(n)$ 時間アルゴリズムが得られる．それはまず $L(l)$ の点から $L(l-1)$ の点へ接続するすべての弧

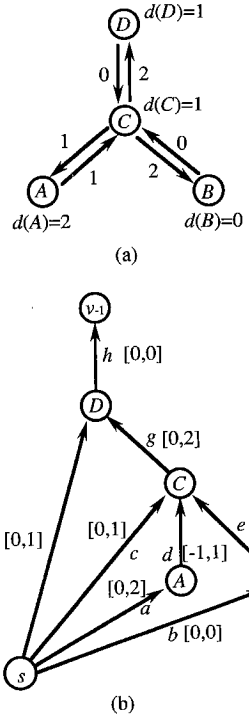

図 2 (a) ネットワーク Nの例，(b) それに対応する有向グラフ D の例

Fig. 2 (a) An examples of a network N, (b) The digraph D correspond to N(b).

a に対して $F(a)$ の値を求め，次に $L(l-1)$ の点から $L(l-2)$ の点へ接続するすべての弧 a に対して $F(a)$ の値を求める，以下同様にこれを繰り返し $L(1)$ の点から $L(0)$ の点（根）へ接続するすべての弧 a に対して $F(a)$ の値を求める．そして今度は逆に $L(0)$ の点から $L(1)$ の点へ接続するすべての弧 a に対して $F(a)$ の値を求める．以下同様にこれを繰り返し，$L(l-1)$ の点から $L(l)$ の点へ接続するすべての弧 a に対して $F(a)$ の値を求める．これにより各点に対する最大 1-回収数が求められ，これから最大 1-回収集合が得られる．この章の終わりに，木状ネットワークにおける 1-回収問題のアルゴリズム 1 を示す．これは木状ネットワークに対して最大フローアルゴリズムを n 回使う自明な $O(n^2)$ 時間アルゴリズムより速く走る．

[アルゴリズム 1] このアルゴリズムが停止したとき，$\{w\}$ は最大 1-回収集合で，$G(v)$ は各点 $\{v\}$ の回収数である．

```
begin
  for all v ∈ V do G(v) ← d(v);
  for all a ∈ A do F(a) ← 0;
  i ← l;
  while i > 0 do begin
    for all arcs (u, v) ∈ A such that lev(u) = i
    and lev(v) = i − 1 do begin
      F(u, v) ← min{G(u), c(u, v)};
      G(v) ← G(v) + F(u, v);
    end;
    i ← i − 1;
  end;
  i ← 1;
  H ← −1;
  while i ≦ l do begin
    for all arcs (v, u) ∈ A such that lev(u) = i
    and lev(v) = i − 1 do begin
      F(v, u) ← min{G(v) − F(u, v), c(v, u)};
      G(u) ← G(u) + F(v, u);
      if H < G(u) then H ← G(u) and w ← u;
    end;
    i ← i + 1;
  end;
end.
```

4. 木状ネットワークの p-回収問題

本章では木状ネットワークにおける疑多項式時間アルゴリズムを与える．それは動的計画法に基づき，根付き木に対して帰りがけ順に部分木に対する局所的な最大回収集合を根に向って求めていく．

$N=(T,c,d)$ $(T=(V,A))$ を **3.** と同様に容量 0 の弧を許した対称な木状ネットワークとする．Vの任意の点 v_0を根とし，**3.** と同様に Tを根付き木とし，$L(\cdot)$, $\mathrm{lev}(\cdot)$, l を定義する．更にここではレベル $\mathrm{lev}(v_{-1})=-1$ をもつ点 $v_{-1}\notin V$を考え，また $L(-1)=\{v_{-1}\}$ とおき，弧 a_0を (v_0,v_{-1}) とする．sを入口，A_sを集合 $\{(s,v)|v\in V\}$ とし，弧集合 A'を $\{(u,v)\in A|\mathrm{lev}(v)<\mathrm{lev}(u)\}\cup A_s\cup\{a_0\}$ と定義する．また $V'=V\cup\{s,v_{-1}\}, D=(V',A')$ とし話を進めると便利である．図 2 (b) にこのような有向グラフ Dと (a) にもとのネットワーク Nの例を示す．v を V の任意の点としたとき，B_vを Dにおいて v へ接続する弧の集合，$n(v)$ を v から隣接する（唯一の）点とする．任意に $B_v-\{(s,v)\}$ の弧にそれぞれ $2,3,\cdots,|B_v|$，弧 (s,v) には 1 とラベル付けをする．$\sigma_v(a)$ を $a\in B_v$ のラベルとする（$\sigma_v^{-1}(x)$ はラベル x をもつ B_vの弧である）．B_vの弧で$\sigma_v(a)$ 以下のラベルをもつものの集合を $B_v(a)$ とおく．すべての弧 $a=(u,v)\in A'$に対して，$D-\{s\}$ から a 以外の弧で v からまたは v へ隣接するものをすべて削除することで得られた成分の中で弧 a を含む部分有向木を T_aとする．D_aを T_aに弧集合 $\{(s,v)|v\in V(T_a)\}$ の弧すべてを加えて得られた有向グラフ，$\tilde{D}_a$を有向グラフ

$$\bigcup_{e\in B_v(a)} D_e = \Big(\bigcup_{e\in B_v(a)} V(D_e),\ \bigcup_{e\in B_v(a)} A(D_e)\Big)$$

とする．

すべての弧 $a=(u,v)\in A'$に対して次のような関数を定義する．但し，$a=(u,v)\in A$ のとき，$\bar{a}$は弧 (v,u) を表すものとする．

$$c^+(a)=\begin{cases} c^*(a) & a\in A'-\{a_0\}\\ 0 & \text{その他の場合,}\end{cases}$$

$$c^-(a)=\begin{cases} -c(\bar{a}) & a\in A'-A_s\\ 0 & \text{その他の場合,}\end{cases}$$

$$C^+(a)=\sum_{a'\in B_v(a)} c^+(a'),$$

$$C^-(a)=\sum_{a'\in B_v(a)} c^-(a')$$

また $C_{\max}=\max\{C^+(a):a\in A'\}$, $C_{\min}=\min\{C^-(a):a\in A'\}$ とおく．図 4 (b) の辺 x についている整数の対は $[c^-(x),c^+(x)]$ を示している．

Xが大きさ p をもつ T^*の出口集合のとき，フローネットワーク T_X^*におけるフロー fは次のような関数 f'で表現できる．すべての $a\in A'$に対して；

$$f'(a)=\begin{cases} 0 & a=a_0\\ f(a) & a\in A_s\\ f(a)-f(\bar{a}) & \text{その他の場合}\end{cases}$$

このとき次の命題が成り立つことが容易にわかる．

[命題 2] $X\subseteqq V$が T^*の出口集合のとき，fが T_X^*のフローであるのはすべての弧 $a\in A'$に対して，

$$c^-(a)\leqq f'(a)\leqq c^+(a),$$

かつすべての点 $v\in V-X$に対して，

$$\sum_{a\in B_v} f'(a)-f'(v,n(v))=0$$

が成り立つときに限る． □

[命題 3] $X\subseteqq V$が T^*の出口集合，fが T_X^*における最大フローであるとき，

$$h_T(X)=\sum_{v\in X}\left\{\sum_{a\in B_v} f'(a)-f'(v,n(v))\right\}$$

である． □

これから D_a に対してと $q_a[b,k]$ を定義する．また括弧内は $\tilde{D}_a$ に対する $r_a[b,k]$ の定義に対応する．$a=(u,v)$ を $A'-A_s(A')$ の任意の弧とする．また $U(a)=V(D_a)-\{s,v\}$ $(W(a)=V(\tilde{D}_a)-\{s,v\})$ とおく．b，k と X をそれぞれ $C_{\min}\leqq b\leqq C_{\max}$ である整数，$0\leqq k\leqq \min\{p,|U(a)|\}(0\leqq k\leqq \min\{p,|W(a)|\})$ である整数，任意の $|X|=k$ なる $U(a)$ $(W(a))$ の出口集合とする．$g_{(X;b)}: A(D_a)$ $(A(\tilde{D}_a))\to Z$ を，$g(a)=b$ $(\sum_{e\in B_{v(a)}} g(e)=b)$，かつすべての $e\in A(D_a)$ $(e\in A(\tilde{D}_a))$ に対して $c^-(e)\leqq g(e)\leqq c^+(e)$，かつすべての $x\in U(a)-X$ $(x\in W(a)-X)$ に対して $\sum_{e\in B_x} g(e)-g(x,n(x))=0$ を満たす関数とする．そのような関数を D_a $(\tilde{D}_a)$ における部分フローと呼び，添字の $(X;b)$ が文脈から明らかな場合は省略する．部分フロー g の値 $\mathrm{val}(g)$ は次のように定義される．

$$\mathrm{val}(g)=\sum_{x\in X}\left\{\sum_{e\in B_x} g(e)-g(x,n(x))\right\}$$

すべての b,X に対してこのような部分フロー g が常に存在するとは限らない．D_a $(\tilde{D}_a)$ が $[b,k]$-実行可能であると言うのは，ある X $(|X|=k)$ に対して部分フロー $g_{(X,b)}$ が存在するときである．そこで D_a $(\tilde{D}_a)$ の最大 $[b,k]$-回収数 $q_a[b,k]$ $(r_a[b,k])$ を次のように定義する．

$$q_a[b,k]\ (r_a[b,k])=\begin{cases}\max\limits_{X,g}\{\mathrm{val}(g)\} & D_a\ (\tilde{D}_a)\text{ が }[b,k]\text{-実行可能}\\ -\infty & \text{その他の場合}\end{cases}$$

また $q_a[b,k]$ $(r_a[b,k])\neq-\infty$ のとき，ある部分フロー $g_{(X';b)}$ が存在して $q_a[b,k]$ $(r_a[b,k])=\mathrm{val}(g_{(X';b)})$ となる X' を D_a $(\tilde{D}_a)$ の最大 $[b,k]$-回収集合と言う．単に q_a (r_a) と書く場合は，それはすべての b,k に対する配列 $q_a[b,k]$ $(r_a[b,k])$ の全体を示すものとする．命題 2，3 と $q_a[b,k]$ の定義から，

$$q_{a_0}[0,p]=H_T(p).$$

である．それ故 $H_T(p)$ を求めるために $q_{a_0}[0,p]$ を計算すればよい．

これから三つの場合にわけて D_a の最大 $[b,k]$-回収数 $q_a[b,k]$ の値について考える．$a=(u,v)$ を $A'-A_s$ の任意の弧，a^* を B_u のラベル $|B_u|$ をもつ弧とする．

(Case 1) $k=0$：そのとき，u は出口でない．それ故，すべての $c^-(a)\leqq b\leqq c^+(a)$ に対して，D_a が $[b,k]$-実行可能なのは $\tilde{D}_{a^*}$ が $[b,k]$-実行可能であるときに限り，$\tilde{D}_{a^*}$ と D_a が $[b,k]$-実行可能ならば $q_a[b,0]=r_{a^*}[b,0]=0$ である．すべての $b<c^-(a),c^+(a)<b$ に対しては D_a が $[b,k]$-実行可能ではないので，

$$q_a[b,0]=\begin{cases}r_a[b,0] & c^-(a)\leqq b\leqq c^+(a)\\ -\infty & \text{その他の場合}\end{cases}$$

が成り立つ．また D_a が $[b,k]$-実行可能となる b に対して，D_a の最大 $[b,k]$-回収集合は空集合である．

(Case 2) $1\leqq k<|U(a)|$：二つの場合について考える．

(Case 2.1) u が出口の場合：$q'_a[b,k]$ をこの場合の $q_a[b,k]$ の値とする．このとき $\tilde{D}_{a^*}$ が $[b',k]$-実行可能となる整数 b'（例えば $b'=0$）が存在する．それ故，すべての $c^-(a)\leqq b\leqq c^+(a)$ に対して，D_a は $[b,k]$-実行可能である．b' が $\tilde{D}_{a^*}$ が $[b',k]$-実行可能となる値をとるとき，$c^-(a)\leqq b\leqq c^+(a)$ に対して

$$q'_a[b,k]=\max_{b'}\{r_{a'}[b',k-1]+(b'-b)\}\qquad(1)$$

が成り立つ．$\tilde{D}_{a^*}$ が $[b',k]$-実行可能でないならば，$r_{a^*}[b',k-1]+(b'-b)=-\infty$ である．それ故，式 (1) は，$C^-(a^*)\leqq b'\leqq C^+(a^*)$ に対して $\tilde{D}_{a^*}$ が $[b',k]$-実行可能でないような値をとったとしても成り立つ．よって次式が成り立つ．

$$q'_a[b,k]=\begin{cases}\max\limits_{b'}\{r_{a^*}[b',k-1]+b'\}-b\\ \qquad c^-(a)\leqq b\leqq c^+(a)\\ -\infty\quad\text{その他の場合}\end{cases}$$

このとき，$b^*_a(b,k)$ を $q'_a[b,k]=r_{a^*}[b',k-1]+b'-b$ となる任意の b' の値とする．

(Case 2.2) u が出口でない場合：$q''_a[b,k]$ をこの場合の $q_a[b,k]$ の値とする．そのとき，すべての $c^-(a)\leqq b\leqq c^+(a)$ に対して，D_a が $[b,k]$-実行可能なときは $\tilde{D}_{a^*}$ は $[b,k]$-実行可能なときに限り，なお $\tilde{D}_{a^*}$ と D_a が $[b,k]$-実行可能ならば $q''_a[b,k]=r_{a^*}[b,k]\geqq 0$ である．またすべての $b<c^-(a),c^+(a)<b$ に対して，D_a は $[b,k]$-実行可能でないので，次式が得られる．

$$q''_a[b,k]=\begin{cases}r_{a^*}[b,k] & c^-(a)\leqq b\leqq c^+(a)\\ -\infty & \text{その他の場合}\end{cases}$$

Case 2.1 の $q'_a[b,k]$ と Case 2.2 の $q''_a[b,k]$ と $q_a[b,k]$ の間には次の関係がある．

$$q_a[b,k]=\max\{q'_a[b,k],q''_a[b,k]\}$$

D_aが$[b,k]$-実行可能であるとき，$R_{a^*}[i,j]$は$\tilde{D}_{a^*}$の任意の最大$[i,j]$-回収集合を表すものとする．D_aが$[b,k]$-実行可能となるbに対して，$q'_a[b,k] > q''_a[b,k]$のときは$R_{a^*}[b^*_a[b,k],k-1]\cup\{u\}$は$D_a$の最大$[b,k]$-回収集合であり，そうでないときは$R_{a^*}[b,k]$は$D_a$の最大$[b,k]$-回収集合である．

(Case 3)　$k=|U(a)|$：そのときuはD_aの出口である．それ故すべての$c^-(a)\leqq b\leqq c^+(a)$に対して，$D_a$は$[b,k]$-実行可能である．入口でない$\tilde{D}_{a^*}$の頂点は出口であるので，$q_a[b,k]=\sum_{e\in A_s}c^+(e)-b=\sum_{x\in U(a)}d(x)-b$である．従って，次の結果が得られる．

$$q_a[b,k]=\begin{cases}\displaystyle\sum_{x\in U(a)}d(u)-b & c^-(a)\leqq b\leqq c^+(a)\\ -\infty & \text{その他の場合}\end{cases}$$

またD_aが$[b,k]$-実行可能となるbに対して，$U(a)$が最大$[b,k]$-回収集合である．

今度は$\tilde{D}_a$の最大$[b,k]$-回収数$r_a[b,k]$の値について二つの場合に分けて考える．$a=(u,v)$をA'の任意の弧とする．aがB_vのラベル1をもつ弧でないとき，a'をラベル$(\sigma(a)-1)$をもつB_vの弧とする．

(Case A)　$a\in A_s$ $(k=0)$：sは出口になることは許されていないので，$0\leqq b\leqq c^+(a)$の場合は次式が成り立つ．

$$r_a[b,0]=0$$

またその他の場合，$\tilde{D}_a$は$[b,k]$-実行可能でない．それ故次式が成り立つ．

$$r_a[b,0]=\begin{cases}0 & 0\leqq b\leqq c^+(a)\\ -\infty & \text{その他の場合}\end{cases}$$

また，$\tilde{D}_a$が$[b,k]$-実行可能となるbに対して，$\tilde{D}_a$の最大$[b,k]$-回収集合は空集合である．

(Case B)　$a\notin A_s$：b_1,b_2を，$b_1+b_2=b$かつ$C^-(a')\leqq b_1\leqq C^+(a')$かつ$c^-(a)\leqq b_2\leqq c^+(a)$を満たす整数とする．$k_1,k_2$を，$k_1+k_2=k$かつ$0\leqq k_1\leqq\min\{p,|W(a')|\}$かつ$0\leqq k_2\leqq\min\{p,|U(a)|\}$を満たす整数とする．ここでは簡単のため$D_1=\tilde{D}_{a'}$，$D_2=D_a$とおく．$r_a[b,k]$の値は$r_{a'}[b_1,k_1]$と$q_a[b_2,k_2]$の値を用いて求めることができるが，これから次の命題を証明する．

[命題4]　$\tilde{D}_a$が$[b,k]$-実行可能で，b_1,b_2,k_1,k_2は，D_1が$[b_1,k_1]$-実行可能で，D_2も$[b_2,k_2]$-実行可能となる値をとるとすると，

$$r_a[b,k]=\max_{k_1,k_2,b_1,b_2}\{r_{a'}[b_1,k_1]+q_a[b_2,k_2]\}$$

である．

(証明)　b'_1,k'_1,b'_2,k'_2を$\max\{r_{a'}[b_1,k_1]+q_a[b_2,k_2]\}=r_{a'}[b'_1,k'_1]+q_a[b'_2,k'_2]$を満たす整数とする．$D_1$において出口集合$X_1(|X_1|=k'_1)$をもち$r_{a'}[b'_1,k'_1]=\mathrm{val}(g_1)$となる部分フロー$g_{(X_1,b'_1)}(=g_1)$が存在する．また$D_2$において出口集合$X_2(|X_2|=k'_2)$をもち，$q_a[b'_2,k'_2]=\mathrm{val}(g_2)$となる部分フロー$g_{(X_2,b'_2)}(=g_2)$も存在する．$\tilde{D}_a$における部分フロー$g_{(X_1\cup X_2;b)}(=g)$を次のように定義する．

$$g(e)=\begin{cases}g_1(e) & e\in A(D_1)\\ g_2(e) & e\in A(D_2)\end{cases}$$

このとき次式が成り立つ．

$$\begin{aligned}&\max\{r_{a'}[b_1,k_1]+q_a[b_2,k_2]\}\\&\quad=\mathrm{val}(g_1)+\mathrm{val}(g_2)\\&\quad=\mathrm{val}(g)\leqq r_a[b,k].\end{aligned}\tag{2}$$

一方，$\tilde{D}_a$において，出口集合X $(|X|=k)$をもち$r_a[b,k]=\mathrm{val}(g')$となる部分フロー$g'_{(X;b)}(=g')$が存在する．$g'|D_1$を$g'$の$D_1$へ，$g'|D_2$を$g'$の$D_2$への制限とすると次式が成り立つ．

$$\begin{aligned}r_a[b,k]&=\mathrm{val}(g')\\&=\mathrm{val}(g'|D_1)+\mathrm{val}(g'|D_2)\\&\leqq\max\{r_{a'}[b_1,k_1]+q_a[b_2,k_2]\}\end{aligned}\tag{3}$$

式(2)，(3)より$r_a[b,k]=\max\{r_{a'}[b_1,k_1]+q_a[b_2,k_2]\}$を得る．　□

D_1が$[b_1,k_1]$-またはD_2が$[b_2,k_2]$-実行可能でないときは，$r_{a'}[b_1,k_1]+q_a[b_2,k_2]=-\infty$である．また逆に，それらが$[b,k]$-実行可能でなければ，同時に$[b_1,k_1]$-と$[b_2,k_2]$-実行可能になることはあり得ない．それ故，命題4は，D_1が$[b_1,k_1]$-またはD_2が$[b_2,k_2]$-実行可能でないことを許しても成り立つ．従って，次の結果が得られた．

$$r_a[b,k]=\begin{cases}\displaystyle\max_{k_1,k_2,b_1,b_2}\{r_{a'}[b_1,k_1]+q_a[b_2,k_2]\}\\ \qquad C^-(a)\leqq b\leqq C^+(a)\\ -\infty\quad\text{その他の場合}\end{cases}$$

b'_1,b'_2,k'_1,k'_2を$r_a[b,k]=r_{a'}[b'_1,k'_1]+q_a[b'_2,k'_2]$を満たす整数とする．$Q_a[i,j]$は$D_2$の最大$[i,j]$-回収集合を表すものとすると，$\tilde{D}_a$が$[b,k]$-実行可能となる$b$に

対して，$R_{a'}[b'_1, k'_1] \cup Q_a[b'_2, k'_2]$ は$\tilde{D}_a$の最大 $[b,k]$-回収集合である．

これらの結果を用いて，木状ネットワークの p-回収問題の疑多項式時間アルゴリズム 2 を構成する．すべての $a=(u,v) \in A' - A_s$に対して a^*を B_uのラベル $|B_u|$ をもつ弧としたとき，そのアルゴリズムは q_aを r_{a^*}を用いて計算し，すべての $a=(u,v) \in A' - A_s$に対して a'を $(\sigma_v(a)-1)$ のラベルをもつ B_vの弧としたとき r_aを，q_aと $r_{a'}$を用いて計算する．すなわち，最初は $L(l)$ の点から接続するすべての弧 a_lに対する r_{a_l} の値を計算し，次に $L(l-1)$ の点から接続するすべての弧 a_{l-1}に対して $q_{a_{l-1}}$の値を計算し，次に $r_{a_{l-1}}$の値を計算し，次に $L(l-2)$ の点から接続するすべての弧 a_{l-2}に対して $q_{a_{l-2}}$の値を計算し，次に $r_{a_{l-2}}$の値を計算し，以下同様にこれを繰り返し，最後に $r_{a_0}[0,k]$ の値を計算する．次に木状ネットワークの p-回収問題を解くアルゴリズム 2 を示す．

[アルゴリズム 2] 前出の結果を使うと $r_a[b,k]$, $q_a[b,k]$ を求める手続きを構成するのは簡単なので，それらは詳しく書かない．このアルゴリズムが停止したときの $Q_{a_0}[0,p]$ は Tの最大 p-回収集合で，$q_{a_0}[0,p]$ はその回収数である．

```
begin
  i ← l − 1;
  while i ≧ 0 do begin
    for all v ∈ L(i) do begin
      j ← 1;
      while j ≦ |Bv| do begin
        a ← σ_v^{-1}(j);
        for all b, k do
          if |W(a)| − k ≦ |V| − p then
            calculate r_a[b, k] and R_a[b, k];
        j ← j + 1;
      end;
      for all b do begin
        a ← (v, n(v));
        a* ← the last arc of B_v;
        if |U(a)| ≦ |V| − p then
          calculate q_a[b, 0] and Q_a[b, 0];
        for all max{1, p − |V| + |U(a)|} ≦ k < |U(a)|
        do begin
          calculate q'_a[b, k] and q''_a[b, k];
          if q'_a[b, k] > q''_a[b, k] then begin
            q_a[b, k] ← q'_a[b, k];
            Q_a[b, k] ← Q_{a*}[b*_a(b, k), k − 1] ∪ {v};
          end;
          else begin
            q_a[b, k] ← q''_a[b, k];
            Q_a[b, k] ← Q_{a*}[b, k];
          end;
        end;
        calculate q_a[b, |U(a)|] and Q_a[b, |U(a)|];
      end;
    end;
    i ← i − 1;
  end; a* = the last arc of B_{v0};
  calculate q'_{a0}[0, p] and q''_{a0}[0, p];
  if q'_{a0}[0, p] > q''_{a0}[0, p] then begin
    q_{a0}[0, p] ← q'_{a0}[0, p];
    Q_{a0}[0, p] ← R_{a*}[b*_{a0}(0, p), p − 1] ∪ {v0};
  end;
  else q_{a0}[0, p] ← q''_{a0}[0, p] and Q_{a0}[0, p] ← R_{a*}[0, p];
  end.
```

例として表 1 に $p=2$ で入力に図 2(a) のネットワークを与えたときのアルゴリズム 2 の動作を示す．そこには各 $a \in A'$が計算される順序 ((1.1)–(1.10)) で，r_a, R_a, q_a, Q_qの表が書かれている．例えば (1.1)r_a/R_aにおける $b=1, k=0$ の欄の $0/\emptyset$ は $r_a[1,0]=0/R_a[1,0]=\emptyset$ であること示し，$-\infty$ が書いてある欄は，そのとき $[b,k]$-実行可能でないことを示す．表 (1.9), (1.10) より，$q_h[0,2]=4$ かつ $Q_h[0,2]=\{A,D\}$ が求めることができる．それ故 $h(2)=4$ で，最大 2-回収集合は $\{A,D\}$ である．

アルゴリズム 2 の時間計算量について考える．r, C と n をそれぞれ最大次数，弧の容量と点の重みでの中で最大の整数，Tの点数とする．$q_a[b,k]$ を計算する手間は $O(rC)$，$R_a[b,k]$ を計算する手間は $O(prC)$ である．よってアルゴリズム 2 の計算量は $O(n(prC)^2)$ である．すなわち $O(p^2n^3C^2)$ である．木状ネットワークに対しては NP-完全であるが，このような疑多項式時間アルゴリズムが存在する．従って，それが強 NP-完全でないことがわかる．

5. むすび

本論文ではネットワークのロケーション問題の一つである p-回収問題を提案した．次に木状ネットワークに対する 1-回収問題の $O(n)$ アルゴリズム，p-回収問題問題の疑多項式時間アルゴリズムを示した．それから木に対する p-回収問題問題に対応する判定問題が強 NP-完全でないことがわかった．特に疑多項式時間アルゴリズムは Cが小さい値のとき十分速く走るし，これから良い近似アルゴリズムを構成できる．この問題はコストより通信トラヒックの問題が重要な場合におけるコンピュータネットワーク上の資源配置の最適化などに応用が期待される．

また木状ネットワークにおける p-回収問題は疑多項式時間アルゴリズムが存在したがそれは非常に高いオーダであったのそれを下げる必要がある．更に，一般のネットワークについての自明でない効率の良い 1-回収問題のアルゴリズムを求めていきたい．更に p-回

表 1 アルゴリズム 2 の動作例

Table 1 An example of behavior of Algorithm 2.

(1.1) r_a/R_a

b	k		
	0	1	2
−3	$-\infty$	—	—
−2	$-\infty$	—	—
−1	$-\infty$	—	—
0	$0/\emptyset$	—	—
1	$0/\emptyset$	—	—
2	$0/\emptyset$	—	—
3	$-\infty$	—	—

(1.2) r_b/R_b

b	k		
	0	1	2
−3	$-\infty$	—	—
−2	$-\infty$	—	—
−1	$-\infty$	—	—
0	$0/\emptyset$	—	—
1	$-\infty$	—	—
2	$-\infty$	—	—
3	$-\infty$	—	—

(1.3) q_d/Q_d

b	k		
	0	1	2
−3	$-\infty$	$-\infty$	—
−2	$-\infty$	$-\infty$	—
−1	$-\infty$	$3/\{A\}$	—
0	$0/\emptyset$	$2/\{A\}$	—
1	$0/\emptyset$	$1/\{A\}$	—
2	$-\infty$	$-\infty$	—
3	$-\infty$	$-\infty$	—

(1.4) q_e/Q_e

b	k		
	0	1	2
−3	$-\infty$	$-\infty$	—
−2	$-\infty$	$2/\{B\}$	—
−1	$-\infty$	$1/\{B\}$	—
0	$0/\emptyset$	$0/\{B\}$	—
1	$-\infty$	$-\infty$	—
2	$-\infty$	$-\infty$	—
3	$-\infty$	$-\infty$	—

(1.5) r_c/R_c

b	k		
	0	1	2
−3	$-\infty$	—	—
−2	$-\infty$	—	—
−1	$-\infty$	—	—
0	$0/\emptyset$	—	—
1	$0/\emptyset$	—	—
2	$-\infty$	—	—
3	$-\infty$	—	—

(1.6) r_d/R_d

b	k		
	0	1	2
−3	$-\infty$	$-\infty$	—
−2	$-\infty$	$-\infty$	—
−1	$-\infty$	$3/\{A\}$	—
0	$0/\emptyset$	$3/\{A\}$	—
1	$0/\emptyset$	$2/\{A\}$	—
2	$0/\emptyset$	$1/\{A\}$	—
3	$-\infty$	$-\infty$	—

(1.7) r_e/R_e

b	k		
	0	1	2
−3	$-\infty$	$-\infty$	$5/\{A,B\}$
−2	$-\infty$	$2/\{B\}$	$5/\{A,B\}$
−1	$-\infty$	$3/\{A\}$	$4/\{A,B\}$
0	$0/\emptyset$	$3/\{A\}$	$3/\{A,B\}$
1	$0/\emptyset$	$2/\{A\}$	$2/\{A,B\}$
2	$0/\emptyset$	$1/\{A\}$	$1/\{A,B\}$
3	$-\infty$	$-\infty$	$-\infty$

(1.8) q_g/Q_g

b	k		
	0	1	2
−3	—	$-\infty$	$-\infty$
−2	—	$-\infty$	$-\infty$
−1	—	$-\infty$	$-\infty$
0	—	$3/\{A\}$	$3/\{A,B\}$
1	—	$2/\{A\}$	$2/\{A,B\}$
2	—	$1/\{A\}$	$1/\{A,B\}$
3	—	$-\infty$	$-\infty$

(1.9) r_f/R_f

b	k		
	0	1	2
−3	$-\infty$	—	—
−2	$-\infty$	—	—
−1	$-\infty$	—	—
0	$0/\emptyset$	—	—
1	$0/\emptyset$	—	—
2	$-\infty$	—	—
3	$-\infty$	—	—

(1.10) r_g/R_g

b	k		
	0	1	2
−3	—	$-\infty$	$-\infty$
−2	—	$-\infty$	$-\infty$
−1	—	$-\infty$	$-\infty$
0	—	$3/\{A\}$	$3/\{A,B\}$
1	—	$3/\{A\}$	$3/\{A,B\}$
2	—	$2/\{A\}$	$2/\{A,B\}$
3	—	$1/\{A\}$	$1/\{A,B\}$

収問題の近似アルゴリズムを考えていくつもりである.

文　献

(1) Ahuja R.K., Magnanti T.L. and Orlin J.B.: "Network Flows", Prentice Hall (1993).

(2) Kariv O. and Hakimi S.L.: "An Algorithmic Approach to Network Location Problems. I: The p-Centers", SIAM J. Appl. Math., **37**, 3, pp.513–538 (1979).

(3) Kariv O. and Hakimi S.L.: "An Algorithmic Approach to Network Location Problems. II: The p-Medians", SIAM J. Appl. Math., **37**, 3, pp.539–560 (1979).

(4) Tamura H., Sengoku M., and Shinoda S. and Abe T.: "Location Problems on Undirected Flow Networks", Trans. IEICE, **E73**, 12, pp.1989–1993 (1990).

(5) Watanabe K., Tamura H. and Sengoku M.: "Problems of Where to Locate p-Sinks in a Flow Network", Proc. 7th Karuizawa Circuits and Systems Workshop, IEICE, pp.339–344 (1994).

(6) Watanabe K., Tamura H. and Sengoku M.: "Problems of Where to Locate p-Sinks in a Flow Network: Complexity Approach", Proc. 8th Karuizawa Circuits and Systems Workshop, IEICE (1995).

(7) Goldberg A.V. and Tarjan R.E.: "A new approach to the maximum flow problem", Proc. the 18th Annual ACM Symposium on Theory of Computing, pp.136–146 (1986).

(8) Garey M.R. and Johnson D.S.: "Computers and Intractability: A Guide to the Theory of NP-completeness", Freeman, San Francisco (1979).

(平成 6 年 12 月 16 日受付，7 年 4 月 13 日再受付)

渡辺　郁

平 2 新潟大・工・情報卒．平 4 同大大学院工学研究科修士課程了．現在，同大学院自然科学研究科博士課程在学中．グラフネットワークの研究に従事．

田村　裕

昭 57 新潟大・教育卒．昭 61 同大大学院理学研究科修士課程了．平 2 同大学院自然科学研究科博士課程了．学術博．同年同大大学院自然科学研究科助手．平 3 同大地域共同研究センター助教授．現在，新潟工科大学助教授．グラフ理論とその応用，計算幾何学とその応用の研究に従事．平 3 年度論文賞受賞．

仙石　正和

昭 42 新潟大・工・電気卒．昭 47 北大大学院博士課程了．工博．同年北大・工・電子助手．新潟大・工・情報助教授を経て，現在，同教授．回路網理論，グラフネットワーク理論，情報伝送特に移動通信の研究に従事．平 3 年度論文賞受賞．IEEE，情報処理学会会員．著書「演習グラフ理論」(共著)．