2003年日本オペレーションズ・リサーチ学会
秋季研究発表会

最大利益根付木問題のアルゴリズム

01102100 法政大学 *古林 隆 KOBAYASHI Takashi
01502860 法政大学 福馬 敏子 FUKUMA Toshiko
明生システムサービス 榊原 英行 SAKAKIBARA Hideyuki
富士フィルム 露木 真理子 TSUYUKI Mariko

1. はじめに

ケーブルテレビのように，センターと利用者をケーブルなどでつなぐことにより提供できるサービスがある．それらをつなぐにはコストがかかる．すべての利用者にサービスを提供しなければならない場合には，コストの最小化を目的とすればよいので，どこをつなぐかは，最小極大木(minimum spanning tree)問題になる．しかし，その必要がない場合は，利用者から得る収入とつなぐためのコストを比べて，つなぐところを決めればよい．これは，センターを根とする木の中で，利益を最大にするものを求める問題になる．ここでは，その近似最適解を求めるアルゴリズムを提案する．

2. 最大利益根付木問題

無向連結グラフ $G=(N,A)$が与えられているとする．点1をセンターとし，点 iには，収入 p_iが与えられている($p_1=0$とする)．点 iと点 jを結ぶ枝を$\{i,j\}$で表す．各枝$\{i,j\}$には，コスト c_{ij}が与えられている．

このとき，点1を根とする木 $G_T=(N_T,A_T)$の中で，利益

$$z=\sum_{i\in N_T} p_i-\sum_{\{i,j\}\in A_T} c_{ij}$$

を最大にするものを求める最大利益根付木問題を考える．

3. アルゴリズム

各点に，その点を根とする木を付随させる．最初はそれぞれの点だけとする．また，木の利益は，それに含まれる点の収入の和から点をつなぐ枝のコストの和を引いた値とする．枝に向きをつけ，その終点の付随木の利益から，その枝のコストを引いた値を枝の重みとする．

(1) 未選択の枝の中から重みが最大である枝を選択する．重みが正でなければ終了．
(2) その枝の始点を付随木に含んでいる点に対しては，その付随木に，選択した枝を付け加えて，終点の付随木を結合する．結合によって閉路ができる場合は，終点の付随木から枝の一部を除く．
(3) 未選択の枝が残っていれば，(1)にもどる．

ここで，点 i の付随木を $G_i=(S_i,T_i)$，その利益を p_i^*で表すことにする．

＜アルゴリズムの詳細＞

手順0（初期値設定）

$$S_i=\{i\} \quad (i\in N),$$
$$T_i=\phi \quad (i\in N),$$
$$p_i^*=p_i \quad (i\in N),$$
$$H=\{(i,j)\mid\{i,j\}\in A, j\neq 1\}$$

とする．

手順1

$\max_{(g,h)\in H}(p_h^*-c_{gh})=p_j^*-c_{ij}$ である$(i,j)\in H$ を求める．

$p_j^*-c_{ij}\le 0$ であれば終了する．

$j\in S_i$ であれば手順3へ．

手順2

$i\in S_k$ であるすべての k に対して，

1) $S_k\cap S_j=\phi$ ならば，

$$S_k=S_k\cup S_j,$$
$$T_k=T_k\cup T_j\cup\{\{i,j\}\},$$
$$p_k^*=p_k^*+p_j^*-c_{ij}$$

とする．

2) $S_k\cap S_j\neq\phi$ ならば，S_j,T_jから一部を除いて，

$$S_k\cap S_j'=\phi,$$
$$S_j'\neq\phi\Rightarrow j\in S_j',$$
$$T_j'\subset T_j\cup\{i,j\},$$
$$T_j'\neq\phi\Rightarrow\{i,j\}\in T_j'$$

を満たす木$G=(S_k\cup S_j',T_k\cup T_j')$の中で，利益が大きいものを求める．

$$S_k=S_k\cup S_j',$$
$$T_k=T_k\cup T_j',$$
$$p_k^*=p_k^*+\sum_{g\in S_j'} p_g-\sum_{\{g,h\}\in T_j'} c_{gh}$$

とする．($S_j'=\phi$のときは，S_k，T_k，p_k^*は変わ

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

らない.)

手順 3

H から(i, j)を除き，H が ϕ でなければ手順 1 にもどり，ϕ であれば終了する.

終了したとき，点 1 の付随木を解とする.

手順 2 の 2) の詳細は，次のとおりである.

① $W_A = \{(x,y) | \{x,y\} \in T_j, x$の方が$y$より$j$に近い$, y \notin S_k\} \cup \{(i,j)\}$,
$W_B = \phi$
とおく.

② $p_k^* = p_k^* + function1(i)$ とする.

③ $function2(i)$ を実行する.

$function1(e)$

① pp=0 とする.

② $(e,f) \in W_A$であるすべてのfについて以下を行う.
$ppe = p_f - c_{ef} + function1(f)$ とする.
$ppe > 0$ ならば,
$W_B = W_B \cup \{(e,f)\}$,
$pp = pp + ppe$
とする.

③ pp を返す.

$function2(e)$

$(e,f) \in W_B$であるすべてのfについて以下を行う.

① $S_k = S_k \cup \{f\}$,
$T_k = T_k \cup \{\{e,f\}\}$
とする.

② $function2(f)$ を実行する.

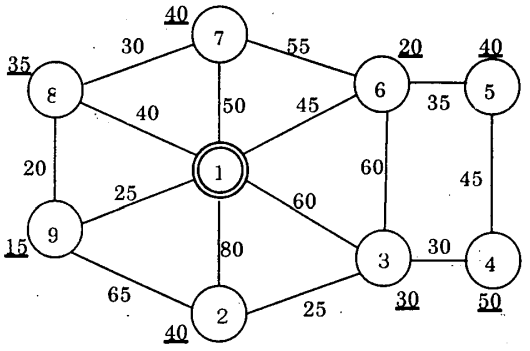
常に$i \in S_i$であり，向きを考慮しているから，$j \in S_i$であっても$i \in S_j$とは限らない.

4. 実行例

図 1 にネットワークの例を示す. 手順 1 で選ばれる(i, j), $p_j^* - c_{ij}$ は, 表 1 のとおりである. 図 2 に, 得られた根付木を示す. $z = 20$ となった.

5. おわりに

提案したアルゴリズムの特徴は, 枝に向きをつけて枝を延ばしやすくし, さらに先まで延ばすと利益が出る場合を見逃さないようにしたことである. ある点と接続する点を直接つなぐと損であっても, 他の点を接続した後でつなげば利益がでる場合を見落とさないようになっている.



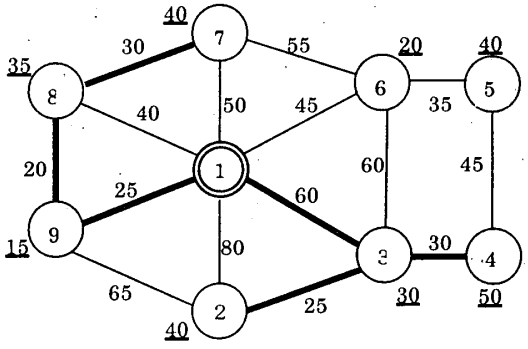
点に付した数値は収入, 枝に付した数値はコストを示す.

図 1. ネットワークの例

表 1. 手順 1 で選ばれた枝と $p_j^* - c_{ij}$

反復	(i,j)	$p_j^* - c_{ij}$	反復	(i,j)	$p_j^* - c_{ij}$
1	(3,4)	20	9	(5,6)	10
2	(2,3)	25	10	(8,7)	10
3	(3,2)	40	11	(8,9)	20
4	(4,3)	35	12	(1,9)	15
5	(5,4)	20	13	(7,8)	15
6	(6,5)	25	14	(1,3)	5
7	(4,5)	15	15*	(1,8)	5
8	(9,8)	15	16*	(6,3)	5

*は手順 2 を実行しない反復



太線は A_T に含まれる枝を示す.

図 2. 得られた根付木

しかし, このアルゴリズムでは必ずしも最適解が得られるとは限らない.

参考文献

1) Ahuja,R.K. et al. : Applications of Network Optimization , Handbooks in Operations Research and Management Science Vol.7 , Elsevier ,1995.

2) 伊理正夫, 古林 隆 : ネットワーク理論, 日科技連出版, 1976.

© 日本オペレーションズ・リサーチ学会. 無断複写・複製・転載を禁ず.