

最大利益根付木問題のアルゴリズム

古林, 隆 / KOBAYASHI, Takashi / FUKUMA, Toshiko / 福馬,
敏子

(出版者 / Publisher)

法政大学工学部

(雑誌名 / Journal or Publication Title)

法政大学工学部研究集報 / Bulletin of the Technical College of Hosei
University

(巻 / Volume)

41

(開始ページ / Start Page)

31

(終了ページ / End Page)

34

(発行年 / Year)

2005-03

(URL)

<https://doi.org/10.15002/00003762>

最大利益根付木問題のアルゴリズム

ALGORITHMS FOR THE MAXIMUM PROFIT ROOTED-TREE PROBLEM

古林 隆*, 福馬 敏子*

Takashi KOBAYASHI and Toshiko FUKUMA

There exists a utility which can be provided by connecting users to a center with such like cables, for example CATV. Here, one of problems is which to select among user candidates and how to connect them such that the profit is maximized. It is called the Maximum Profit Rooted-tree problem. The algorithm for obtaining a near optimal solution by repeating to connect trees associated with nodes has been proposed, but it is undesirable in the point of computing time. So, we present a new algorithm by which computing time can be reduced. It consists of repeating to get the maximum weighted paths. We compare two algorithms for values of the objective function and computing times in many cases.

Key Words : Maximum Profit Rooted- tree, maximum weighted path

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連結することにより提供できるサービスがある。それらを連結する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にすればよいので、最小極大木(minimum spanning tree)問題になる[2]が、その必要がない場合は、利用者から得る収入と連結するためのコストを比べて、連結する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最大利益根付木問題になる[3]。その近似最適解を求めるアルゴリズムとして、古林等[3]は、各点に木を付随させて、それらを連結していくことを繰り返していく付随木連結法を提案し、阿部等[1]が改良しているが、計算量の点からあまり望ましいものではない。そこで、最大重み経路を求めることを繰り返して近似最適解を得ることによって、計算量を減少させるアルゴリズム—最大重み経路法を提案し、両者を比較する。

2. 最大利益根付木問題

点の集合を N , 無向枝の集合を A とする無向連結グラフ $G=(N, A)$ が与えられている。点 l をセンターとし、点 i の収入を $P_i (P_l=0)$, 点 i と点 j を結ぶ枝 $\{i, j\}$ のコストを $C_{ij} (C_{ij}=C_{ji}>0)$ とする。

このとき、点 l 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のが最大利益根付木問題である。ここで、木の枝に根の近い方から遠い方へ向きをつけることによって得られる有向枝の集合を A_{Td} とすると、

$$Z = \sum_{(i, j) \in A_{Td}} (P_j - C_{ij})$$

となる。

3. 付随木連結法

アルゴリズムの概要は、次のとおりである。

各点 i に、それを根とする木 (S_i, T_i) を付随させる。最初は、それぞれの点だけとする。また、付随木の利益は、

$$Z_i = \sum_{h \in S_i} P_h - \sum_{\{g, h\} \in T_i} C_{gh}$$

とする。枝 $\{i, j\}$ を 2 本の有向枝 (i, j) と (j, i) に分け、 (i, j) の重みを $z_j - C_{ij}$ とする。

手順 1. $S_i = \{i\}$, $T_i = \phi$, $z_i = P_i (i \in N)$,

$A_d = \{\{i, j\} \mid \{i, j\} \in A, j \neq l\}$ とする。

手順 2. A_d の中で、 $z_j - C_{ij}$ を最大にする (i, j) を求める。

$z_j - C_{ij} \leq 0$ であれば手順 5 へ。 $j \in S_i$ であれば、手順 4 へ。

手順 3. すべての $h \in S_i$ に対して、枝 (i, j) を加えることによって、点 h の付随木に点 j の付随木を連結させる。ただし、これによって、閉路ができるときは、点 j の付随木の一部を除くことにする。

手順 4. A_d から (i, j) を除き、 A_d が ϕ でなければ手順 2 に

* 経営工学科

もどり、 ϕ であれば手順5へ。

手順5. $N_T = S_1$ とする。元のグラフ G で枝 $\{i,j\}$ の長さを C_{ij} として、 N_T を張る最小木を求めて、それを $G_T = (N_T, A_T)$ とする。

4. 最大重み経路法

最大重み経路法は、5段階(手順)に分かれている。最大重み経路(それに含まれる枝の重みの和を最大にする経路)を求めることを核とするが、予め最大重み経路が存在するように、すなわち、正の重みの閉路が存在しないようにしておく必要がある。手順1, 2は、そのためのものである。手順3, 4で、 N_T が定まるが、それまでに得られた A_T は、最小のコストで N_T を張っているとは限らないので、手順5で、 N_T を張る最小木を求めることにする。

手順1. $P_i \geq C_{ij}$ かつ $P_j \geq C_{ij}$ である点 i と点 j が存在すれば、それらを併合する。 P の値を $P_i + P_j - C_{ij}$ とし、点 i との間にも点 j との間にも枝が存在する点に対しては、コストが大きい方の枝を除く。

併合後のグラフを $G_w = (N_w, A_w)$ とし、点 i ($i \in N_w$) に併合されている点の集合(i を含む)を S_i とする。 ($S_1 = \{1\}$ である。)

手順2. 枝 $\{i,j\}$ ($i < j$) の両方向の重みを決定する。

i から j への向きを $w_{ij} = P_j - C_{ij}$, j から i への向きの重みを $w_{ji} = P_i - C_{ij}$ とする。

ただし、 $i=1$ または $w_{ij} > 0$ であれば、 $w_{ij} = -\infty$ とし、 $w_{ji} > 0$ であれば、 $w_{ji} = -\infty$ とする。

手順3. $N_{w1} = \{1\}$ とする。

$N_{w0} = N_w - N_{w1}$ が空になるまで、以下の(3.1)から(3.4)を繰り返す。

(3.1) N_{w1} の大きさが2以上のときは、 N_{w1} の2点間を結ぶ枝に対しては、重みはすべて0であるとする。

N_{w1} (のいずれかの点) から各点 j ($j \in N_{w0}$) への最大重み経路を求め、その重みを v_j とする。

(3.2) v_j を最大にする j (の一つ) を j_{max} , 点 j_{max} への経路を $R_{j_{max}}$, $R_{j_{max}}$ に含まれていて、 N_{w1} に含まれない点の集合を N_{WR} とする。

(3.3) N_{w1} に N_{WR} を加える。

(3.4) $i \in N_{WR}$, $j \in N_{w0} - N_{WR}$ である枝 $\{i,j\}$ が存在すれば、 $w_{ji} = -\infty$ とし、 $w_{ij} = -\infty$ であれば $w_{ij} = P_j - C_{ij} (< 0)$ とする。

最大重み経路 $R_{j_{max}}$ をすべてつなぐことによって N_w を張る木が得られる。

手順4. 手順3で求めた N_w を張る木から利益を減少させる枝を“切り落とす”。

その重みと(根の方から見て)それより先の枝の重

みの和が負である枝を除く。

点1とつながっている点の集合を N_{WT} とし、

$$N_T = \sum_{i \in N_{WT}} S_i$$

とする。

手順5. 元のグラフ G で枝 $\{i,j\}$ の長さを C_{ij} として、 N_T を張る最小木を求めて、それを $G_T = (N_T, A_T)$ とする。

[正の重みの閉路が存在しないことの証明]

任意の閉路を L とし、 L に含まれる点の集合を N_L , 向きも付けた枝の集合を A_L , 重みを $w(L)$ とする。

$P_i > C_{ij}$ である $(i,j) \in A_L$ が存在すれば、 $w_{ij} = -\infty$ より、 $w(L) = -\infty$ 。

すべての $(i,j) \in A_L$ に対して $P_i \leq C_{ij}$ であれば、 $w(L) \leq \sum (P_j - C_{ij}) = \sum P_j - \sum C_{ij} = \sum (P_i - C_{ij}) \leq 0$ 。

(証明終)

5. 最大重み経路法の実行例

図1にグラフの例を示す。手順1で、点9は、点8に併合され、さらに、点11が併合されるので、

$$N_w = \{1, 2, 3, 4, 5, 6, 7, 8, 10, 12\},$$

$$S_8 = \{8, 9, 11\}$$

となる。

手順3で求められた N_w を張る木を図2に示す。手順4で、点6, 7, 10, 12が除かれるので、

$$N_{WT} = \{1, 2, 3, 4, 5, 8\},$$

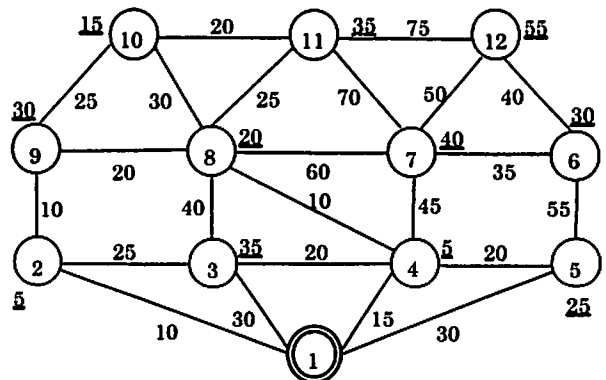
$$N_T = \{1, 2, 3, 4, 5, 8, 9, 11\}$$

となる。

手順5で求められた、 N_T を張る最小木を図3に示す。

$$Z = \sum_{i \in N_T} P_i - \sum_{\{i,j\} \in A_T} C_{ij} = 155 - 110 = 45$$

となった。



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 グラフの例

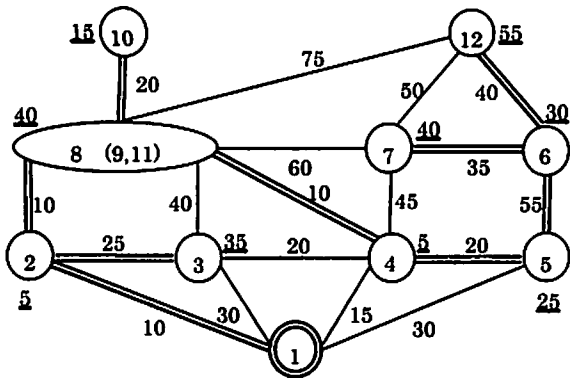


図2 N_w を張る木

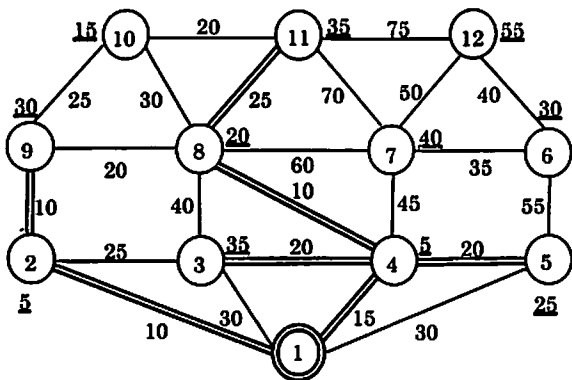


図3 N_T を張る最小木

6. 数値実験の結果

いろいろなグラフで、 P と C に一樣乱数を与えて数値実験を行い、目的関数値および計算時間の比較を行った。

グラフとしては、次の3種類を用いた。

G1: 阿部, 市川[3]が採用したグラフ

点の数 n は 300 まで任意に設定できる。

G2: 一辺の点の数が奇数である正方形の格子グラフでセンターを中央に置いたグラフ

G3: G2 でセンターを角に置いたグラフ

P には、後に示す(5)を除いて、1 から 80 まで的一様乱数、 C には 1 から 100 まで的一様乱数を与えることにした。

また、いずれの場合も両方のアルゴリズムで解くことを 100 回繰り返した。使用した計算機は dynabook PAEX1522PDETW である。

以下では、付随木連結法、最大重み経路法をそれぞれ番号 1, 2 で区別する。また、最大重み経路法の目的関数値(z_2)の付随木連結法のそれ(z_1)に対する減少率 $(z_1 - z_2) / z_1$ を drz 、計算時間を t で表す。

(1) G1 における目的関数値の比較

点の数を $n=100$ (枝の数 $m=207$) としたときの drz の分布を表 1 に示す。

表 1 G1 における目的関数値の比較

$drz(\%)$	回数
-5 ~ 0	14
0	2
0 ~ 5	67
5 ~ 10	14
10 ~ 15	3
計	100
平均	2.9
最大	13.0

$drz = (z_1 - z_2) / z_1$, z_1 : 付随木連結法, z_2 : 最大重み経路法

(2) G1 における計算時間の比較

G1 で n を 50 から 300 まで変化させて、計算時間 t を測定した。100 回の平均を表 2 に示す。 n と t の両方の対数をとって、直線をあてはめた結果、

付随木連結法 $t_1 = 0.0323n^{2.88} (\mu \text{ sec})$

最大重み経路法 $t_2 = 0.1260n^{1.80} (\mu \text{ sec})$ を得た。

表 2 G1 における平均計算時間(msec)

n	m	t_1	t_2
50	98	2.667	0.149
70	141	6.632	0.263
100	207	18.497	0.490
150	316	58.273	0.982
200	432	143.651	1.709
300	648	456.981	3.807

t_1 : 付随木連結法, t_2 : 最大重み経路法

(3) グラフの形状による違い

G2 で $n=121$ (枝の数 $m=220$) としたときの drz の分布を表 3 に示す。また、 n を 49 から 289 まで変化させたときの t を表 4 に示す。これより、

付随木連結法 $t_1 = 0.0364n^{2.82} (\mu \text{ sec})$

最大重み経路法 $t_2 = 0.1067n^{1.84} (\mu \text{ sec})$ を得た。

グラフの形状による違いは、ほとんど見られなかった。

(4) センターの位置による違い

G2, G3 で drz の平均と平均計算時間を比較した。結果を表 5 に示す。最大重み経路法は、センターが中央にある場合に多少よい解が得られるという傾向が見られるが、ほとんど違いはなかった。

(5) P の分布による違い

G1 で P の分布の最大値 $pmax$ を 80 から 50 まで変化させたときの drz の平均と平均計算時間を比較した。結果を表 6 に示す。 $pmax$ が小さくなるに従って、 t_1 が小さくなるのは、各点の付随木の大きさ(点の数)が小さくなるからと思われる。逆に、 t_2 が大きくなるのは、手順 1 で点

の併合が少なくなり、手順3で計算するときの点の数 n_w が大きくなるからと考えられる。

表3 G2における目的関数値の比較

drz(%)	回数
-10 ~ -5	1
-5 ~ 0	9
0	1
0 ~ 5	58
5 ~ 10	29
10 ~ 15	2
計	100
平均	3.7
最大	11.7

$drz = (Z_1 - Z_2) / Z_1$, Z_1 : 付随木結合法, Z_2 : 最大重み経路法

表4 G2における平均計算時間(msec)

n	m	t_1	t_2
49	84	2.215	0.146
81	144	8.631	0.343
121	220	26.726	0.694
169	312	69.051	1.324
225	420	159.146	2.338
289	544	330.201	3.812

t_1 : 付随木結合法, t_2 : 最大重み経路法

表5 センターの位置による違い

n	G2(中央)			G3(角)		
	t_1	t_2	drz(%)	t_1	t_2	drz(%)
49	2.215	0.146	2.4	2.280	0.141	4.1
121	26.726	0.699	3.7	26.846	0.678	3.9
225	159.146	2.338	3.8	159.781	2.286	4.0

いずれも 100 回の平均。

表6 Pの分布による変化

pmax	t_1	t_2	n_t	n_w	drz(%)
50	15.963	0.743	53,53	56	4.0
60	17.696	0.603	63,64	46	2.4
70	18.387	0.522	68,71	40	3.7
80	18.497	0.490	74,76	33	2.9

n_t : N_T の大きさ (前が付随木結合法), n_w : 最大重み経路法の手順1で残った点の数, いずれも 100 回の平均。

7. おわりに

最大重み経路法を用いることによって、目的関数の値は、多少悪くなることもあるが、計算量は、大幅に減少した。今後の課題としては、点の結合後の P の再計算などによって、最適解に近づける(目的関数の値を良くする)ことが考えられる。

参考文献

- [1]阿部壮紘, 市川加奈子: 最大根付木問題のアルゴリズム, 2003 年度経営工学科卒業論文, 2004.
- [2]伊理正夫, 古林隆: ネットワーク理論, 日科技連出版, 1976.
- [3]古林隆, 福馬敏子他: 最大利益根付き問題のアルゴリズム, 2003 年秋季研究発表会アブストラクト集, pp.166-167, 日本オペレーションズ・リサーチ学会, 2003.
- [4]Rao, V.V. and Sridharan, R.: Minimum-Weight Rooted Not-Necessarily-Spanning Arborescence Problem, Networks 39, pp.77-87, 2002.