

最大利益根付木問題のアルゴリズム II

—最大重み経路法—

01102100 法政大学 *古林 隆 KOBAYASHI Takashi
01502860 法政大学 福馬 敏子 FUKUMA Toshiko

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サービスを提供する場合に発生する問題に最大利益根付木問題がある。古林等[1]は、この問題の近似最適解を求めるアルゴリズムとして、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方法(付随木連結法と呼ぶことにする)を提案しているが、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 I をセンターとし、点 i の収入を $p_i(p_i=0)$ 、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 I 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。

3. アルゴリズム

アルゴリズムは、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。 ($S_i=\{I\}$ 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 への向きの重みを $w_{ij}=p_j - c_{ij}$,

j から i への向きの重みを $w_{ji}=p_i - c_{ij}$

とする。

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

手順3. $N_{W1}=\{I\}$ 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) $i \in N_{WR}$ 、 $j \in N_{W0} - N_{WR}$ 、 $w_{ij}=-\infty$ である枝 $\{i, j\}$ が存在すれば、 $w_{ij}=p_j - c_{ij}(<0)$ 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 I とつながっている点の集合を N_{WT} とし、

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

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)

4. 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,

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

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

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.

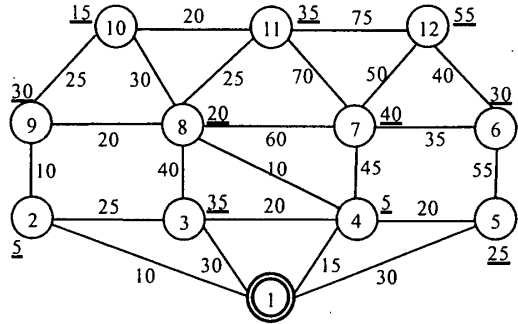
5. 数値実験の結果

点の数 $n=100$, 枝の数 $m=207$ のグラフで, p には, 1から80までの一様乱数, c には1から100までの一様乱数を与えることにして, 両方のアルゴリズムで解くことを100回繰り返した. 付随木連結法の目的関数値を z_1 , 最大重み経路法のそれを z_2 とし, $\Delta z = (z_1 - z_2) / z_1$ を求めたところ, 負になったのが14回, 0になったのが2回, 正で5%以下が67回であった. なお, 最大値は13%であった. 次に, 点の数を50から300まで変化させて, 各100回の平均の計算時間 t を比較した. n と t の両方の対数をとって, 直線をあてはめた結果,

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

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

を得た.



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

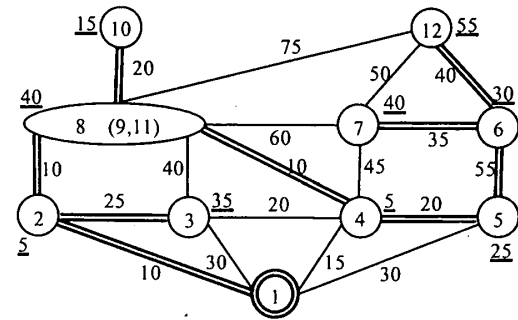


図2. N_W を張る木

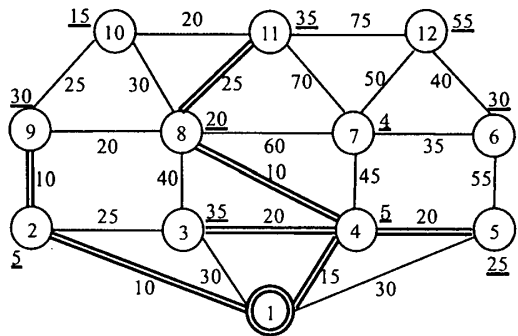


図3. N_T を張る最小木

6. 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, 多少悪くなることがあるが, 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] 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.
 [2] Rao, V.V. and Sridharan, R.: Minimum-Weight Rooted Not-Necessarily-Spanning Arborescence Problem, Networks 39, pp. 77-87, 2002.