A-031

最大利益根付木問題に対するヒューリスティックアルゴリズム

A Heuristic Algorithm for Maximum Profit Rooted Subtree Problem

千葉 英史* 古林 隆* 福馬 敏子*

Eishi Chiba Takashi Kobayashi Toshiko Fukuma

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を通して連結することで提供できるサービスがある．ケーブル連結のために，その長さに応じたコストがかかる．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，コスト最小化を目的にすればよいので，最小全域木問題として定式化できるが，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考慮して，連結するところを決めればよい．これはグラフ上の組合せ最適化問題として定式化できる．つまり，無向グラフが与えられたとき，点に付した収入和から枝に付したコスト和を引くことで得られる利益が最大となるような，センターを根とする（必ずしも全域である必要がない）木を求める問題である．この問題は，有向グラフ上の最大コスト有向木問題へと帰着できる．関連研究として，上述の最小コスト版を扱ったものが存在する ([1], [5])．

本研究では，ケーブル接続問題解決のために，上述の問題を考える．次にこの問題が NP 困難であることを示し（Node Weighted シュタイナー木問題 [2] からの帰着），ヒューリスティックアルゴリズムを与える．このアルゴリズムは，制限された有向グラフ上で，効率良く最長路を繰返し計算するのが特徴である．アルゴリズムは実装され，計算実験を通して，文献 [3] の手法と比較考察を行った．その結果，本手法が評価値に関して悪い結果を示す場合が見られたが，計算時間に関しては大幅に改善することが分かった．さらに，整数計画問題として定式化を行った．

2 問題の定式化

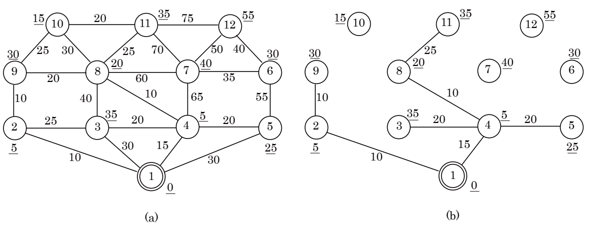
連結無向グラフ $G = (N, A)$，N はセンター及び複数のユーザー候補からなる集合，A は N の二つの要素を接続可能な枝からなる集合とする．各点 $i (\in N)$ は非負の利益 $p_i (\geq 0)$ を持ち，各枝 $\{j, k\} (\in A)$ は正コスト $c_{jk} (> 0)$ $(c_{jk} = c_{kj})$ を持つ．p_i はユーザー i からの収入に対応し，c_{jk} は点 j と点 k を接続するためのコスト（必要なケーブルの長さ）に対応する．さらに，グラフ G 中の点をセンターとユーザー候補で区別する：つまり，センターを $1 (\in N)$ として，ユーザー候補を $2, 3, \ldots, n$ $(\in N)$ で表す．また，センターにおける利益 $p_1 = 0$ と仮定する．以上の仮定で，次の評価関数

$$profit(T) = \sum_{i \in N_T} p_i - \sum_{\{i,j\} \in A_T} c_{ij}$$

を最大化するような点 1 を含む G の連結部分グラフ $T = (N_T, A_T)$ を求めたい．その結果，A_T に対応するケーブルを利用して，N_T に含まれるユーザーがセンターからのサービスを受けることになる．

点 1 を含む G の連結部分グラフが閉路を持たないならば，そのグラフは点 1 を根とする必ずしも全域である必要が無い木と呼ばれる．以下では簡単のために，根付き部分木と呼ぶ．$profit(T)$ を最大化する部分グラフ T は，根 1 を含む根付き部分木である．なぜなら，もしそうでないなら，T 中の閉路から任意の枝を取り除くことで，T の連結性を保ったまま，$profit(T)$ の値を増やすことができるからである．

入力としてグラフ G，利益 p_i，コスト c_{ij} が与えられたとき，$profit(T)$ を最大化するような根付き部分木 T を求める問題を，最大利益根付き木問題と呼ぶ．下図は，入力例と，それに対する最大利益を達成する根付き部分木を示す．



(a) は入力例で，丸と二重丸は，それぞれユーザー候補とセンターを表し，点上の収入と枝上のコストが示されている．(b) は最適解を示す．この根付き部分木 T の全体利益 $profit(T)$ は 45 である．そのような最大利益の根付き部分木は，一般に唯一ではない．

3 ヒューリスティックアルゴリズム

3.1 点の併合

G 中で $p_i \geq c_{ij}$ かつ $p_j \geq c_{ij}$ を満たす任意の 2 点 i, j を併合することから始める．併合とは，点 i と j を同一視して，新しい一点 k と見なして，点 k の利益 $p_k = p_i + p_j - c_{ij}$ を付し，新しいグラフ G' にすることを意味する．従って，それ以外の点及び利益はそのままである．その際，枝 $\{i, j\}$ は G' では除去される．端点の一方のみが i あるいは j に一致する G の枝は，G' ではその端点が新しい点 k に更新される．ただし，G' が点 k と接続する 2 本の並列枝を含む場合には，コストの大きい方の枝を除く．そのため，G' は多重グラフにはならない．それ以外の G の枝およびコストは G' でもそのままである．併合処理は，G' 中で，$p_i \geq c_{ij}$ かつ $p_j \geq c_{ij}$ を満たす任意の 2 点 i, j を併合し，再帰的に続ける．各点 k に併合された点集合を S_k と記す．

*法政大学 理工学部 経営システム工学科

Email: {e-chiba, t.kobaya, fukuma}@hosei.ac.jp

3.2　双方向有向グラフの構成

双方向有向グラフ $G_W = (N_W, A_W)$ を導入する．点集合に関しては G' と同じで，枝集合に関しては，G' の各枝を双方向にする．さらに，G_W の各枝 (i,j) に以下の重みを付す．

$$w_{ij} = \begin{cases} -\infty, & \text{if } j = 1, \\ p_j - c_{ij} & \text{otherwise.} \end{cases} \tag{1}$$

G_W は，$w_{ij} > 0$ かつ $w_{ji} > 0$ を満たす枝 (i,j) を持たない．なぜなら，もしそのような枝を持つならば，G' で $p_j > c_{ij}$ かつ $p_i > c_{ij}$ が成立つため，点 i と j が併合されるからである．式 (1) による重みを付した後，$w_{ij} > 0$ を満たす全ての枝に対して，$w_{ji} = -\infty$ と付し直す．

Lemma 1. グラフ G_W は正の重みの有向閉路を持たない．

3.3　最長路の計算

G_W の全域木を求めるために，グラフ G'_W を導入する．G'_W の点集合と枝集合は G_W と同じで，G'_W の各枝 (i,j) に対して，枝の重み $w'_{ij} = w_{ij}$ とする．G'_W で始点 i から終点 j までの最長路の長さを $v(i,j)$ と記す．

G'_W における最長路の計算：

$N_{W1} := \{1\}$, $N_{W0} := N_W \setminus \{1\}$ と初期設定する．

$N_{W0} \neq \emptyset$ が成立つ間，以下を繰り返す．

1 各点 $j \in N_{W0}$ に対して，集合 N_{W1} から点 j への最長路を求める．その最長路の長さを $v(N_{W1}, j) = \max_{i \in N_{W1}} v(i,j)$ とする．

2 $v(N_{W1}, j)$ の最大値をとる N_{W1} から j への経路の点集合を N_{WR} とすると，$N_{W1} = N_{W1} \cup N_{WR}$，$N_{W0} = N_{W0} \setminus N_{WR}$.

3 $i \in N_{W1}$, $j \in N_{W1}$ を満たす全ての枝 (i,j) に対して，$w'(i,j) = 0$ と更新する．

4 $i \in N_{W1}, j \in N_{W0}$ を満たす全ての枝 (i,j) に対して，$w'_{ji} = -\infty$ と更新する．

5 $i \in N_{W1}, j \in N_{W0}$ を満たす各枝 (i,j) に対して，もし $w'_{ij} = -\infty$ ならば，$w'_{ij} = p_j - c_{ij}$ と更新する．

Remark 1. *ループ 1* 巡目の *Step 1* では，*Lemma 1* より正の閉路は存在しない．*Step 3, 4, 5* で *Step 1* で求めた最長路に関連した枝の重みが更新されるが，枝の重みの付け方より，G'_W は正の閉路を持たない．

Remark 2. *Step 1* で求める最長路は，枝の重みの符号を逆にすることによって，最短路になる．符号を逆にしたグラフには負に閉路が存在しないので，最短路を多項式時間で求めることが出来る *[4]*.

Step 1 で計算される最長路をつなぐと，G'_W の全域木となり，この全域木を G_W の全域木と見なす．

3.4　枝の削除と最小木の計算

G_W の全域木から，利益を減少させる枝を削除する．つまり，その枝の重みと，(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すべて除く．枝の削除処理後，点 1 とつながっている点の集合を N_{WT} とし，

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

元のグラフ G に対して，点集合 N_T を張る最小木 $G_T = (N_T, A_T)$ を求める．

4　整数計画問題としての定式化

問題入力に対して，以下の有向グラフ $D = (N_D, A_D)$ を構成する．

- $N_D := N$.
- $A_D := \{(i,j) | \{i,j\} \in A, j \neq 1\}$.
- 各枝 $(i,j) \in A_D$ に重み $w_{ij} := p_j - c_{ij}$ を付す．

このグラフ上で，重み和が最大となるような根付き部分木を求めたい．そのために，次の 0-1 変数を導入する．

$$x_{ij} = \begin{cases} 1, & \text{根付き部分木が枝 } (i,j) \text{ を含むとき}, \\ 0, & \text{そうでないとき}. \end{cases}$$

整数計画問題としての定式化 (IP)

$$\text{max.} \quad \sum_{(i,j) \in A} w_{ij} x_{ij} \tag{2}$$

$$\text{s. t.} \quad \sum_{(i,j) \in A} x_{ij} \leq 1 \quad (j \in N \setminus \{1\}), \tag{3}$$

$$x_{ij} \leq \sum_{(k,i) \in A \setminus \{(j,i)\}} x_{ki} \quad ((i,j) \in A \setminus \mathrm{OUT}(1)), \tag{4}$$

$$\sum_{i \in S, j \in S} x_{ij} \leq |S| - 1 \quad (S \subseteq N_D,\, |S| > 1). \tag{5}$$

式 (3) は，各点の入次数が 1 以下であることに対応する．式 (4) は，根付き部分木の枝ならば，その枝の始点の入次数が 1 以上であることに対応する．文献 [5] では，式 (3) と (4) を，それぞれ incidence constraint, connectivity constraint と呼ぶ．式 (5) は閉路を除くための制約条件である．

参考文献

[1] C. Duhamel, L. Gouveia, P. Moura and M. Souza: “Models and heuristics for a minimum arborescence problem,” *Networks,* vol. 51, pp. 34–47, 2008.

[2] S. Engevall, M. Göthe-Lundgren and P. Värbrand: “A strong lower bound for the Node Weighted Steiner Tree Problem,” *Networks,* vol. 31, pp. 11–17, 1998.

[3] T. Kobayashi, T. Fukuma, H. Sakakibara and M. Tsuyuki: “Algorithm for the maximum profit rooted-tree problem (in Japanese),” The 2003 Fall National Conference of Operations Research Society of Japan, pp. 166–167, 2003.

[4] B. Korte and J. Vygen: “Combinatorial Optimization: Theory and Algorithms,” Second Edition, Chapter 7, Springer-Verlag, 2002.

[5] V.V. Rao and R. Sridharan: “Minimum-weight rooted not-necessarily-spanning arborescence problem,” *Networks,* vol. 39(2), pp. 77–87, 2002.