

Prize Collecting Steiner Tree問題に対する ヒューリスティクス

HOSOKAWA, Yuki / 細川, 祐樹

(出版者 / Publisher)

法政大学大学院理工学・工学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 理工学・工学研究科編 / 法政大学大学院紀要. 理工学・工学研究科編

(巻 / Volume)

57

(開始ページ / Start Page)

1

(終了ページ / End Page)

4

(発行年 / Year)

2016-03-24

(URL)

<https://doi.org/10.15002/00013376>

Prize Collecting Steiner Tree 問題に対する ヒューリスティクス

A HEURISTIC ALGORITHM FOR THE PRIZE COLLECTING STEINER TREE PROBLEM

細川 祐樹

Yuki HOSOKAWA

指導教員 五島 洋行

法政大学大学院理工学研究科システム工学専攻修士課程

The Prize Collecting Steiner Tree (PCST) problem is an important problem in the field of combinatorial optimization. In this research, we develop two heuristics (H1, H2) for PCST problems. The heuristics consists of two stages. In the first stage, a spanning tree is computed, which is based on these heuristics. In the second stage, we delete vertices to improve the objective function value. Through computational experimentations, we found that method H1 is faster than method H2. Given a set of real-world instances, we obtained a good approximation ratio in both methods H1 and H2.

Key Words : Prize Collecting Steiner Tree Problem, Heuristic, Benchmark Instance

1. はじめに

Prize Collecting Steiner Tree (PCST) 問題は、組合せ最適化の重要な問題の1つである。この問題は、電線ケーブルや水道管の設置などのインフラ整備の効率化、広告看板に関する収益の最大化に対して応用が可能である。PCST 問題は Steiner Tree 問題を一般化したものであり、Steiner Tree 問題は NP 困難であることが証明されているため、PCST 問題も NP 困難である。

PCST 問題は NP 困難であるが、整数計画問題として定式化し最適解を求められる [1]。しかし整数計画問題の制約条件は入力の点数に比例して増加し、最適解を実時間内で求められないことがある。そのため、整数計画緩和問題の定式化から近似解を求める手法が存在する [1]。ほかにも、PCST 問題に対しての近似アルゴリズムはいくつか知られている。たとえば、プライマルデュアル法に基づいた近似アルゴリズムである [2]。この近似アルゴリズムは近似比 2 を保証している。さらに、近似比 1.9672 を保証するアルゴリズムが提案されている [3]。この近似アルゴリズムは現在、近似比が最も小さいことが知られている。

本研究では PCST 問題に対して、2つのヒューリスティクス (H1, H2) を提案し、PCST 問題を短時間で近似的に解く手法を検討する。

各ヒューリスティクスは、2段階からなる。第1段階では、全域木を求める。H1 はグリーディアルゴリズムに従い、局所的に最適な選択を繰り返す。H2 は入力を有向グラフに変換し、多フェーズグリーディアルゴリズムに従い全域木を求める。第2段階では、全域木から目的関数を最小にする部分木を求める。H1, H2 ともに、入力から最適な部分木を導くアルゴリズムを用いる。

入力をランダムグラフとベンチマークに存在する5つの K, P, C, D, Cologne グループとし、計算実験を行う。ランダムグラフについて、ヒューリスティクスの性能を比較する

対象に、3-近似アルゴリズムを用いる [1]。計算実験から得られた近似比の精度や計算時間について考察する。

2. PCST 問題

無向グラフ $G = (V(G), E(G))$ 、各辺 e にコスト $c_e \geq 0$ 、各点 i にペナルティ $\pi_i \geq 0$ 、特別な点 $r \in V(G)$ が与える。点 r を含む木を T とするとき、以下の目的関数

$$\sum_{e \in E(T)} c_e + \sum_{i \in V(G) \setminus V(T)} \pi_i, \quad (1)$$

を最小にする木 T を求めたい [1]。

3. PCST 問題に対する最適解の求め方

PCST 問題の入力を整数計画問題として定式化する詳細を示す。入力の点と辺に対応するバイナリ変数 x_e ($e \in E(G)$)、 y_i ($i \in V(G)$) を導入する。

$$x_e = \begin{cases} 1 & e \in E(T), \\ 0 & \text{otherwise.} \end{cases} \quad y_i = \begin{cases} 1 & i \in V(T), \\ 0 & \text{otherwise.} \end{cases}$$

変数 x_e , y_i の導入によって、以下のように整数計画問題に定式化することができる。

$$\text{目的関数} \quad \sum_{e \in E(G)} c_e x_e + \sum_{i \in V(G)} \pi_i (1 - y_i), \rightarrow \min \quad (2)$$

$$\text{制約条件} \quad \sum_{e \in \delta(S)} x_e \geq y_i, \quad (S \subset V(G) \setminus \{r\}, S \neq \emptyset, i \in S) \quad (3)$$

$$y_r = 1, \quad (4)$$

$$y_i \in \{0, 1\}, \quad (i \in V(G) \setminus \{r\}) \quad (5)$$

$$x_e \in \{0, 1\}. \quad (e \in E(G)) \quad (6)$$

$\delta(S)$ は、 $V(G)$ を S でカットした辺の集合である。近似比を求める際、この定式化を用いて最適解を求める。

4. 比較対象となる近似アルゴリズム

ヒューリスティクスの性能を比較する対象として、以下の3-近似アルゴリズムを用いる。

3-近似アルゴリズム

1. 入力を整数計画の緩和問題に定式化し、その解を \mathbf{x}^* と \mathbf{y}^* とする。
2. $U \leftarrow \{i \in V(G) : y_i^* \geq \alpha\}$. ただし $\alpha = \frac{2}{3}$.
3. すべての $i \in U$ を含む、できるだけ最小コストの木 T を求める。

3-近似アルゴリズムは単純なラウンディングアルゴリズムを適用している。ラウンディングアルゴリズムとは近似アルゴリズムの基本設計技法の1つである。はじめに入力を整数計画の緩和問題に定式化する。その緩和問題の解を \mathbf{x}^* と \mathbf{y}^* とする。次に U を条件 $y_i^* \geq \alpha$ を満たす点 $i \in V(G)$ の集合とする。パラメータ α は適切な値に定められ、PCST問題の α 値は $\alpha = \frac{2}{3}$ である。次に、 U の要素をすべて含む木を構築する。その木は条件を満たす中で、最小コストの木を選ぶ。

5. ヒューリスティクス

2つのヒューリスティクス (H1, H2) を提案する。各ヒューリスティクスは、2段階からなる。第1段階では、それぞれ用いるアルゴリズムに従い全域木を求める。第2段階では、得られた全域木から最適な部分木を求める。以下に提案ヒューリスティクス (H1) を示す。

ヒューリスティクス (H1)

全域木の構築

1. 根を選択。
2. グリーディな辺の選択。
3. すべての点を含むまで、繰り返す。

点の削除

● 点の削除は最適な部分木を求めることと同意義であり、最適な部分木を求めるアルゴリズムを後に示す。そのアルゴリズムは入力が無向木の際にも適用できる。

ヒューリスティクス (H2) では、全域木を求める際、最小コスト有向木を求めるアルゴリズムを適用する。そのため、入力の無向グラフを有向グラフへ変換する必要がある。全域木を求めた後、H1と同じようにして最適な部分木を求める。以下に無向グラフを有向グラフに変換するアイデアと、最小コスト有向木を求めるアルゴリズムを示す。

(1) 有向グラフへの変換

無向グラフを有向グラフに変換するアイデアを提案する。このことは入力と与えられるペナルティをコストに反映させるという自然なアイデアから生まれている。

無向辺とそのコストを図1(a)に示す。点 i , 点 j にそれぞれ π_i, π_j が付随し、それらを結ぶ無向辺のコストを c_{ij} とする。その無向辺を2本の有向枝に変換したのが図1(b)である。点 i から点 j に向かう有向枝のコストを c_{ij}^* と仮定すると、 $c_{ij}^* := \pi_j - c_{ij}$ と定義する。

この動作を各辺について行うことによって、入力の無向グラフはペナルティを取り除いた有向グラフになる。

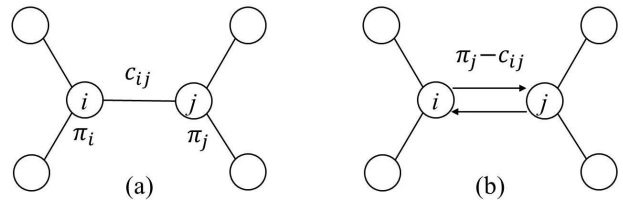


図1 一般的な無向辺の例 (a) と変換される2本の有向枝 (b)。

(2) 最小コスト有向木の選択

有向グラフから最小コストの有向木を求める問題を最小コスト有向木問題といい、最適な木を出力するアルゴリズムが存在する [4]。その解決法は多フェーズグリーディアルゴリズムと呼ばれ、再帰的、逐次的に有向木を求めるアルゴリズムである。無向グラフから最小コストの無向木を求める問題の解決法であるクラスカル法などでは最適な有向木を求めることができない。

6. 最適部分木問題

有向木 $T = (V(T), E(T))$ 、有向枝 $e \in E(T)$ のコスト w_e 、根 r が与えられる。点 r を含む部分木を T' とするとき、以下の目的関数

$$\sum_{e \in E(T')} w_e, \quad (7)$$

を最大にする木 T' を求めたい。

(1) 最適な部分木の探索

上記の問題において、最適な部分木 T' を求めるアルゴリズムを提案する。

アルゴリズム 6.1

1. 根を始点として深さ優先探索を行い、各点に後行順でラベル付け。
2. 後行順に、各点 v に対して、以下を実行。
 - 点 v の親を u とし、 u から v に向かう有向枝を e とする。
 - 各点に $f_{i \in V(T)}$ を導入し、初期値を0とする。
 - **If** $f_v - w_e > 0$,
 f_u を $f_u = f_u + f_v - w_e$ に更新。
 - **Else** 点 v を含めたすべての子孫を削除。

アルゴリズム 6.1 は入力有向木の際、最適な部分木を求められる。また無向木に関しても、有向木に変換することでアルゴリズム 6.1 が適用できる。したがって入力が無向木である H1 にも適用できる。アルゴリズム 6.1 の正当性の証明は割愛する。

(2) 時間計算量の解析

入力の点数を n 、枝数を m と仮定する。深さ優先探索による点のラベル付けは根を始点としているため、その計算時間は $O(n+m)$ である。点の削除について、一度判定し、削除された点は2度と削除されない。そのため各点における点の削除は全体で $O(n)$ である。よって全体で $O(n+m)$ とできる。ここで全域木の性質から $m = n - 1$ より、全体の時間計算量は $O(n)$ となる。したがってアルゴリズム 6.1 は多項式時間で実装できる。

表 1 点数 10 に固定したときの各近似比.

$ V $	$ E $	H1	H2	3-近似
10	15	1.055	1.045	1.030
10	20	1.107	1.065	1.024
10	25	1.158	1.081	1.010
10	30	1.203	1.123	1.012
10	35	1.246	1.115	1.010
10	40	1.277	1.156	1.012

7. 計算実験

新しく提案する 2 つのヒューリスティクスについて、ランダムグラフとベンチマークの入力を用いた性能検証を行う。実験環境は以下のとおりである。

Machine: lenovo ThinkPad X240,
CPU: Intel(R) Core(TM) i3-4010U 1.70GHz,
OS: Microsoft Windows 7 Enterprise,
Memory: 4GB,
Compiler: Microsoft Visual C++ 2008 Express Edition.

ベンチワークとはあらゆる分野に存在し、アルゴリズムの性能を確認するための入力である。PCSP 問題にもベンチマーク問題は存在し、本研究では K, P, C, D, Cologne グループについて計算実験を行う。ランダムグラフにおける最適解は前述した整数計画問題の定式化から導く。整数計画問題を解くために非商用ソフト SCIP ソルバーを用いた。

(1) 点数を固定したランダムグラフに対する結果

ランダムグラフを点数 10 に固定し、辺数をいくつか設定して 2 つのヒューリスティクス H1, H2 の性能比較を行った。性能比較の対象に 3-近似アルゴリズムを用いた。各サンプル数を 100 ずつ生成し、それらのグラフ構造はランダムに設定する。ただし、各サンプルの点はすべて r へのパスが存在する。各点のペナルティ、各辺のコストはどちらも [1, 100] の一様乱数にした。

点数が 10 における近似比の結果を表 1 に示す。近似比とは、各ヒューリスティクスで求めた評価値を最適値で割った商である。横軸はそれぞれ点数、辺数、H1 の近似比、H2 の近似比、3-近似アルゴリズムで求めた近似比である。各近似比は、100 の試行を繰り返して算出した近似比の平均である。例えば、点数、辺数がそれぞれ 10, 20 のとき、H2 を用いた近似比の平均値は 1.65 である。H1 と H2 について、辺数の増加とともに近似比は増加し、辺数に関わらず、H2 の近似比は H1 より良好な結果を示した。図 1 から、辺数が多いほど H2 の近似精度は H1 より優れているという傾向があった。一方、3-近似アルゴリズムは辺数の増加とともに、近似比が減少した。

(2) 大規模なランダムグラフに対する結果

大きな入力サイズに対して、2 つのヒューリスティクス H1, H2 の性能比較を行った。辺数を 1000 に固定したときの評価値の結果を表 2 に示す。横軸はそれぞれ点数、辺数、H1 の評価値、H2 の評価値である。H1 と H2 について、表 2 より点数の増加とともに評価値も増加した。評価値を比較すると、H2 の方が H1 より明らかに良好な結果を示した。

辺数を 1,000 に固定したときの計算時間を表 2 に示す。点数を 10 に固定したときと同様に、点数の増加に比例して H1 と H2 の計算時間は高速であることが確認された。

表 2 大規模なグラフにおける H1, H2 の評価値.

$ V $	$ E $	H1	H2
100	1,000	865	633
200	1,000	2,923	2,421
300	1,000	5,748	5,120
400	1,000	9,085	8,563
500	1,000	13,015	12,661

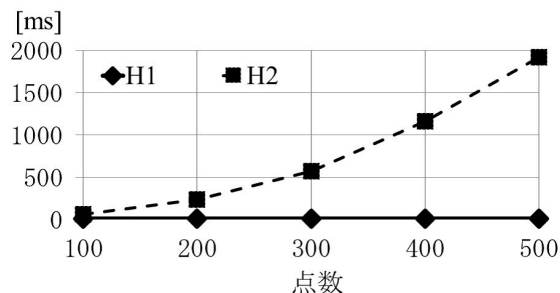


図 2 大規模なグラフにおける H1, H2 の計算時間 [msec.]

(3) KP グループに対する結果

Johmson et al. [6] が計算実験で用いたベンチマークがある。それらのベンチマークには 2 つのグループがあり、その 1 つが K と呼ばれる。K グループは地図上の構造に良く似ている。入力の点を交差点と仮定しているため、ペナルティ 0 である点を多く持っている。もう 1 つは P と呼ばれ、構造はランダムである。P の特徴は総点数に占めるペナルティ 0 でない点の割合がおおよそ 5 割であり、K と比べて 2 倍ほど多い。サンプル数は K が 23, P が 11 であり、最大の点数は 400, 辺数は 1,576 であった。これらのベンチマークは Lucena[7] や Canuto et al. [8] もヒューリスティクスの性能比較に用いている。

K, P グループにおける近似比の結果を表 3 に示す。横軸はサンプル名、点数、辺数、H1 の近似比、H2 の近似比である。点数、辺数、近似比はともに K, P の試行を繰り返して算出した平均値である。近似比を比較すると、H1 の方が H2 より良好な結果を示した。特に K に対する近似比について、H1 は H2 よりおよそ 1.4 小さい結果を得た。これは総点数に占めるペナルティ 0 である点の割合が、K が P よりも約 2 倍多いためであると推測する。

(4) CD グループに対する結果

Canuto et al. [8] は OR-Library に存在する PCST の入力例を用いて、新たに 80 の入力を生じた。それらは C と D グループに分類されていて、サンプル数は各 40 である。C の点数は 500, D の点数は 1,000 であり、ほとんどの点がペナルティ 0 である。C の辺数は 625 から 12,500 の間、D の辺数は 1,250 から 25,000 の間である。C, D について、コ

表 3 KP グループにおける H1, H2 の近似比.

サンプル	$ V $	$ E $	H1	H2
K	248	920	1.129	1.133
P	245	745	1.266	1.351

表 4 C グループにおける H1, H2 の近似比.

サンプル	$ V $	$ E $	H1	H2
C-A	500	4,156	1.366	1.505
C-B	500	4,156	1.829	1.850
C	500	4,156	1.598	1.677

表 5 D グループにおける H1, H2 の近似比.

サンプル	$ V $	$ E $	H1	H2
D-A	1,000	8,313	1.368	1.452
D-B	1,000	8,313	1.893	1.846
D	1,000	8,313	1.630	1.699

ストは $[1, 10]$ の一様乱数である. C, D は, さらに A と B に分類され, 0 でない点のペナルティは A が $[1, 10]$, B が $[1, 100]$ の一様乱数に設定されている.

C, D グループにおける近似比の結果を表 4, 表 5 に示す. 横軸はサンプル名, 点数, 辺数, H1 の近似比, H2 の近似比である. 点数, 辺数, 近似比はともに, サンプルの試行から算出した平均値である. 近似比を比較すると, C, D どちらのグループも H1 の方が H2 より良好な結果を示した. 一方, B に分類された入力に関して, 近似比はどちらも 1.8 を超えた.

(5) 光ファイバーの設置計画に基づいた入力に対する結果

P. Bachhiesl et al. [9] が作成した現実問題に基づいたベンチマークがある. それらのベンチマークも 2 つに分類され, Cologne1, Cologne2 と呼ばれる. サンプルはドイツのケルンにおける光ファイバーの設置計画に基づいて, その総数は合わせて 29 存在する. それらは GIS のデータに基づいているため, 点数と辺数が等しい入力が多い. しかし顧客の選択や価値, 経由地点の変化を考慮しているため, 各サンプルに違いがある. Cologne1, Cologne2 はさらに 5 つのグループに分類される.

Cologne1, Cologne2 グループにおける近似比の結果を表 6, 表 7 に示す. 横軸は点数, 辺数, H1 の近似比, H2 の近似比である. 点数, 辺数, 近似比はともに, 各サンプルの試行から算出した平均値である. 近似比を比較すると, Cologne1, Cologne2 ともに H1 が H2 と比べて良好な結果を示した. 各サンプルにおいて, 点数, 枝数の違いが少ないためか, H1, H2 の近似精度に大きな違いはみられなかった. Cologne1 の i04 に分類されるサンプルは H1 と H2 の近似比が等しくなる結果も得た.

表 6 Cologne1 グループにおける H1, H2 の近似比.

サンプル	$ V $	$ E $	H1	H2
i01	768	6,332	1.079	1.148
i02	769	6,342	1.048	1.088
i03	771	6,342	1.050	1.061
i04	761	6,293	1.317	1.317
i05	716	6,296	1.166	1.193
Cologne1	766	6,321	1.119	1.150

表 7 Cologne2 グループにおける H1, H2 の近似比.

サンプル	$ V $	$ E $	H1	H2
i01	1,819	16,743	1.165	1.239
i02	1,820	16,740	1.454	1.516
i03	1,825	16,762	1.289	1.364
i04	1,817	16,719	1.019	1.031
i05	1,826	16,794	1.140	1.182
Cologne2	1,821	16,752	1.213	1.266

8. おわりに

本研究は, PCST 問題に対する 2 つのヒューリスティクスを提案し, 近似精度と計算時間の観点から計算実験を行った. 大規模なランダムグラフにおいて, H1 は高速に動くことを確認した. 高速道路に近い構造をした K や光ファイバーの設置計画に基づいた Cologne グループの入力において, H1, H2 ともに良好な近似解を得た. これは交差点や経由地点を多くもつ入力に対して, 効果的といえる. したがって H1 は現実問題に近い入力において, 高速かつ精度の良い近似解を出力できると推察する.

PCST 問題に対する既存解法に, 主双対法に基づいたアルゴリズムも存在する [2]. 今後の課題として, 計算実験からこのアルゴリズムと 2 つの提案ヒューリスティクスを比較することが挙げられる.

参考文献

- 1) D.P. Williamson and D.B. Shmoys, The Design of Approximation Algorithms, Cambridge University Press, 2011.
- 2) 浅野孝夫, 近似アルゴリズム, 丸善出版, pp.212–214, 2014.
- 3) A. Archer, M. Bateni, M. Hajiaghayi, and H. Karloff, “Improve approximation algorithms for prize-collecting Steiner tree and TSP”, SIAM Journal on Computing, vol.40, no.2, pp.309–332, 2011.
- 4) 浅野孝夫, 浅野泰仁, 小野孝男, 平田富男, アルゴリズムデザイン, 共立出版, 2008.
- 5) M.X. Goemans, and D.P. Williamson, “A general approximation technique for constrained forest problems”, SIAM Journal on Computing on Computing, vol.24, no.2, pp.296–317, 1995.
- 6) D.S. Johnson, M. Minkoff, and S. Phillips, “The prize-collecting Steiner problem: Theory and practice”, In Proceedings of 11th ACM-SIAM Symposium on Discrete Algorithms, pp.760–769, 2000.
- 7) A. Lucena, and M.G.C. Resende, “Strong lower bounds for the prize-collecting Steiner problem in graphs”, Discrete Applied Mathematics, vol.141, pp.277–294, 2004.
- 8) S.A. Canuto, M.G.C. Resende, and C.C. Ribeiro, “Local search with perturbations for the prize-collecting Steiner tree problem in graphs”, Networks, vol.38, pp.50–58, 2001.
- 9) P. Bachhiesl, M. Prosegger, G. Pautius, J. Paulus, J. Werner, and H. Stoger, “Simulation and optimization of the implementation costs for the last mile of fiber optic networks”, Network and Spatial Economics, vol.3, no.4, pp.467–487, 2003.