直径の小さなグラフ上の全域木混雑度問題 (計算理論とアルゴリズムの新潮流)

AUTHOR(S):

久保, 浩平; 山内, 由紀子; 来嶋, 秀治; 山下, 雅史

CITATION:

久保, 浩平 ...[et al]. 直径の小さなグラフ上の全域木混雑度問題 (計算理論とアルゴリズムの新潮流). 数理解析研究所講究録 2015, 1941: 17-21: KJ00009803681.

ISSUE DATE:

2015-04

URL:

http://hdl.handle.net/2433/223807

数理解析研究所講究録
第1941巻 2015年 17-21
2014年度冬のLAシンポジウム [C-1]

直径の小さなグラフ上の全域木混雑度問題

久保 浩平 Kohei Kubo　山内 由紀子 Yukiko Yamauchi　来嶋 秀治 Shuji Kijima　山下 雅史 Masafumi Yamashita

九州大学
Kyushu University

1 はじめに

全域木混雑度とは，Ostrovskii[5] が定義したグラフパラメータである．無向グラフ H は連結とし，H の全域木を T とする．各 $\{u,v\} \in E(H)$ に対し，T 上の u–v パスを迂回路と呼ぶ．$e \in E(T)$ の**混雑度**を e を含む迂回路の本数とし，$\mathrm{cng}_T(e)$ で表す．T の H での混雑度を T の全ての辺の混雑度の最大値と定義し，$\mathrm{cng}_H(T)$ で表す．すなわち，$\mathrm{cng}_H(T) = \max_{e\in E(T)} \mathrm{cng}_T(e)$ とする．H の**全域木混雑度**を H の全ての全域木のうち，最小の混雑度で定義し，$\mathrm{stc}(H)$ で表す．すなわち，

$$\mathrm{stc}(H) = \min\{\mathrm{cng}_H(T) \mid T \text{ は } H \text{ の全域木}\}$$

とする．全域木混雑度問題とは，グラフ H と正整数 k が与えられたとき，$\mathrm{stc}(H)$ が k 以下かどうかを判定する問題である．また，$\mathrm{cng}_G(T^*) = \mathrm{stc}(G)$ を満たす全域木 T^* を G の**最小混雑全域木 (minimal congestion spanning tree)** と言う．

$v \in V(H)$ の H での隣接頂点の集合を $N_H(v)$ で表す．v の H での次数を $d_H(v)$ で表す．頂点の部分集合 $S \subseteq V(H)$ で誘導される H の部分グラフを $H[S]$ と表す．$e \in H$ に対して，$H-e$ を H から e を除いて得られるグラフとする．$A, B \subseteq V(H)$ に対して，$E(A,B) = \{\{u,v\} \in E(H) \mid u \in A, v \in B\}$ と定義する．$S \subseteq V(H)$ に対して，$\theta_H(S) = E(S, V(H)\setminus S)$ と定義する．T を連結な木とする．$e \in E(T)$ に対して，$T-e$ の2つの連結成分の1つを $A_T(e)$ とする．$B_T(e) = V(T) \setminus A_T(e)$ とする．$\mathrm{cng}_T(e)$ に対して成り立つ事実を以下に観察としてまとめる．

観察 1.1. *([4])* H を連結無向グラフとし，T を H の全域木とする．以下が成り立つ．

$$\begin{aligned}\mathrm{cng}_T(e) &= |\theta_H(A_T(e))| \\ &= \sum_{v\in A_T(e)} d_G(v) - 2|E(H[A_T(e)])|.\end{aligned}$$

全域木混雑度問題については様々な結果が報告されている [1, 2, 4]．特に，岡本ら [4] は split グラフや chain グラフなどの比較的単純な構造のグラフに対してさえ全域木混雑度問題が NP 完全であることを示した．split グラフの NP 完全性から，グラフの直径が3以下に制限される密なグラフに限定しても，全域木混雑度問題は NP 完全であることがわかる．

本研究では直径が3以下のグラフ上の全域木混雑度問題を扱う．2章で任意のグラフに対する全域木混雑度の下界と，本研究で扱うグラフクラスを導入する．3章では split グラフ上の近似アルゴリズムを設計する．4章で co-chain グラフ上の多項式時間アルゴリズムを述べる．

2 準備

2.1 最小混雑木

本節では全域木混雑度問題の実行可能領域を緩和した問題である最小混雑木を導入する．最小混雑木は3章の近似アルゴリズムの設計の際に，全域木混雑度の下界を達成する木として用いる．

H の **木混雑度 (tree congestion)** を

$$\mathrm{tc}(H) = \min\{\mathrm{cng}_H(T) | T \text{ は } V(H) \text{ 上の木 }\}$$

と定義する．$\mathrm{cng}_H(T^*) = \mathrm{tc}(H)$ を満たす木 T^* を H の**最小混雑木 (minimal congestion tree)** という．最小混雑木の各枝は G の辺とは限らないことに注意されたい．$a, b \in V(H)$ に対し，$\mu_H(a,b)$ を H での辺素 $a-b$ パスの最大本数とし，$\mu_H = \max_{a.b \in V(H)} \mu_H(a,b)$ とする．Ostrovskii[5] の結果の一部を次の定理にまとめる．

定理 2.1. ([5]) 任意のグラフ H に対して，

1. $\mathrm{stc}(H) \geq \mu_H = \mathrm{tc}(H)$.
2. 次数が μ_H 以下の頂点が葉であるような H の最小混雑木が存在する．

2.2 グラフクラス

本研究で扱うグラフクラスを導入する．グラフ G の頂点集合 $V(G)$ がクリーク C と独立集合 I に分割できるとき，G は split グラフであるという．すなわち $V(G) = C \cup I$ である．

chain グラフとは，2 部グラフ $H = (P, Q; E)$ で，かつ P 上に隣接頂点集合に関する線形順序 $<_P$ が定義できるグラフである．ここでの線形順序 $<_P$ とは，任意の $u, v \in P$ に対して，$u <_P v$ ならば $N_H(u) \subseteq N_H(v)$ が成り立つものと定義する．co–chain グラフとは chain グラフの補グラフ $\bar{G} = (P, Q; \bar{E})$ である．

split グラフ，co–chain グラフともにグラフの直径は高々3 である．

3 split グラフ上の近似アルゴリズム

本章では split グラフ上の近似アルゴリズムについて述べる．L_T を T の葉の集合とする．$\alpha(T) = 2\frac{|V(T)|-1}{|L_T|}$ とする．次の定理が成り立つ．

定理 3.1. *split* グラフ上の全域木混雑度問題に対する $\alpha(T[C]) + 2$ 近似アルゴリズムが存在する．

本稿では定理 3.1 の証明は省略する．Algorithm1 に本章のアルゴリズムを与える．アルゴリズムの概要は以下の通りである．まず split グラフ上の最小混雑木を求める．次に最小混雑木から全域木を求める問題を整数計画問題として定式化する．これを LP 緩和し，反復丸め法 [3] を用いて近似解を得る．

まず split グラフ上の最小混雑木について述べる．以下の補題が成り立つ．

補題 3.2. G を *split* グラフとする．任意の独立集合の頂点が葉であるような G の最小混雑木が存在する．

証明. 任意の $u \in I$ に対して，$d_G(u) \leq |C|$ である．また，次数が 2 以上である独立集合の頂点が存在するとき，$\mu_G \geq |C|$ が成り立つ．$d_G(u) \leq \mu_G$ なので，定理 2.1 の 2 より，任意の独立集合の頂点が葉であるような G の最小混雑全域木が存在する． □

次に最小混雑木から全域木を求める問題を定式化する．補題 3.2 を満たす最小混雑木を T とする．このとき，明らかに $T[C]$ は連結である．T から全域木を得るために独立集合の各頂点を G の辺で接続しなおす必要がある．この問題を整数計画問題へ定式化する．定式化のために記号を導入する．$F = E(C, I)$ とする．split グラフ上の全域木 T と $e \in E(T)$ に対して，$C_{A_T(e)} = A_T(e) \cap C, C_{B_T(e)} = B_T(e) \cap C, I_{A_T(e)} = A_T(e) \cap I, I_{B_T(e)} = B_T(e) \cap I$ とする．$\Gamma(i, e; T)$ を

$$\Gamma(i,e;T) = \begin{cases} N_G(i) \cap C_{A_T(e)} & i \in I_{B_T(e)}, \\ N_G(i) \cap C_{B_T(e)} & i \in I_{A_T(e)}. \end{cases}$$

と定義する．$e \in T[C]$ に対して，$B_e = \sum_{u \in I} |\Gamma(u, e; T)|$ とする．変数 $f \in F$ に対して，x_f を

$$x_f = \begin{cases} 1 & f \in T' \\ 0 & \text{otherwise} \end{cases}$$

と定義すると，定式化は以下のようになる．

(IP)

subject to.

$$\sum_{v\in N_G(u)} x_{\{u,v\}} = 1 \qquad \forall u \in I, \qquad (1)$$

$$\sum_{u\in I_{B_T(e)}} \sum_{v\in C_{A_T(e)}} (d_G(u) - 2|\Gamma(u,e;T)|)x_{\{u,v\}} + \sum_{u\in I_{A_T(e)}} \sum_{v\in C_{B_T(e)}} (d_G(u) - 2|\Gamma(u,e;T)|)x_{\{u,v\}} \leq B_e \qquad \forall e \in T(C), \qquad (2)$$

$$x_f \in \{0,1\} \qquad \forall f \in F. \qquad (3)$$

便宜のため，$I' := I, C' := C\backslash\{v_1\}, F' := \{\{u,v\} \in F \mid u \in I' \text{ and } v \in C'\}$ として，(IP) を LP 緩和すると，以下の線形計画問題を得る．

(LP)$[I', C'; F'; B']$

subject to.

$$\sum_{v\in N_G(u)} x_{\{u,v\}} = 1 \qquad \forall u \in I, \qquad (4)$$

$$\sum_{u\in I_{B_T(e)}} \sum_{v\in C_{A_T(e)}} (d_G(u) - 2|\Gamma(u,e;T)|)x_{\{u,v\}} + \sum_{u\in I_{A_T(e)}} \sum_{v\in C_{B_T(e)}} (d_G(u) - 2|\Gamma(u,e;T)|)x_{\{u,v\}} \leq B_e \qquad \forall e \in T(C), \qquad (5)$$

$$x_f \geq 0 \qquad \forall f \in F. \qquad (6)$$

補題 3.3. (LP)$[I, C; F; B]$ は実行可能解を持つ．

証明. 任意の $u \in I$ と任意の $v \in N_G(u)$ に対して，$x_{\{u,v\}} = \frac{1}{d_G(u)}$ と置く． □

$v \in C$ に対して $d_F(v) = |N_G(v) \cap I|$ とする．以下の補題が成り立つ．

補題 3.4. x を全ての要素が小数値であるような (LP) の端点解とする．このとき，$d_{F'}(v) \leq \alpha(T[C'])$ を満たす頂点 $v \in L_{T[C']}$ が存在する．

補題 3.4 から，Algorithm1 は多項式時間で停止する．

Algorithm 1 反復丸め

1: $E(T') := E(T) \cap E(G)$, $e = \{v_i, v_j\} \in E(T[C])$ とし，$v_i \in A_T(e)$ とする．
2: LP:=(LP) とする．表記の便宜のため，$I' := I$, $F' := F$, $C' := C$ とする．
3: **while** $I' \neq \emptyset$ **do**
4: LP$[\, I', C'; F'; B'\,]$ を解き，その端点解を x とする．
解 x に従って，LP を以下の要領で更新する．
5: $E(T') := E(T') \cup \{\{u,v\} \in F' \mid x_{\{u,v\}} = 1, u \in I', v \in C'\}$,
6: **for** $\{u,v\} \in F'$ s.t. $x_{\{u,v\}} = 1$ **do**
7: $I' := I' \setminus \{u\}$
8: 任意の $e \in E(T[C])$ に対して，
$u \in A_T(e), v \in B_T(e)$ または $u \in B_T(e), v \in A_T(e)$ ならば，
$B'_e := B'_e - (d_G(u) - 2\Gamma(e,u,T))$ と更新する．
9: **end for**
10: $F' := \{f \in F' \mid 0 < x_f < 1\}$ と更新する．
11: 便宜のため，
$C'' := \{v \in C' \mid$ 補題 3.4 の条件を満たす $\}$,
$I'' := \bigcup_{v\in C''} N_{F'}(v)$ と表す．
12: **for** $v \in C''$ **do**
13: $E(T') := E(T') \cup \{\{u,v\} \in F' \mid u \in N_{F'}(v)\}, C' := C' \setminus \{v\}$,
$I' := I' \setminus N_{F'}(v), F' := F' \setminus \{\{u,v\} \in F' \mid u \in N_{F'}(v)\}$ と更新する．
14: **end for**
15: **end while**
16: **return** T'

4 co-chainグラフ上の多項式時間アルゴリズム

本章ではco-chainグラフ上の多項式時間アルゴリズムについて述べる．まずco-chainグラフ上の最小混雑全域木に対して成り立つ補題を述べる．次にアルゴリズムを述べ，最後にアルゴリズムの正当性の証明を行う．

4.1 co-chainグラフ上の最小混雑全域木の性質

以下の補題を述べる．$p^{\max} = \mathrm{argmax}\{d_G(p) \mid p \in P\}, q^{\max} = \mathrm{argmax}\{d_G(q) \mid q \in Q\}$ とする．

補題 4.1. $G = (P; Q, E)$ を *co-chain* グラフとする．G の最小混雑全域木のなかで，以下の条件のうち少なくとも一方を満たすような木 T^* が存在する．

1. $T^*[P]$ が $p^{\max}$ を除いて葉で，かつ任意の $p \in P \setminus \{p^{\max}\}$ に対して $\{p, p^{\max}\} \in E(T^*)$.
2. $T^*[Q]$ が $q^{\max}$ を除いて葉で，かつ任意の $q \in Q \setminus \{q^{\max}\}$ に対して $\{q, q^{\max}\} \in E(T^*)$.

証明の方針．まず次の補題を示す．

補題 4.2. $G = (P; Q, E)$ をco–bipartiteグラフとする．T を G の最小混雑全域木とする．このとき，T から $T^*[P]$ が連結，または $T^*[Q]$ が連結であるような G の最小混雑全域木 T^* に変形できる．

次にco-chainグラフでは補題4.2から補題4.1を満たす全域木へ変形できることを示す． □

以降の議論では最小混雑全域木は補題4.1の1を満たすと仮定する．

4.2 アルゴリズムと正当性

本節ではco-chainグラフ上のアルゴリズムについて述べる．アルゴリズムの方針は最小混雑全域木の候補となる全域木を全て列挙し，最も混雑度の低い木を出力する，というものである．$Q^P = \{q \in Q \mid N_G(q) \cap P \neq \emptyset\}$, $|Q^P| = l$ とし，Q^P の各頂点 q^P に次数の降順で添字をつける．すなわち $d_G(q_1^P) \geq d_G(q_2^P) \geq \cdots \geq d_G(q_l^P)$ とする．具体的なアルゴリズムをAlgorithm2で述べている．Algorithm2が最小混雑全域木を出力することを証明する．すなわち，以下の定理を証明する．

Algorithm 2 co-chain(G)

1: $E(T_1) \leftarrow \bigcup_{p \in P \setminus p^{\max}} \{\{p^{\max}, p\}\} \cup \bigcup_{q \in Q \setminus \{q_1^P\}} \{\{q_1^P, q\}\} \cup \{\{q_1^P, p^{\max}\}\}$.
2: $Q_1^{T_1} \leftarrow Q$.
3: **for** $j = 2, \ldots, l$ **do**
4: $\quad Q_j^{T_j} \leftarrow Q_{j-1}^{T_{j-1}} \setminus \{q_{j-1}^P\}$
5: $\quad E(T_j) \leftarrow E(T_{j-1}[P]) \cup \{\{q_j^P, p^{\max}\}, \{q_{j-1}^P, p^{\max}\}\} \cup \bigcup_{q \in Q_j^{T_j} \setminus \{q_j^P\}} \{\{q_j^P, q\}\}$
6: **end for**
7: $E(T_{l+1}) \leftarrow \bigcup_{r \in N_G(p^{\max})} \{\{p^{\max}, r\}\}$
8: 任意の $q_i^P \in Q^P$ に対して，$Q_i^{T_{l+1}} \leftarrow \{q_i^P\}$ とする．
9: **for** $\forall q \in Q \setminus Q^P$ **do**
10: $\quad q_{i^*}^P = \mathrm{argmin}_{q_i^P \in Q^P} \max_{q_i^* \in Q^P} |\theta_G(Q_i^{T_{l+1}} \cup \{q\})|$
11: $\quad E(T_{l+1}) \leftarrow E(T_{l+1}) \cup \{\{q_{i^*}^P, q\}\}$
12: $\quad Q_{i^*}^{T_{l+1}} \leftarrow Q_{i^*}^{T_{l+1}} \cup \{q\}$
13: **end for**
14: $T_{\min} = \mathrm{argmin}_{T_i} \mathrm{cng}_G(T_i)$
15: **return** $T_{\min}$

定理 4.3. *Algorithm2* は *co-chain* グラフ上の全域木混雑度問題に対する多項式時間アルゴリズムである．

Algorithm2が多項式時間で停止することは明らかである．よってAlgorithm2の出力 $T_{\min}$ が最小混雑全域木であることを証明する．co-chainグラフ上の全域木 T に対して，h を $T[Q]$ の連結成分の数とし，$Q_1^T, Q_2^T, \ldots, Q_h^T$ を $T[Q]$ の各連結成分とする．定理4.3の証明のためにco-chainグラフ上の最小混雑全域木に対するいくつかの補題を導入する．

補題 4.4. 補題 *4.1* の条件を満たす最小混雑全域木の中で，$|Q_i^{T^*} \cap Q^P| \geq 2$ を満たす $Q_i^{T^*}$ が高々 *1* つしか存在せず，さらにそのような $Q_i^{T^*}$ に対して，$|Q_i^{T^*}| > \frac{|Q|}{2}$ であるような木 T^* が存在する．

補題 4.5. T^* を補題 *4.1,4.4* の条件を満たす G の最小混雑全域木とする．$Q_1^{T^*}$ は $|Q_1^{T^*} \cap Q^P| \geq 2$ を満たす Q の部分集合とする．このとき $Q \setminus Q^P \subseteq Q_1^{T^*}$ を満たす G の最小混雑全域木 T^* が存在する．

補題 4.6. 補題 *4.14.4,4.5* を満たす最小混雑木の中で，任意の $q \in Q_1^{T^*} \cap Q^P$ と任意の $q' \in Q^P \setminus Q_1^{T^*}$ に対して $d_G(q) \leq d_G(q')$ を満たす G の最小混雑全域木 T^* が存在する．

補題 4.7. $T^* = \mathrm{argmin}\{\mathrm{cng}_G(T_l), \mathrm{cng}_G(T_{l-1})\}$ とする．T^* は [$p^{\max}$ から出ている辺を全て枝に持つ] という条件のもとで，最適な木である．

定理 *4.3* の証明．T^* を補題 4.1 の条件を満たす最小混雑全域木とする．条件 1 を満たすと仮定する．

$|Q_i^{T^*} \cap Q^P| \geq 2$ を満たす $Q_i^{T^*}$ が存在するとき，補題 4.4 より，そのような $Q_i^{T^*}$ はただ 1 つのみである．これを $Q_1^{T^*}$ とする．補題 4.5,4.6 より，任意の $q' \in Q^P \setminus Q_1^{T^*}$ は T^* において葉であり，かつ任意の $q \in Q_1^{T^*} \cap Q^P$ に対して $d_G(q) \leq d_G(q')$ が成り立つ．この条件を満たす全域木はアルゴリズムの 2〜4 行目で列挙している．

$|Q_i^{T^*} \cap Q^P| \geq 2$ を満たす $Q_i^{T^*}$ が存在しないとき，補題 4.7 より，アルゴリズムの 5 行目以降で最適解を出力している．

以上から $T_{\min}$ は co-chain グラフに対する最小混雑全域木である． □

5 おわりに

本稿では直径の小さなグラフ上の全域木混雑度問題を扱った．まず，split グラフに対して反復丸め法を用いた近似アルゴリズムを設計した．次に co-chain グラフに対して多項式時間の厳密アルゴリズムを設計した．

今後の課題は co-bipartite グラフや，直径 2 のグラフ上の全域木混雑度問題について計算複雑性の解明や，近似アルゴリズムの設計を行うことである．

参考文献

[1] H.L. Bodlaender, K. Kozawa, T. Matsushima, Y. Otachi, Spanning tree congestion of k–outerplanar graphs, *Discrete Mathematics*, 311 (2011), pp. 1040–1045.

[2] K. Kozawa, Y. Otachi, K. Yamazaki On spanning tree congestion of graphs, *Discrete Mathematics*, 309 (2009), pp. 4215–4224.

[3] L.C. Lau, R. Ravi, M. Singh, *Iterative Methods in Combinatorial Optimization*, Cambridge University Press (2011).

[4] Y. Okamoto, Y. Otachi, R. Uehara, T. Uno, Hardness results and an exact exponential algorithm for the spanning tree congestion problem, *Lecture Notes in Computer Science*, 6648 (2011), pp. 452–462.

[5] M.I. Ostrovskii, Minimal congestion trees, *Discrete Mathematics*, 285 (2004), pp. 219–226.