数理解析研究所講究録
第 1799 巻 2012 年 137-144

2011 年度冬の LA シンポジウム [27]

Approximating Steiner Tree and Tree Cover problems in Directed Graphs

日比 智也*　　藤戸敏弘†

概 要

任意の Steiner tree T は，根と葉以外にターミナルを含まない極大部分木に一意に分解できるが，それらのいずれもが高さ L 以下であるとき，T を L 制限 Steiner tree という．本稿では，有向グラフ上の Steiner tree (DST) 問題に対するグリーディ近似解法を新たに提案する．同解法は，最小 L 制限 Steiner tree のコストを OPT_L として，コストが $O(L\log n)\cdot\mathrm{OPT}_L$ 以下の DST を出力し，L が定数であれば多項式時間であることを示す．更に，同解法を用いることで，有向グラフ上の Tree cover (DTC) 問題が $O(\log n)$ 倍近似可能であることを示す．

1 INTRODUCTION

STEINER TREE 問題は，入力として無向グラフ $G=(V,E)$，辺コスト関数 $c:E\rightarrow\mathbb{R}_+$，根 $r\in V$，terminal の集合 $X\subseteq V$ が与えられたとき，全ての terminal を含む最小コストの木 T を計算する問題である．STEINER TREE 問題は SAT に続いて Karp が示した NP 完全問題の一つであり [10]，実際には $\frac{96}{95}$ 倍近似困難であることも知られている [5]．その近似可能性についてはこれまで盛んに研究され続けており，現時点の最良近似は Byrka らによる 1.39 倍近似アルゴリズムである [2]．

TREE COVER 問題は無向グラフ $G=(V,E)$ とコスト関数 $c:E\rightarrow\mathbb{R}_+$ が与えられた時，E の各辺から端点を少なくとも一つ含む最小コストの木を計算する問題で，Arkin, Halldórsson, Hassin に導入された [1]．TREE COVER は辺コストが一定の場合，連結頂点被覆問題と等価であり，同問題は頂点被覆と同等以上に計算困難であることから [8]，TREE COVER 問題も NP 困難であるが，2 倍近似可能であることが知られている [16, 7]．

一般に，無向グラフ上の問題を有向グラフ上で考えると，より難しくなる傾向がある．有向グラフ上での STEINER TREE 問題は，有向グラフ $G=(V,A)$，辺コスト関数 $c:A\rightarrow\mathbb{R}_+$，根 $r\in V$，terminal の集合 $X\subseteq V$ が与えられたとき，r から発し各 terminal へ至るパスが存在するような最小コストの有向木 T を求める問題である (本稿では，有向木を各辺が根から葉の方向へ向かう木として定義する)．有向グラフ上での TREE COVER 問題は，有向グラフ $G=(V,A)$，辺コスト関数 $c:A\rightarrow\mathbb{R}_+$，および根 $r\in V$ が与えられたとき，G 内の各辺について，その head か tail，もしくは両方を含む有向木の中で最小コストなものを求める問題である．有向グラフ上での STEINER TREE 問題と TREE COVER 問題をそれぞれ DIRECTED STEINER TREE (DST) 問題と DIRECTED TREE COVER (DTC) 問題と呼ぶ．この二つの一般化された問題にはともに SET COVER 問題からの簡単な還元が存在し，$\Omega(\log n)$ 倍近似困難であり，$P=NP$ でなければ $O(\log n)$ 倍よりも良い近似はできない [6]．更にこの還元から，根から non-terminal へ向かう辺，および non-terminal から terminal へ向かう辺しかもたない二部グラフにおいても DST 問題は $\Omega(\log n)$ 倍近似困難であり，DTC 問題も同様の二部グラフで同じ近似困難性を有することがわかる．

DST 問題に対し，Charikar らは $i\geq 2$ について $O(n^i k^{2i})$ 時間で $i^2(i-1)k^{1/i}$ 倍近似アルゴリズムを貪欲的なアプローチによって与えた (ただし，$k=|X|$)．これより，任意の定数 $\epsilon>0$ につ

*豊橋技術科学大学 大学院工学研究科情報・知能工学専攻
†第 1 著者に同じ

いて，$O(|X|^{\epsilon})$ 倍多項式時間近似が得られ，また $i = \log k$ とすることで，$n^{O(\log|X|)}$ 擬多項式時間で $O(\log^3|X|)$ 倍近似可能であることが示された [17, 3, 4]．この結果により，DST 問題に対する多項式時間 polylog 倍近似アルゴリズムの存在が予想され，Konjevod が 2005 年に，Rothvoßが 2011 年に異なるアプローチを試みたが [11, 15]，いずれも Charikar らの近似保証を改善するに至らず，現在も DST 問題に対する最良の近似保証は $O(|X|^{\epsilon})$ である．

DST 問題の重要性は，この問題が他の多くの問題を一般化したものである点である．たとえば，SET COVER 問題，DTC 問題，GROUP STEINER TREE 問題などがあげられる．3 つ目の GROUP STEINER TREE 問題について，Halperin と Krauthgamer は木状のグラフに対する GROUP STEINER TREE 問題に対してさえ $\Omega(\log^{2-\epsilon} n)$ 倍近似困難であり，それによって GROUP STEINER TREE 問題の一般化である DST 問題もまた同等の近似困難性を持つことを示した [9]．

入力グラフに制限を加えた DST 問題に関しては，Carikar らのアルゴリズムによって入力グラフの最長パスが定数 L に制限されたグラフにおいて $O(\log n)$ 倍近似を得ることができる [4]．また Zosin と Khuller は，terminal を除いた頂点集合 $V \setminus X$ が深さ L の木であるグラフの場合に対し，$(L+1)\cdot O(\log n)$ 倍近似アルゴリズムを示した [18]．

本論文では，L 制限 Steiner tree という限定された DST を考え，最小 L 制限 Steiner tree コストの $2\cdot(L-1)\cdot H(|X|)$ 倍以下のコストを持つ DST を出力し，任意の定数 L で多項式時間な近似アルゴリズムを示す．ただし，$H(k) = \frac{1}{1}+\frac{1}{2}+\cdots\frac{1}{k} \leq 1+\ln k$ である．また，この結果から $V \setminus X$ が誘導するグラフの最長パスが定数 L に制限された入力グラフにおける DST 問題に対し，$2\cdot(L+1)\cdot H(|X|)$ 倍近似アルゴリズムを与える．

DTC 問題は DST 問題の特別な場合である．有向グラフ G における DTC とは，G 内の各辺からその head か tail もしくは両方を含む有向木である．DTC 問題の目的は，非負辺コスト付き有向グラフ G が与えられた時，最小コストの DTC を計算することである．前述の通り，DTC 問題についても SET COVER からの簡単な還元が存在し，二部グラフでも $\Omega(\log n)$ 倍近似困難であることが分かる [6]．この問題に対し，Nguyen は近似アルゴリズムを提示し，D^+ を G 内の頂点の最大出次数としたとき，その近似保証は $\max\{2, \ln(D^+)\}$ であると記している [12]．しかし，同保証の証明には明らかな誤りがあり，その主張は成り立たず，一般グラフにおける DTC 問題の近似可能性は未解決である．本稿では，DTC 問題を terminal 集合と non-terminal 集合への分割により定まる二部グラフ上の DST 問題に還元し，前述の近似アルゴリズムを用いることで，DTC 問題が $O(\log n)$ 倍近似可能であることを示す．

2 DIRECTED STEINER TREE

DST 問題では，有向グラフ $G = (V, A)$，辺コスト関数 $c : A \to \mathbb{R}_+$，根 $r \in V$，および terminal の集合 $X \subseteq V$ を入力とし，r から各 terminal へ至るパスを含む有向木（有向 Steiner tree，DST）の中で，最小コストのものを求める問題である．

本節では DST 問題に対し，L 制限 Steiner tree という制限され DST の中で最小なもののコストの $2\cdot(L-1)\cdot H(|X|)$ 倍以下のコストを持つ DST を出力する，そして L が定数であれば多項式時間である近似アルゴリズムを示す．またこの結果から，$V \setminus X$ が定数 L を超える長さのパスを誘導しないグラフにおいては，DST 問題が $2\cdot(L+1)\cdot H(|X|)$ 倍できることを示す．これは，入力に制限を加えた DST 問題についてこれまでに知られている結果をすべて含むものである．

2.1 Greedy algorithm

任意の DST を観察すると，それ（の辺集合）を根と葉のみが terminal である部分木に一意に分解できることが分かる．根と葉のみが terminal である木は full Steiner tree と呼ばれ，次のように定義される．

定義 1. *full Steiner Tree T* は次のような有向木である．

1. T の根 $r(T)$ は $X \cup \{r\}$ に含まれる.

2. T の葉はすべて X に含まれる.

3. そのほかの T の頂点は X に含まれない.

部分グラフ T の頂点集合と辺集合をそれぞれ $V(T)$, $A(T)$ と表す. T のコスト, つまり T 内の辺のコストの合計を $c(T)$ で表す. T が有向木のとき, $r(T)$ はその根を表し, $X(T)$ は T が含む $r(T)$ 以外の terminal の集合とする ($X(T) = X \cap (V(T) \setminus \{r(T)\})$).

定義 2. 木 T のコスト密度 $d(T)$ は木 T の中の *terminal* 一つあたりにかかるコストであり, $d(T) = c(T)/|X(T)|$ と定義する.

ある 2 つの有向木 T_1, T_2 を仮定する. ここでさらに T_1 のある頂点 $v_1 \in V(T_1)$ を根とする, $r(T_2)$ へのパスを含むような有向木が追加されたとする. すると $r(T_1)$ から $r(T_2)$ へのパスができることにより, $r(T_1)$ から T_1 と T_2 に含まれるすべての頂点へのパスができる. つまり, 新たに有向木を追加することによって T_1 と T_2 は $r(T_1)$ を根とする一つの木にまとめられたと考えることができる.

このことから, 最初はバラバラであった有向木の集合にさらに有向木を加えていくことで一つの有向木にまとめる Greedy アルゴリズムを考えることができる.

次の性質をもった G の部分グラフ Partial Solution を定義する.

定義 3. 次の性質を満たすグラフ G の部分グラフ $F = (V_F, A_F)$ を *Partial Solution* と呼び, *PSol* と表す.

1. F 内で $\delta^-(v) = 0$ である頂点 v の集合 R_F は非空で, $r \in R_F \subseteq X \cup \{r\}$ である.

2. F 内で, 各頂点は R_F のいずれかの頂点から到達可能である.

3. $V_F \supseteq X \cup \{r\}$

F 内で $x_k \in R_F$ から到達可能な頂点の集合を V_k と表す.

辺のない terminal のみで構成される部分グラフ $(X \cup \{r\}, \emptyset)$ は明らかに PSol である. 同様に $R_F = \{r\}$ である PSol には r からすべての terminal へのパスが存在するので DST が存在し, 深さ優先探索などでコストが $c(A_F)$ 以下の DST を構築できる.

さらにアルゴリズムで使用する操作を 2 つ定義する.

定義 4. 木 T を縮約するとは, T 内のすべての辺コストを 0 にし, T に含まれる $r(T)$ 以外の *terminal* を *terminal* 集合から取り除く操作である (すなわち, $c(e) \leftarrow 0\ \forall e \in A(T^*)$, $X \leftarrow X \setminus X(T)$).

定義 5. *metric closure* はすべての頂点対に対して, コストがその頂点間の最短パスと等しい辺をグラフ上に仮定する操作である.

DST 問題に対するグリーディアルゴリズムを示す. Charikar らのアルゴリズムでは, 根を入力で与えられた r に固定して, コスト密度のよい高さ一定の有向木を再帰的に計算し, 全ターミナルを含むまで繰り返し追加していくことで, Steiner tree を構築している. これに対し, 我々のアルゴリズムでは, その時点で r から到達不可な任意のターミナルを根とする高さ一定有向木の中から, コスト密度がよいものを計算し追加することを繰り返す.

Algorithm 1. *Greedy algorithm*

1. $F \leftarrow (X \cup \{r\}, \emptyset)$
2. $X_0 \leftarrow X$
3. $i \leftarrow 0$
4. repeat until $X_i = \emptyset$
   (a) metric closure を計算
   (b) $X \cup \{r\}$ 内の頂点を根とする木でコスト密度が最小な有向木 T^* を計算
   (c) $F \leftarrow F \cup T^*$
   (d) T^* を縮約. $c(e) \leftarrow 0 \forall e \in A(T^*)$, $X_{i+1} \leftarrow X_i \setminus X(T^*)$
   (e) $i \leftarrow i + 1$
5. F から DST を再構築

補題 1. *Algorithm 1* の各繰り返しの始めで F は *PSol* であり，$R_F = X_i \cup \{r\}$ である．

Proof. アルゴリズム開始時の F は $(X \cup \{r\}, \emptyset)$ であり，明らかに $R_F = X \cup \{r\}$ である PSol である．

i 番目の繰り返しで F は $R_F = X_i \cup \{r\}$ である PSol であるとする．i 番目の繰り返しで $x_k \in X \cup \{r\}$ を根とする有向木 T_i^* が F に加えられたとする．T_i^* が加えられたことによって，x_k から $X(T_i^*)$ の各 terminal へのパスでき，さらに $X(T^*)$ からのパスが存在した頂点集合にも x_k からのパスができる．よって $T_i^* \cup F$ は $R_F = \{r\} \cup (X_i \backslash X(T^*))$ である PSol となる．$X_i \backslash X(T^*)$ は X_{i+1} であるので，$i+1$ 番目の繰り返しにおいても F は $R_F = X_{i+1} \cup \{r\}$ であるような PSol である．

よって補題は成り立つ． □

補題 2. 有向木 T^* を多項式時間で見つけることができるならば，*Algorithm 1* は多項式時間で正しく動作する．

Proof. アルゴリズムの各繰り返しで明らかに X のサイズは一つ以上小さくなる．また T^* を多項式時間で見つけることができるならば，アルゴリズムの各ステップは多項式時間で動作する．よって，Algorithm 1 は多項式時間で動作する．

繰り返しが終了したとき，$X_i = \emptyset$ である．よって補題1より，繰り返しを抜けたあと F は $R_F = \{r\}$ である PSol である．そのような PSol からはコストが $c(A_F)$ 以下の DST を容易に作ることができる．

よって Algorithm 1 は正しく動作する． □

補題 3. コスト密度が $\alpha/|X_i|$ 以下の T^* を各繰り返しで見つけることができれば，*Algorithm 1* が出力する *DST* のコストは $\alpha \cdot H(|X|)$ 以下である．

Proof. $x_1, x_2, ..., x_{|X|}$ はアルゴリズムによって収縮されていった順にソートされた terminal とする．ある有向木 T^* が F に加えられた時，$X(T^*)$ が terminal 集合から取り除かれる．これはコスト $c(T^*)$ で $X(T^*)$ を解に加えたと考えることができ，各 terminal に $d(T^*)$ のコストを支払うことで，その terminal を解に加えたと考えることができる．したがって，アルゴリズム全体のコストはそれぞれの terminal に支払ったコストの総和と考えることができるので $\sum_{i=1}^{|X|} d(i)$ と表すことができる．ある x_i をカバーする繰り返しのときの $|X_i|$ は少なくとも $|X| - i + 1$ であるので，

$$
\begin{aligned}
\sum_{i=1}^{|X|} d(T_i) &\le \sum_{i=1}^{|X|} \alpha \cdot |X| - i + 1 \\
&= \alpha \sum_{i=1}^{|X|} \frac{1}{|X| - i + 1} \\
&= \alpha \cdot H(|X|)
\end{aligned}
$$

よって，コスト密度が $\alpha/|X|$ である T^* を見つけることができれば，Algorithm 1 はコストが $\alpha \cdot H(|X|)$ である DST を出力する． □

2.2 Finding minimum density Steiner tree

実際にコスト密度が良い木 T^* を見つけることは難しい．そこである制限を加えた DST について考える．

定義 6. *l-level* 木とは根から葉までに l より多くの辺が存在しない有向木である．

定義 7. 構成要素の *full Steiner tree* がすべて *l-level* 木である *DST* を l 制限 *Steiner tree* と呼ぶ．

グラフ G 内の r を根とする最小コスト L 制限 Steiner tree を T_{OPT_L} とする．また，full Steiner tree の中で最もコスト密度の良い L-level 木を T_L^* とする．本節では，Charikar らのアルゴリズムによってコスト密度が $d(T^*) \le (L-1) \cdot d(T_L^*)$ である T^* を見つけることができることを示し，さらに T_L^* のコスト密度を保証することで、コスト密度が $d(T^*) \le 2 \cdot (L-1) \cdot c(T_{OPT_L})/|X_i|$ の木を多項式時間で見つけることができることを示す．

terminal 集合 X から k 個の頂点を含む r を根とする有向木で，コスト密度最小の i-level 木のコスト密度を $d_{OPT}^{(i)}(k, r, X)$ と表す．Charikar らは [4] で次の補題を示した．

補題 4. *Charikar* らのアルゴリズムで，指定された頂点 r を根とする，$d(T_{BEST}) \le (i-1) \cdot d_{OPT}^{(i)}(k, r, X)$ であるような *i-level* 木 T_{BEST} を $O(n^i k^{2i})$ 時間で見つけることができる．

あらゆる k と $r' \in X \cup \{r\}$ について Charikar らのアルゴリズムを実行することで，次の系が導ける．

系 1. 定数 $L \geq 2$ に対して，$d(T^*) \leq (L-1) \cdot d(T_L^*)$ であるような *L-level* 木 T^* を見つける多項式時間近似アルゴリズムが存在する．

Algorithm 1 の各繰り返しで常に T_L^* を見つけることが出来たとしても，Algorithm 1 は T_{OPT_L} を出力するとは限らない．よって，Algorithm 1 の各繰り返しにおける T_L^* のコスト密度に対する保証を与える必要がある．

ある繰り返しでの F と T_{OPT_L} が与えられたとする．補題 1 より F は PSol であり，$R_F = \{r\} \cup \tilde{X}$ とする．このとき，T_{OPT_L} からコスト密度が $2 \cdot c(T_{OPT_L})/|\tilde{X}|$ であるような L-level 木 T^* を見つけることができることを示す．明らかに $d(T_L^*) \leq d(T^*)$ である．

F と T_{OPT_L} が与えられたとき，次のアルゴリズムでコスト密度の小さい T^* を得ることができる．

Algorithm 2. L 制限 *Steiner tree* の最適解 T_{OPT_L} から T^* を選ぶ

1. 集合族 $\mathcal{T} \leftarrow \emptyset$
2. x_i から，$i \neq j$ である最初の $v_j \in V_j$ にぶつかるまで T_{OPT_L} を逆向きにたどる．それにより求めたパスを $\mathcal{T}$ に追加する．これをすべての terminal に対して行う．
3. $\mathcal{T}$ 内のパスで同じ頂点から始まるパスは一つの有向木にまとめる．
4. $\mathcal{T}$ に含まれる有向木で最もコスト密度が良い有向木 T^* を一つ選ぶ．
5. T^* にある $\delta^+(v) = \delta^-(v) = 1$ である頂点を metric closure の辺を使って一本の辺に置き換える．
6. T^* の根を $v_j \in V_j$ から $r(V_j)$ に置き換える (縮約よってコストが 0 である辺 $(r(V_j), v_j)$ が存在するため，metric closure の辺を用いてコスト密度と level が変わらない有向木を作ることができる)．

この Algorithm 2 で得られた T^* のコスト密度が $2 \cdot c(T_{OPT_L})/|\tilde{X}|$ であるような L-level 木 T^* であることを証明する．

補題 5. *Algorithm 2* の木 $T \in \mathcal{T}$ において，T 内に $v_i \in V_i$ から x_i へのパスが存在するならば，T の v_i から x_i の間の頂点で T が分岐することはない (出次数が 1 より大きいものはない)．

Proof. T の v_i から x_i へのパスの間で T が分岐したとする．すると，Step 2 である頂点 $x_j \in \tilde{X}$ から v_i から x_i へのパスの間の頂点を通り，$v_i \in V_i$ を超えて $r(T)$ まで T_{OPT_L} を辿ったパスが存在することになる．これは Step 2 の操作に矛盾する． □

補題 6. *Algorithm 2* によって求まる T^* は *L-level* 木である．

Proof. Step 4 で選ばれたある T^* について考える．このときの T^* は T_{OPT_L} の部分グラフであり，一つの連結成分であるので T^* は有向木である．

さらに $r(T^*)$ からある T^* の葉の頂点までのパス長が L 以下であることを示すことによって，T^* が L-level 木であることを証明する．

Algorithm 2 より，T^* の葉は terminal である．$r(T^*)$ から葉である x_i まで T^* 上のパスを考える．Step 2 より，$r(T^*)$ から x_i へのパス上には $V_i \cup (V \setminus V_F)$ である頂点しか存在しない．また T_{OPT_L} は L 制限 Steiner tree であるので，$r(T^*)$ から距離 L 以内に頂点 $x' \in X \setminus \tilde{X}$ が存在する．$x' \in X$ であることから $x' \in V_F$ であるので，$x' \in V_i$ である．よって，補題 5 より T^* 上での x' から x_i へのパスに含まれる頂点で分岐はない．r から距離 L 以内に x' が存在し，x' から x_i へのパスは Step 5 で一本の辺に変換されることから，Step 5 での T^* は L-level Tree である．Step 6 では木の高さが変わらないため，補題は成り立つ． □

補題 7. 最小 L 制限 *Steiner tree* T_{OPT_L} から計算した T^* のコスト密度は $2 \cdot c(T_{OPT_L})/|\tilde{X}|$ 以下である．

Proof. Algorithm 2 で作られた集合族 $\mathcal{T}$ 全体のコスト密度が高々$(2 \cdot c(T_{OPT}))/|\tilde{X}|$ であることを示し，その中で最もコスト密度の良い部分集合を選択するという手順から補題を証明する．

Algorithm 2 によってすべての terminal が $\mathcal{T}$ 内の木にカバーされるので $\mathcal{T}$ は $|\tilde{X}|$ 個の terminal をカバーする．

Step 5，6 の操作ではコスト密度は良くなったとしても悪くなることはないので，Step 4 での $\mathcal{T}$ を考える．Algorithm 2 の手順より，木 $T \in \mathcal{T}$ の辺には T_{OPT_L} に含まれる辺しか含まれない．よって，ある共通の頂点を含む木 $T \in \mathcal{T}$ が高々2つまでしかないことを示すことによって $\mathcal{T}$ 全体のコストが $2 \cdot T_{OPT_L}$ 以下であることを示す．

$T \in \mathcal{T}$ は T_{OPT_L} の部分木である．よって，ある二つの木 $T_1, T_2 \in \mathcal{T}$ が共通の頂点をもつならば，$r(T_1) \in V(T_2)$ または $r(T_2) \in V(T_1)$ である．そうでないならば，$r(T_1) \notin V(T_2)$ かつ $r(T_2) \notin V(T_1)$ で T_1 と T_2 が共通の頂点を持つことになり，T_1 からの辺と T_2 からの辺の二つが入る入次数が 2 の頂点が T_{OPT_L} に存在することになるため，T_{OPT_L} が木であることに矛盾する．よって，各 $T \in \mathcal{T}$ の根となる頂点が高々2つの木にしか含まれないことを示せば，補題は成り立つ．

ある木 $T \in \mathcal{T}$ の根 $r(T)$ を考える．Step 2 より，$r(T) \in V_i$ と仮定できる．Step 2 で x_i 以外の x_j から辿ったパスが $r(T)$ まで来たとするならば，必ず $r(T)$ で止まる．よって頂点 $r(T)$ を超えて辿ることができるパスは x_i からのパスのみだったはずである．$r(T)$ で止まったパスは Step 3 によって T にまとめられるため，$r(T)$ を含むような木は T と x_i をカバーした木の高々2つしかない存在しない．よって，どの頂点も高々2つの木のみにしか含まれず，$\mathcal{T}$ のコストの合計は $2 \cdot c(T_{OPT_L})$ を超えない．

したがって，$\mathcal{T}$ 全体のコスト密度は $2 \cdot c(T_{OPT_L})/|\tilde{X}|$ である．$\mathcal{T}$ の中で最もコスト効果の高い部分集合が選択されることから，Algorithm 2 で選ばれる部分集合のコスト密度は $2 \cdot c(T_{OPT_L})/|\tilde{X}|$ を超えない． □

補題 8. *Charikar* らのアルゴリズムによって，コスト密度が $2 \cdot (L-1) \cdot c(T_{OPT_L})/|X_i|$ 以下の *L-level* 有向木 T_{BEST} を見つけることができる．

Proof. 補題 6，7 より，明らかに

$$d(T_L^*) \leq \frac{2 \cdot c(T_{OPT_L})}{|X_i|}$$

したがって，系 1 より

$$d(T_{BEST}) \leq 2 \cdot (L-1) \cdot \frac{c(T_{OPT_L})}{|X_i|}$$

よって補題は成り立つ． □

2.3 Approximating DIRECTED STEINER TREE

補題 3，8 より、ただちに次の定理 1 が成り立つ．

定理 1. *Algorithm 1* は，コストが $2 \cdot (L-1) \cdot H(|X|) \cdot c(T_{OPT_L})$ 以下の *DST* を出力し，L が定数であれば多項式時間で動作する．

L 制限 Steiner tree に対する近似比は Zelikovsky も与えており，その近似比は $(2 + \ln k)^{L-1}$ であるが [17]，定理 1 はこれを改善するものである．

一般の DST 問題に対する Algorithm 1 の近似比は，Charikar らの解析と同じ手法を適用でき，$2 \cdot i^2(i-1) \cdot |X|^{1/i}$ となる．これは Charikar らと同じく $O(k^{\epsilon})$ 倍近似であり，polylog 倍近似は得られていない．

DST 問題に対する近似アルゴリズムは，一般の場合だけではなく，入力グラフを制限した場合も研究されている．$V \setminus X$ が辺を誘導しないグラフは quasi-bipartite グラフと呼ばれるが，無向グラフにおける STEINER TREE 問題では，入力を quasi-bipartite グラフに制限することで，一般グラフ上よりも良い近似保証の得られることが繰り返し観察されている [13, 14, 2]．例えば Robins と Zelikovsky の結果では，一般グラフでは約 1.55 倍近似であったのに対し，quasi-bipartite グラフに制限すると約 1.28 倍近似が得られている [14]．有向グラフにおいては，SET COVER 問題からの簡単な還元より，quasi-bipartite グラフにおいても DST 問題は $\Omega(\log n)$ 倍近似困難であることが分かる．しかし，quasi-bipartite グラフにおける任意の DST は 2 制限 Steiner tree である．よって，次の系が導かれる．

系 2. *quasi-bipartite* グラフにおける *DST* 問題は，$2 \cdot H(|X|)$ 倍近似可能である．

つまり，一般グラフにおける $\Omega(\log^{2-\epsilon} n)$ 倍近似困難性 [9] と併せることで，有向グラフにおいては，

一般グラフと quasi-bipartite グラフとで近似可能性に差が生じることが示された．より一般には，次の系が成り立つ．

系 3. $V \setminus X$ が誘導するパスの長さが定数 L 以下であるグラフにおいて，*DST* 問題は $2\cdot(L+1)\cdot H(|X|)$ 倍近似可能である．

この系は，グラフ全体，あるいは $V \setminus X$ の深さがある定数で抑えられた場合について，これまでに得られている既知結果を包含する．

3 Directed Tree Cover

DTC 問題は，有向グラフ $G=(V,A)$，辺コスト関数 $c: A \to \mathbb{R}_+$，および根 $r \in V$ が与えられたとき，G 内の各辺について，その head か tail もしくは両方を含み r を根とする有向木でコスト最小なものを求める問題である．本節では DTC 問題から二部グラフ上の DST 問題への近似率保存帰着を示すことで，DTC 問題に対する $O(\log n)$ 倍近似アルゴリズムを示す．

DTC 問題の入力として，辺コスト付きの $G=(V,A)$ と根 $r \in V$ が与えられたとする．DTC 問題のインスタンス (G,r,c) から DST 問題のインスタンス (G',r',X,c') を作る．まず，G の各辺 (u,v) に対して terminal $x_{(u,v)} \in X$ を追加する．G' の頂点集合は $V' = V \cup X$ である．追加した頂点は全て terminal であり，もとあった頂点は全て non-terminal とする．根 r' は r と同じ頂点とする．そして，辺 (u,v) を削除し，代わりに同じコストの辺 $(x_{(u,v)},v)$ を追加する．さらにコストが 0 の辺 $(u,x_{(u,v)})$ と $(v,x_{(u,v)})$ を追加する．これらの辺の集合が G' の辺集合 A' となる．$|X|=|A|$，$|A'|=3|A|$ であり，明らかに多項式時間で構成可能である．

補題 9. G' は，$(X, V' \setminus X)$ を頂点集合の分割とする 2 部グラフである．

Proof. G の各辺 (u,v) について，terminal $x_{(u,v)} \in X$ を挿入するとともに，辺 (u,v) の代わりに，u，v ともに追加した terminal とつながる辺が加えられ，A' はそのような辺しか含まない．つまり，G' で V に含まれる頂点と隣接する頂点は terminal のみであり，terminal 同士が隣接することもない． □

補題 10. *DTC* 問題のインスタンス (G,r,c) から作成した *DST* 問題のインスタンス (G',r',X,c') について，G' における任意の *DST* T_{DST} から G における同コストの *DTC* T_{DTC} を多項式時間計算できる．また，G の任意の *DTC* に対し同コストの *DST* が G' に存在する．

Proof. G' 上の r' を根とする木 T' が与えられた時，G 上の T は T' が辺 $(u,x_{(u,v)})$ と $(x_{(u,v)},v)$ の両方を含んでいたときかつその時に限り，辺 (u,v) を含むとする．このような T は明らかに T' と同じコストを持つ r を根とする木であり，これは明らかに多項式時間で計算できる．

T' が DST 問題の実行可能解 T_{DST} であったとする．その T' に対する T を考える．T_{DST} がすべての $x_{(u,v)} \in X$ を含むことから，T' は u と v の少なくとも一方を含む．T' が含んでいた u と v は T に変換したときにもそのまま保持されるため，T は T_{DST} と同じコストである DTC 問題の実行可能解となる．

よって DTC 問題のインスタンス (G,r,c) から作成した DST 問題のインスタンス (G',r',X,c') について，G' における任意の DST T_{DST} から同じコストの G における DTC T_{DTC} を多項式時間で計算できる．

G 上の r を根とする木 T が与えられた時，G' 上の木 T' は T が辺 (u,v) を含んでいるときかつその時に限り，辺 $(u,x_{(u,v)})$，$(x_{(u,v)},v)$ を含むとする．このような T' は明らかに T と同じコストを持つ r' を根とする木である．

T が DTC 問題の実行可能解 T_{DTC} であったとする．その T に対する T' を考える．T_{DTC} の頂点集合が vertex cover であることから T' はすべての $x_{(u,v)} \in X$ に対して，u と v の少なくともどちらか一方を含む．よって，T' にコストが 0 である辺 $(u,x_{(u,v)})$ か $(v,x_{(u,v)})$ を適切に加えることで DST 問題の実行可能解 T_{DST} を作ることができる．

したがって DTC 問題のインスタンス (G,r,c) から作成した DST 問題のインスタンス (G',r',X,c') には任意の DTC T_{DTC} のコストと同じコストの G' における DST T_{DST} が存在する． □

定理 2. *DTC* 問題は $2\cdot H(|A|)$ 倍近似可能である．

Proof. 補題 10 より，DTC 問題を DST 問題に近似率保存帰着することができる．還元したグラフは補題 9 より，terminal である頂点と terminal でない頂点で分かれる 2 部グラフである．このグラフは明らかに quasi-bipartite グラフである．よって系 2 より DTC 問題を $2 \cdot H(|X|)$ 倍近似で解く事ができる． □

参考文献

[1] E. Arkin, M. Halldórsson, and R. Hassin. Approximating the tree and tour covers of a graph. *Inform. Process. Lett.*, 47:275–282, 1993.

[2] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. An improved LP-based approximation for Steiner tree. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pp. 583–592, 2010.

[3] G. Calinescu and A. Zelikovsky. The polymatroid Steiner problems. *J. Comb. Optim.*, 9(3):281–294, 2005.

[4] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed Steiner problems. *J. Algorithms*, 33(1):73–91, 1999.

[5] M. Chlehík and J. Chlebíková. The Steiner tree problem on graphs: Inapproximability results. *Theoretical Computer Science*, 406:207–214, 2008.

[6] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[7] T. Fujito. How to trim an mst: A 2-approximation algorithm for minimum cost tree cover. In *33rd ICALP*, Vol. 4051 of *LNCS*, pp. 431–442, 2006.

[8] M. Garey and D. Johnson. The rectilinear Steiner-tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32:826–834, 1977.

[9] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the 35th annual ACM Symposium on Theory of Computing*, pp. 585–594, 2003.

[10] R. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pp. 85–103. Plenum Press, New York, 1972.

[11] G. Konjevod. Directed Steiner trees, linear programs and randomized rounding. Manuscript, 8 pages, 2005.

[12] V. Nguyen. Approximating the minimum directed tree cover. In *International Conference on Combinatorial Optimization and Applications*, pp. 144–159, 2010.

[13] S. Rajagopalan and V. Vazirani. On the bidirected cut relaxation for the metric Steiner tree problem. In *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 742–751, 1999.

[14] G. Robins and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM J. Discret. Math.*, 19:122–134, 2005.

[15] T. Rothvoß. Directed steiner tree and the lasserre hierarchy. *Arxiv preprint arXiv:1111.5473*, 2011.

[16] C. Savage. Depth-first search and the vertex cover problem. *Inform. Process. Lett.*, 14(5):233–235, 1982.

[17] A. Zelikovsky. A series of approximation algorithms for the acyclic directed Steiner tree problem. *Algorithmica*, 18:99–110, 1997.

[18] L. Zosin and S. Khuller. On directed Steiner trees. In *Proceedings of the 13th annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 59–63, 2002.