数理解析研究所講究録
第1649巻 2009年 153-159

層別グラフにおける有向木被覆問題の近似について

豊橋技術科学大学大学院 工学研究科情報工学専攻 多田 哲馬 (Tetsuma Tada)
Department of Information and Computer Sciences,
Toyohashi University of Technology
豊橋技術科学大学 工学部情報工学系 藤戸 敏弘 (Toshihiro Fujito)
Department of Information and Computer Sciences,
Toyohashi University of Technology

概要

グラフの木被覆問題を一般化し，有向グラフにおける木被覆問題について考える．集合被覆問題が，層数2の層別グラフにおける有向木被覆問題として現れることから，問題を層別グラフに限定し，一般の t 層の場合に $O(\log^{t-1} n)$ 倍近似可能であることを示す．

1 イントロダクション

グラフ $G=(V,E)$ において，各辺 $e \in E$ は，e 自身と e に隣接する辺を支配するといい，G の各辺が辺集合 $D \subseteq E$ のある辺に支配されるとき，D は G の辺支配集合とよばれる．連結グラフ G における木 T は，その辺集合が G の辺支配集合を形成するとき，木被覆（あるいは，連結辺支配集合）とよばれるが，これは，T が張る頂点集合が G の頂点被覆であることと等価である．辺コスト付き連結グラフ G が与えられ，G の最小コスト木被覆を計算する問題を木被覆問題というが，辺コストが一定の場合，同問題は連結頂点被覆問題（つまり，連結グラフを誘導するような最小頂点被覆を求める問題）と等価であり，後者は頂点被覆と同等以上に計算困難であることから [7]，木被覆問題もNP困難である．

木被覆問題は，Arkin, Halldórsson, Hassin に導入され [1]，コスト一定の場合に2倍近似アルゴリズムを，一般コストの場合に3.55倍近似アルゴリズムが与えられたが，一定コストの場合には，Savage による2倍近似アルゴリズム [10] が既にえられていた．その後，一般コストの場合にもより良い近似アルゴリズムが開発され [9, 5]，現在，2倍近似可能であることが知られている [5]．

従来の無向グラフにおける木被覆問題の一般化として，有向グラフにおける木被覆問題が考えられる．ここでは，入力有向グラフにおいて，各辺の head もしくは tail を含む有向木が解となる（本論文では，各辺が根から葉の方向へ向かう木として定義する）．同問題に関しては今のところほとんど何も知られていないが [9]，辺コストが一定の場合，2倍近似することが可能である（[6] 参照）．ところが，一旦任意の非負コスト（実際には，0か1のコストでも）が許されると，有向木被覆問題は集合被覆問題を内包するので，集合被覆の近似困難性 [3] より，(NP $\not\subset$ DTIME($n^{O(\log\log n)}$) を仮定して）近似保証は $\Omega(\log n)$ となる．逆に，各辺への最短経路を合成するという単純な方法で，$|E|$ 倍近似を得るのは容易であるが，より良い方法として，有向シュタイナー木問題へ還元してやれば，Charikar らのアルゴリズムにより $O(n^{\epsilon})$ 近似可能である [2]．

前述のとおり，集合被覆問題は特別な有向木被覆問題として現れるが，ここでは，層別グラフという構造のグラフに限定される．そこで本論文では，層別グラフ上の有向木被覆問題について考察する．

1.1 諸定義

層別グラフ $G=(V,E)$ とは，以下の構造をもつグラフをいう．

1. 頂点集合 V は，$L_0, L_1, \ldots, L_t$ に分割される．すなわち，$V = L_0 \cup L_1 \cup \cdots \cup L_t$ で，$i \neq j$ ならば，$L_i \cap L_j = \emptyset$.

2. $|L_0| = 1$.

3. 各辺 $(u,v) \in E$ には，$u \in L_{i-1}$ かつ $v \in L_i$ となる i が存在する．つまり，辺集合 E は，$E_i \subseteq L_{i-1} \times L_i$ である $E_1, E_2, \cdots, E_t$ に分割される．

任意の集合被覆問題は，集合族を L_1，台集合を L_2 とし，集合-要素の包含関係を E_2 で表わし，L_0 の唯一の要素 r から L_1 の各要素へ，対応する集合のコストともつ辺を付けてやると，r を根とする有向木被覆問題で実現できる（ただし，E_2 の辺コストはすべて 0）．$t = 2$ である層別グラフ

そこで本論文では，層別グラフにおいて r を根とする有向木被覆問題について考察する．特に，$V = L_0 \cup L_1 \cup \cdots \cup L_t$ であるとき，G を t 層グラフ，G 上の有向木被覆問題を t 層 *DTC* という．G 内に，r から到達できない辺が存在すると，r を根とする有向木被覆も存在しえないので，そのようなケースは除外する．

頂点 u に接続する辺の集合を $\delta(u)$，u を始点とする辺の集合を $\delta^+(u)$ と表し，$X \subseteq V$ について $\delta(X) = \bigcup_{u \in X} \delta(u)$ である．任意のグラフ G について，$V(G)$ でその頂点集合を，$E(G)$ でその辺集合を表す．

2 部分木被覆への拡張と部分集合被覆への還元

部分集合被覆問題（PSC）では，通常の集合被覆インスタンス $(U, \mathcal{S})$ に加えて整数 k $(0 \leq k \leq |U|)$ が与えられ，k 個以上の要素を被覆する最小コスト部分集合族が求められる．

いま，$G = (V, E)$ を t 層 DTC の入力グラフとする．任意の頂点 $u \in V$ につき，G 内で u から到達可能な頂点により誘導される部分グラフを $G(u)$ と表記する．$G(u)$ は u を根とする層別グラフであり，特に $u \in L_i$ のとき，$(t-i)$ 層グラフであることがわかる．G における t 層 DTC を，次の問題に拡張する：

問題 2.1. 根指定部分有向木被覆問題（*RPDTC*）：インスタンスは，層別グラフ $G = (V, E)$，辺コスト $c : E \to \mathbb{Q}_+$，$u \in V$，および非負整数 k．いま $u \in L_i$ で，$G(u) = (V(u) = \{u\} \cup \bigcup_{j=i+1}^{t} L'_j, E(u) = \bigcup_{j=i+1}^{t} E'_j)$ とするとき，u を根とする有向木で，k 本以上の辺を支配するものを解とし，そのコストを最小化する問題．

明らかに，$u \in L_0, k = |E|$ のとき，RPDTC は G に対する t 層 DTC と一致する．以下では，頂点 u を根とする有向木を u-tree とよび，辺 (u,v) に v-tree を付けたしてできる木を (u,v)-tree とよぶ．

次に，RPDTC を PSC に還元する．RPDTC インスタンス (G, c, u, k) から，以下のように PSC インスタンス $(U, \mathcal{S}, k)$ を作成する．

- 台集合：u-tree により支配可能な辺，つまり u から到達可能な辺の集合，すなわち，$U = \delta(V(u))$.

- 集合族：$\mathcal{S} = \{\delta(V(T)) \mid T$ は E'_{i+1} の辺を高々1 本含む u-tree$\} = \{\delta(u)\} \cup \{\delta(u) \cup \delta(V(T)) \mid T$ は L'_{i+1} の頂点を根とする有向木$\}$.

- 集合コスト：$\delta(V(T)) \in \mathcal{S}$ のコスト（ただし T は E'_{i+1} の辺を高々1 本含む u-tree）は，$\sum_{e \in E(T)} c(e)$.

定理 2.2. 上記の *RPDTC* から *PSC* への還元は，近似比を保存する．

証明. いま T が RPDTC の解であるとする．T を，それぞれが E'_i の辺を 1 本だけ含み，互いに辺素な u-tree $T_1, T_2, \cdots, T_j$ に分解すると，集合族 $\mathcal{T} = \{\delta(V(T_1)), \delta(V(T_2)), \cdots, \delta(V(T_j))\}$ が被覆する辺集合 $\bigcup_{i=1}^{j} \delta(V(T_i)) = \delta(V(T))$ は T が支配する辺集合と一致するので，$\mathcal{T}$ は PSC の解となり，そのコストは $\sum_{i=1}^{j} c(T_i) = c(T)$ である．これより，RPDTC の任意の解には，同コストの PSC 解が存在し，よって，PSC の最適値が RPDTC のそれを超えることはない．

一方，PSC の任意の解 $\mathcal{T} = \{\delta(V(T_1)), \delta(V(T_2)), \cdots, \delta(V(T_j))\}$（ただし T は E'_{i+1} の辺を高々1 本含む u-tree）から，$T_1, T_2, \cdots, T_j$ を合成し，更に有向木でない場合は不必要な辺だけを取り除いてやれば，$\mathcal{T}$ が被覆する辺集合を支配するので，コストを $\mathcal{T}$ のそれから増やすことなく，RPDTC の解を得ることができる．従って，このようにして得られる RPDTC 解の最適値に対する比は，PSC のそれ以下になる． □

3 劣モジュラ被覆問題と Approximately Greedy アルゴリズム

前節の結果より，RPDTC を近似するには，RPDTC インスタンス (G, c, u, k) から PSC インスタンス $(U, \mathcal{S}, k)$ を構成し，PSC を近似してやればよい．PSC は，通常の SC と同様に，貪欲法により $O(\log n)$ 近似を保証できることが知られており [8, 11]，同手法を用いるのが妥当な選択である．しかしながら，ここでの PSC インスタンスに現れる部分集合の数は，一般に RPDTC インスタンスのサイズの指数オーダーとなるため，その中から真に greedy な選択をすることは困難である．そこで，本論文では近似的 greedy 選択を用いる解法による RPDTC の近似を試みる．本節では，そのための理論的枠組みを以下に示す．

ある有限集合 N について，関数 $f : 2^N \to \mathbb{R}_+$ は，非減少的かつ劣モジュラであるとする．つまり，$S, T \subseteq N$ について，$f(S) + f(T) \geq f(S \cap T) + f(S \cup T)$ であり，$S \subseteq T \subseteq N$ ならば $f(S) \leq f(T)$.

問題 3.1. 劣モジュラ被覆問題（Submodular Set Cover}：インスタンスは，有限集合 N，各要素 $j \in N$ のコスト c_j，および非減少的劣モジュラ関数 $f : 2^N \to \mathbb{Z}_+$（簡単のため，非負整数値関数とする）で，$f(S) = f(N)$ をみたす最小コスト集合 $S \subseteq N$ を計算する問題，つまり，

$$(SSC) \qquad \min_{S \subseteq N} \{\sum_{j \in S} c_j : f(S) = f(N)\}.$$

任意の $S \subseteq N$ について，$f_S(X) = f(X \cup S) - f(S)$ と定義される，2^{N-S} 上の関数 f_S を，f の $N - S$ 上への縮約という．$f_S(j)$ で $f_S(\{j\})$ を表す．

SSC 問題は，$f(\emptyset) = 0$ であるとき，グリーディー法により $H(\max_{j \in N} f(\{j\}))$ 倍近似を保証できることが知られているが（ただし，$H(k) = \sum_{i=1}^{k} \frac{1}{i}$）[12]，本論文ではグリーディー法の変型版を導入する．

アルゴリズム 3.2. *Approximately Greedy algorithm* AGreedy *for SSC:*

1. Set $t = 1, S^0 = \emptyset$.
2. While S^{t-1} is not a solution of SSC (i.e., $f(S^{t-1}) < f(N)$) do
3. /* Let $\theta^t = \min_{j \in N - S^{t-1}} \left\{ \frac{c_j}{f_{S^{t-1}}(j)} \right\}$ and $j_t = \text{argmin}_{j \in N - S^{t-1}} \left\{ \frac{c_j}{f_{S^{t-1}}(j)} \right\}$. */

4. Find $\tilde{j}_t \in N - S^{t-1}$ s.t. $\frac{c_{\tilde{j}_t}}{f_{S^{t-1}}(\tilde{j}_t)} = \tilde{\theta}^t \le r \cdot \theta^t$.

5. Set $S^t = S^{t-1} \cup \{\tilde{j}_t\}$ and $t = t + 1$.

6. Set $T = t - 1$ and output S^T.

アルゴリズム AGreedy は，通常のグリーディー法と同様に，S^{t-1} が SSC 解となるまで繰り返し要素を追加する．相違点は，各反復 t で追加される要素の選択にあり，通常のグリーディー法では，常にコスト効果 $\frac{c_j}{f_{S^{t-1}}(j)}$ が最も良い j_t を選択するところ，AGreedy では，最小値 θ^t の高々 r 倍のコスト効果をもつ $\tilde{j}_t$ でよしとするところにある．

定理 3.3. *SSC*問題に対する AGreedy は，$f(\emptyset) = 0$ であるとき，最小値の高々$r \cdot H(\max_{j\in N} f(\{j\}))$ 倍の解を計算する．

証明. SSC が，次の整数線形計画問題で定式化できることは，Wolsey によって示された [12]：

$$
\begin{array}{llll}
 & z_{\mathrm{IP}} = & \min \sum_{j\in N} c_j x_j & \\
 & \text{s.t.} & & \\
(\mathrm{IP}) & & \sum_{j\in N-S} f_S(j) x_j \ge f_S(N-S) & S \subseteq N \\
 & & x_j \in \{0,1\} & j \in N
\end{array}
$$

この IP の線形緩和として

$$
\begin{array}{llll}
 & z_{\mathrm{LP}} = & \min \sum_{j\in N} c_j x_j & \\
 & \text{s.t.} & & \\
(\mathrm{LP}) & & \sum_{j\in N-S^t} f_{S^t}(j) x_j \ge f_{S^t}(N-S^t) & t = 0,1,\cdots,T-1 \\
 & & x_j \ge 0 & j \in N
\end{array}
$$

を用いると，次の双対 LP がえられる：

$$
\begin{array}{llll}
 & z_{\mathrm{D}} = & \max \sum_{t=0}^{T-1} f_{S^t}(N-S^t) y_{S^t} & \\
 & \text{s.t.} & & \\
(\mathrm{D}) & & \sum_{S^t : j \notin S^t} f_{S^t}(j) y_{S^t} \le c_j & j \in N \\
 & & y_{S^t} \ge 0 & t = 0,1,\cdots,T-1
\end{array}
$$

f が非減少的劣モジュラであることから，$f_{S^{t-1}}(j) \ge f_{S^t}(j)$, $(\forall t \in \{0,1,\cdots,T-1\}, j \in N - S^t)$. これより，一般に $\tilde{j}_t \ne j_t$ ではあるが，$0 < \theta^1 \le \theta^2 \le \cdots \le \theta^T$. 一方，任意の $j \in N$ について，$f_{S^{k-1}}(j) > 0$ かつ $f_{S^k}(j) = 0$ となる $k \le T$ が存在し，$f_{S^0}(j) \ge f_{S^1}(j) \ge \cdots \ge f_{S^{k-1}}(j) > 0$ となるので，

$$
\theta^1 f_{S^0}(j) + (\theta^2 - \theta^1) f_{S^1}(j) + \cdots + (\theta^k - \theta^{k-1}) f_{S^{k-1}}(j) \le (\max_{1\le t\le k} \theta^t f_{S^{t-1}}(j)) H(f_{S^0}(j))
$$

が成り立ち (see Proposition 3 [12]) , $\theta^t \leq \frac{c_j}{f_{S^{t-1}}(j)}$, $t = 0, 1, \cdots, k-1$, かつ, $f_{S^0}(j) \leq \max_{i \in N} f_{S^0}(i)$ であるので,

$$\theta^1 f_{S^0}(j) + (\theta^2 - \theta^1) f_{S^1}(j) + \cdots + (\theta^k - \theta^{k-1}) f_{S^{k-1}}(j) \leq c_j H(\max_{i \in N} f_{S^0}(i)).$$

したがって, $y_{S^0} = \theta^1 / H(\max_{i \in N} f_{S^0}(i))$, $y_{S^t} = (\theta^{t+1} - \theta^t)/H(\max_{i \in N} f_{S^0}(i))$, $t = 1, 2, \cdots, T-1$, とおくと, y は双対実行可能解となる.

一方, AGreedy は各反復で $\tilde{j}_t$ を選択し, コストは $c_{\tilde{j}_t} = \tilde{\theta}^t f_{S^{t-1}}(\tilde{j}_t) = \tilde{\theta}^t (f(S^t) - f(S^{t-1}))$ だけ増加するが, $\tilde{\theta}^1 t \leq r \cdot \theta^t$, $\forall t$, でもある. よって, AGreedy の計算する SSC 解 S^T のコストは,

$$\begin{aligned}
c(S^T) &= \sum_{t=1}^{T} \tilde{\theta}^t (f(S^t) - f(S^{t-1})) \\
&\leq r \sum_{t=1}^{T} \theta^t (f(S^t) - f(S^{t-1})) \\
&= r \left[\theta^1 (f(N) - f(S^0)) + \sum_{t=2}^{T} (\theta^t - \theta^{t-1})(f(N) - f(S^{t-1})) \right] \\
&= rH(\max_j f(j)) \left[f_{S^0}(N - S^0)\theta^1 / H(\max_j f(j)) \right. \\
&\quad \left. + \sum_{t=2}^{T} f_{S^{t-1}}(N - S^{t-1})(\theta^t - \theta^{t-1}) / H(\max_j f(j)) \right]
\end{aligned}$$

となり, y の双対実行可能性より,

$$c(S^T) \leq rH(\max_j f(j)) \cdot z_{\mathrm{LP}} \leq rH(\max_j f(j)) \cdot z_{\mathrm{IP}}$$

がえられる. □

4 Approximately Greedy による RPDTC の近似

SSC は, 集合被覆だけでなく PSC も内包しているので ([4] 参照), AGreedy を PSC に適用すると, 定理 3.3 が成り立つ. 以下では, RPDTC インスタンス (G, c, u, k) から得られた PSC インスタンスを PSC(G, c, u, k) と表し, これに適用された AGreedy を AGreedy(G, c, u, k) と表記する. 再度, $G(u) = (V(u) = \{u\} \cup \bigcup_{j=i+1}^{t} L'_j, E(u) = \bigcup_{j=i+1}^{t} E'_j)$ とする. PSC に適用された AGreedy は, 繰り返し集合を選び, 解に加えていくが, PSC(G, c, u, k) から集合を選ぶことは, E'_{i+1} の辺を高々 1 本含む u-tree を選ぶことに他ならない. また, これらの有向木 T からの選択基準は, $c(T)/\min\{|\delta(V(T))|, k\}$ である.

いま T は, $\{u\}$ であるか, さもなければ E'_{i+1} の辺を 1 本だけ含むが, この辺を (u, v) に固定すると, T は (u, v) に v-tree T_v をつないだ (u, v)-tree であり,

$$\frac{c(T)}{\min\{|\delta(V(T))|, k\}} = \frac{c((u, v)) + c(T_v)}{\min\{(|\delta(u)| - 1) + |\delta(V(T_v))|, k\}}$$

である. 任意の v-tree T_v は, 最大で $|\delta(V(G(v)))|$ 本までの辺を支配可能であるが, $l = 0, 1, \cdots, \min\{|\delta(V(G(v)))|, k\}$ について, l 本以上の辺を支配する最小コストの v-tree を求

めれば，それらから $c(T)/\min\{|\delta(V(T))|,k\}$ を最小化する (u,v)-tree T^* を得ることができる．しかしここで，l 本以上の辺を支配するコスト最小の v-tree を求める問題は，(G,c,v,l) をインスタンスとする RPDTC に他ならない．そこで，この RPDTC を再度 PSC に還元し，得られる PSC(G,c,v,l) に（再帰的に）AGreedy を適用することで，各 l について l 本以上支配する v-tree を求め，それらの中で $c(T)/\min\{|\delta(V(T))|,k\}$ を最小化する (u,v)-tree $\tilde{T}$ を計算する．より詳細には，AGreedy のステップ 4：

Find $\tilde{j}_t \in N - S^{t-1}$ s.t. $\frac{c_{\tilde{j}_t}}{f_{S^{t-1}}(\tilde{j}_t)} = \tilde{\theta}^t \le r \cdot \theta^t$.

は，次のアルゴリズムで具体化される．

アルゴリズム 4.1. *Approximately Greedy Choice used in* AGreedy *for PSC(G,c,u,k)*
ただし，$G(u) = (V(u) = \{u\} \cup \bigcup_{j=i+1}^{t} L'_j, E(u) = \bigcup_{j=i+1}^{t} E'_j)$.

1. if $|\delta(u)| \ge k$ then return $\tilde{T} = \{u\}$ else $\tilde{\theta} = \infty$.
2. for $(u,v) \in E'_{i+1}$ do
3.   for $l = |\delta(v)|$ to $k - (|\delta(u)| - 1)$ do
4.   call AGreedy(G,c,v,l) and let $\bar{T}$ be the output
5.   Set $\bar{\theta} = (c((u,v)) + c(\bar{T}))/((|\delta(u)| - 1) + |\delta(V(\bar{T}))|)$.
6.   if $\tilde{\theta} > \bar{\theta}$ then set $\tilde{\theta} = \bar{\theta}$ and $\tilde{T} = \bar{T}$.
7. return $\tilde{T}_v$

ここで，ステップ 4. で呼び出される，PSC(G,c,v,k) に対する AGreedy の近似比を r 以下とすれば，

$$\frac{c(\tilde{T})}{\min\{|\delta(V(\tilde{T}))|,k\}} \le r \cdot \frac{c(T^*)}{\min\{|\delta(V(T^*))|,k\}}$$

となるので，定理 3.3 より，PSC(G,c,u,k) での近似比は $rH(m)$ 以下となる．

5 近似保証

インスタンス (G,c,u,k) の u が L_i に限定された RPDTC を $(t-i)$-RPDTC，対応す PSC を $(t-i)$-PSC と表す．前節の議論より，$(j+1)$-PSC に対する AGreedy の近似保証は，j-RPDTC に対する近似保証 r がいえれば，$rH(m) = O(r\log n)$ を導ける．

そこで，まず 1-RPDTC(G,c,u,k) を考える．すなわち，$G(u) = (V(u),E(u))$ で，$V(u) = \{u\} \cup L'_t$，$E(u) = \delta^+(u) \subseteq E_t$ である．いま，$|E(u)| = s$ として $L'_t = \{v_1, v_2, \cdots, v_s\}$，$E(u) = \{e_1 = (u,v_1), e_2 = (u,v_2), \cdots, e_s = (u,v_s)\}$ とおく．表 A$[1\ldots s,\ 0\ldots(|\delta(L'_t)| - s)]$ を用意し，

A$[i,j] = \{e_1, e_2 \cdots, e_i\}$ の辺を使い，$\delta(L'_t) - E(u)$ の辺を j 本支配するための最小コスト

を動的計画法で計算する．A$[i,0] = 0$ $(\forall i)$，A$[1,j] = c(e_1)$ (for $1 \le j \le |\delta(v_1) - 1|$)，その他は A$[i,j] = \infty$ と初期化し，次式により値（および対応する辺集合）の更新を繰り返し，全エントリを計算する：

$$A[i+1,j] = \min\{A[i,j], c(e_{i+1}) + A[i, j - (|\delta(v_{i+1})| - 1)]\}$$

1-RPDTC(G, c, u, k) のアルゴリズムは，まず，$|\delta(u)|$ と k を比較し，$|\delta(u)| \geq k$ であれば，$\{u\}$ を出力して終了する．そうでない場合（つまり $|\delta(u)| < k$），$\mathrm{A}[s, k - |\delta(u)|]$ に対応する辺集合から構成される木を出力する．このように，1-RPDTC は正確に解くことができ，よってこの場合 $r = 1$ である．

一般の RPDTC(G, c, u, k) に対し，$u \in L_i$ として，$i = t - 1$ であれば上記アルゴリズムを，$i \leq t - 2$ であれば AGreedy を実行すれば，近似比 $O(\log^{t-i-1} n)$ の解がえられる．よって，

定理 5.1. *t 層 DTC に対し，$O(\log^{t-1} n)$ 倍近似保証可能である．*

参考文献

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

[2] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, Approximation Algorithms for Directed Steiner Problems. J. Algorithms, 33:73-91, 1999.

[3] U. Feige, A Threshold of $\ln n$ for Approximating Set Cover. J. ACM 45(4):634–652, 1998.

[4] T. Fujito, On approximation of the submodular set cover problem. Oper. Res. Lett., 25:169–174, 1999.

[5] T. Fujito. On approximability of the independent/connected edge dominating set problems. Inform. Process. Lett., 79(6):261–266, 2001.

[6] T. Fujito. How to trim an MST: A 2-approximation algorithm for minimum cost tree cover, Proc. ICALP (1), 431-442, 2006.

[7] M.R. Garey and D.S. Johnson. The rectilinear Steiner-tree problem is NP-complete. SIAM J. Appl. Math., 32(4):826–834, 1977.

[8] M.J. Kearns, The Computational Complexity of Machine Learning, MIT Press, Cambridge, MA, 1990.

[9] J. Könemann, G. Konjevod, O. Parekh and A. Sinha. Improved approximations for tour and tree covers. Algorithmica, 38(3): 441–449, 2003.

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

[11] P. Slavík, Improved performance of the greedy algorithm for partial cover, Inform. Process. Lett., 64(5):251–254, 1997.

[12] L.A. Wolsey, An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica, 2(4):385–393, 1982.