Theoretical Computer Science 410 (2009) 661–669

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Construction of strongly connected dominating sets in asymmetric multihop wireless networks

Deying Li [a,b], Hongwei Du [c], Peng-Jun Wan [d,*], Xiaofeng Gao [e], Zhao Zhang [f], Weili Wu [e]

[a] Key Laboratory of Data Engineering and Knowledge Engineering (Renmin University of China), MOE, PR China
[b] School of Information, Renmin University of China, Beijing 100872, PR China
[c] Department of Computer Science, City University of Hong Kong, Kowloon Tang, Hong Kong
[d] Department of Computer Science, Illinois University of Technology, Chicago, IL 60616, USA
[e] Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA
[f] College of Mathematics and System Sciences, Xinjiang University, Urumqi, PR China

ARTICLE INFO

Article history:
Received 8 July 2008
Received in revised form 21 September 2008
Accepted 25 September 2008
Communicated by D.-Z. Du

Keywords:
Asymmetric wireless network
Strongly connected dominating set

ABSTRACT

Consider an asymmetric wireless network represented by a digraph $G = (V, E)$. A subset of vertices U is called a strongly connected dominating set (SCDS) if the subgraph induced by U is strongly connected and every vertex not in U has both an in-neighbor in U and an out-neighbor in U. SCDS plays an important role of the virtual backbone in asymmetric wireless networks. Motivated by the construction of a small virtual backbone, we study the problem Minimum SCDS, which seeks a smallest SCDS of a digraph. For any constant $0 < \rho < 1$, there is no polynomial-time $\rho \ln n$-approximation for Minimum SCDS unless $NP \subseteq DTIME(n^{o(\ln n)})$, where n is the number of nodes. However, none of the polynomial-time heuristics for Minimum SCDS proposed in the literature are logarithmic approximations. In this paper, we present a polynomial-time $(3H(n-1)-1)$-approximation algorithm for Minimum SCDS, where H is the harmonic function. The approximation ratio of this algorithm is thus within a factor of 3 from the best possible approximation ratio achievable by any polynomial-time algorithm.

1. Introduction

Virtual backbone is a fundamental structure in multihop wireless networks with a broad range of applications (cf. a recent survey [3] and the references therein). A virtual backbone is a subset U of nodes satisfying that any pair of non-adjacent nodes can communicate with each other only through the nodes in U. Virtual backbone in symmetric multihop wireless networks has been extensively studied in the literature [2,4,11,13–15,19]. A symmetric multihop wireless network can be represented by an undirected graph $G = (V, E)$, and a virtual backbone is exactly a connected dominating set (CDS) of G, which is a subset U of nodes such that the subgraph induced by U is connected and each node not in U is adjacent to some node in U. The problem Minimum CDS seeks a CDS of the smallest cardinality in a specified graph. Both the approximation hardness and the approximation algorithms for Minimum CDS have been well studied. For any constant $0 < \rho < 1$, there is no polynomial-time $\rho \ln n$-approximation for Minimum CDS unless $NP \subseteq DTIME(n^{o(\ln n)})$ [8], where n is the number of nodes. A greedy $(\ln \Delta + 3)$-approximation and a greedy $(\ln \Delta + 2)$-approximation for Minimum CDS were presented in [8] and [12] respectively, where Δ is the maximum degree in the graph. Furthermore, when restricted on unit-disk graphs,

* Corresponding author.
E-mail addresses: deyingli@ruc.edu.cn (D. Li), hongwei@cs.cityu.edu.hk (H. Du), wan@cs.iit.edu (P.-J. Wan), xxg05200@utdallas.edu (X. Gao), zhzhao@xju.edu.cn (Z. Zhang), weiliwu@utdallas.edu (W. Wu).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2008.09.058

Minimum CDS admits a polynomial-time approximation scheme [6], i.e., for any $\varepsilon > 0$, there exists a polynomial-time $(1+\varepsilon)$-approximation, and efficient distributed constant-approximations [1,5,10,18].

In many scenarios, the communication links in multihop wireless networks are asymmetric in nature. For example, the nodes may have different transmission ranges because of the heterogeneity of the nodes, or because of the range assignment for the purpose of energy conservation. In such cases, it is possible for a pair of nodes u and v to have a communication link from u to v and no communication link from v to u. A multihop wireless network with asymmetric communication links, referred to as asymmetric multihop wireless network, can only be modelled by a directed graph $G = (V, E)$. Correspondingly, a virtual backbone is a *strongly connected dominating set* (SCDS) of G, which is a subset U of nodes such that the sub-digraph induced by U is strongly connected and each node not in U has both an in-neighbor in U and an out-neighbor in U. The problem Minimum SCDS seeks an SCDS of the smallest cardinality in a specified digraph. Since SCDS is a generalization of CDS, Minimum SCDS at least as hard as Minimum CDS in terms of approximality. Therefore, for any constant $0 < \rho < 1$ there is no polynomial-time $\rho \ln n$-approximation for Minimum SCDS unless $NP \subseteq DTIME(n^{o(\ln n)})$.

In contrast to the existence of many provably-good approximation algorithms for Minimum CDS in the literature, there are only a few approximation algorithms for Minimum SCDS in the literature. Thai *et al.* [16,17,7] gave several approximation algorithms for Minimum SCDS. But none of these algorithms are logarithmic approximations. In this paper, we presented a polynomial-time $(3H(n-1)-1)$-approximation algorithm for Minimum SCDS, where H is the harmonic function. The approximation ratio of this algorithm is thus within a factor of 3 from the best possible approximation ratio achievable by any polynomial-time algorithm.

2. Preliminaries

Let $G = (V, E)$ be a digraph. A node in G is said to be a *sink* node if its out-degree is zero, and an *internal* node otherwise. The set of internal nodes in G is denoted by $I(G)$. A subgraph is said to be spanning if its vertex set is exactly V. A subgraph of G is called as an *arborescence* rooted at a node $s \in V$ if in this subgraph the in-degree of s is zero, and the in-degree of any other node is exactly one. An arborescence rooted at s is also called an s-arborescence. For each node $v \in V$, $\delta^-(v)$ (respectively, $\delta^+(v)$) denotes the in-degree (respectively, out-degree) of v in G, and $N^-(v)$ (respectively, $N^+(v)$) denotes the set of in-neighbors (respectively, out-neighbors) of v in G. For any subset U of V, we denote by $G[U]$ the subgraph of G induced by U. The *reverse* of G denoted by G^R is a digraph obtained from G by reversing the direction of every arc. In other words, $G^R = (V, E^R)$ with

$$E^R = \{(u, v) \mid (v, u) \in E\}.$$

Lemma 2.1. *Let $G = (V, E)$ be a strongly connected digraph and s be an arbitrary node in V. Suppose that T_1 is a spanning s-arborescence in G, and T_2 is a spanning s-arborescence in G^R. Then $I(T_1) \cup I(T_2)$ is an SCDS of G.*

Proof. Let $U = I(T_1) \cup I(T_2)$. We first show that $G[U]$ is strongly connected. Let u and v be two arbitrary distinct nodes in U. T_2^R contains a path P_2 from u to s, and T_1 contains a path P_1 from s to v. Both of P_1 and P_2 are in $G[U]$. The concatenation of P_2 and P_1 is a path from u to v in $G[U]$.

Next, consider a node $u \in V \setminus U$. u must be a sink both in T_1 and in T_2. Let u_1 be the parent of u in T_1, and u_2 be the parent of u in T_2. Then, both u_1 and u_2 belong to U. In addition, u_1 is an in-neighbor of u in G, and u_2 is an out-neighbor of u in G. ■

Motivated by Lemma 2.1, we introduce the problem Spanning Arborescence with Fewest Internal Nodes (SAFIN): Given a digraph $G = (V, E)$ and a source node $s \in V$, compute a spanning s-arborescence T with minimum $|I(T) \setminus \{s\}|$. It is easy to argue that SAFIN is at least as hard as Minimum CDS. Let $\mathcal{A}$ be an arbitrary polynomial-time approximation for SAFIN. Table 1 describes a polynomial-time approximation algorithm **SCDS** $(\mathcal{A})$ for Minimum SCDS with the parametric $\mathcal{A}$ as a subroutine. For the purpose of reducing the running time, a small subset S of candidates for the source node is selected as the beginning. Specifically, let u be the node u which minimize satisfying that $\min\left(\delta^-(v), \delta^+(v)\right)$ over all nodes $v \in V$. If $\delta^-(u) \le \delta^+(u)$, then S consists of u and all its in-neighbors; otherwise, S consists of u and all its out-neighbors. Clearly, every SCDS must contain at least one node in the selected S.

Theorem 2.2. *Suppose that $\mathcal{A}$ is a ρ-approximation for SAFIN. Then Algorithm* **SCDS** $(\mathcal{A})$ *produces an SCDS of size at most $2\rho \cdot opt - 2\rho + 1$, where opt is the size of a minimum SCDS.*

Proof. Let U^* be a minimum SCDS in G. Then, $U^* \cap S \neq \emptyset$. Consider a node $s \in U^* \cap S$. Then both $G[U^*]$ and $G^R[U^*]$ are strongly connected. Let T_1' (respectively, T_2') be a spanning arborescence of $G[U^*]$ (respectively, $G^R[U^*]$) with root s. Note that every node $v \in V \setminus U^*$ has an incoming neighbor $v_1 \in U^*$ (respectively, $v_2 \in U^*$) in G (respectively, G^R). Let T_1'' (respectively, T_2'') be the arborescence expanded from T_1' (respectively, T_2') by adding an arc (v_1, v) (respectively, (v_2, v)) for each $v \in V \setminus U^*$. Then, T_1'' (respectively, T_2'') is a spanning arborescence of G (respectively, G^R) rooted at s, and both $I\left(T_2'\right)$ and $I\left(T_2''\right)$ are contained in U^*. Hence,

$$\left|I\left(T_1''\right)\right| \le \left|U^*\right| = opt,$$
$$\left|I\left(T_2''\right)\right| \le \left|U^*\right| = opt.$$

Table 1
Outline of the algorithm **SCDS** ($\mathcal{A}$).

```
Algorithm SCDS (A):
  u ← arg min_{v∈V} min (δ⁻ (v) , δ⁺ (v));
  If δ⁻ (u) ≤ δ⁺ (u) then
    S ← {u} ∪ N⁻ (u) ;
  else
    S ← {u} ∪ N⁺ (u) ;
  U ← V;
  for each s ∈ S,
    T1 ← spanning s-arborescence in G output by A;
    T2 ← spanning s-arborescence in G^R output by A;
    if |I(T1) ∪ I(T2)| < |U| then
      U ← I(T1) ∪ I(T2);
  output U.
```

Let T_1 (respectively, T_2) be the spanning s-arborescence of G (respectively, G^R) output by the algorithm $\mathcal{A}$. Then,

$$|I(T_1) \setminus \{s\}| \leq \rho \left|I\left(T_1''\right) \setminus \{s\}\right| \leq \rho\,(opt-1)\,,$$
$$|I(T_2) \setminus \{s\}| \leq \rho \left|I\left(T_2''\right) \setminus \{s\}\right| \leq \rho\,(opt-1)\,.$$

Note that $I(T_1)$ and $I(T_2)$ have node s in common. So, for the output U of Algorithm **SCDS** ($\mathcal{A}$), we have

$$\begin{aligned}|U| &\leq |I(T_1) \cup I(T_2)| \\ &\leq 1 + |I(T_1) \setminus \{s\}| + |I(T_2) \setminus \{s\}| \\ &\leq 1 + 2\rho\,(opt-1)\,\rho \\ &= 2\rho \cdot opt - 2\rho + 1. \quad \blacksquare\end{aligned}$$

In the next section, we will develop a $(1.5H(n-1)-0.5)$-approximation algorithm $\mathcal{A}$ for SAFIN. For such $\mathcal{A}$, the algorithm **SCDS** ($\mathcal{A}$) is a $(3H(n-1)-1)$-approximation algorithm for Minimum SCDS by Theorem 2.2.

3. The approximation algorithm

First, we introduce some notations and terminologies. We give a unique ID to each node. Let s be the source node. For any $B \subseteq V$ containing s, let $G\langle B\rangle$ denote the spanning subgraph of G whose arc set consists of all arcs of G leaving from the nodes in B. A strongly-connected component of $G\langle B\rangle$ is said to be an *orphan* with respect to B if it neither contains the source s nor has an incoming arc. For each orphan component, the node with the smallest ID in this component is referred to as its *head*. We use $h(B)$ to denote the number of heads (equivalently, the number of orphans) with respect to B. Clearly, $G\langle B\rangle$ contains a spanning s-arborescence if and only if $h(B) = 0$.

Now, we give an overview on the algorithm $\mathcal{A}$ for SAFIN. The algorithm maintains a set B of nodes, which is initialized to $\{s\}$. Repeat the following iteration while $h(B) > 0$. Choose an arborescence by a greedy strategy, add all internal nodes of the chosen arborescence to B, and then update $h(B)$. When $h(B) = 0$, output a spanning s-arborescence in $G\langle B\rangle$. The key ingredient of this algorithm is the greedy strategy used by each iteration to select a proper arborescence. In the next, we will present the details of this greedy strategy.

Fix a subset B of V with $s \in B$ and $h(B) > 0$. The price of an arborescence T with respect to B is defined as

$$p(T) = \frac{|I(T) \setminus \{s\}\,|}{\text{the number of heads w.r.t. } B \text{ in } T}$$

the ratio of $|I(T) \setminus \{s\}\,|$ to the number of heads w.r.t. B contained in T. Ideally, one would greedily wish to use a cheapest (i.e., least-priced) arborescence to merge the orphans. While a cheapest arborescence can be computed easily when $h(B)$ is small, it is hard to be computed in general. Our approach is to restrict our selection of a cheapest arborescence from a polynomial number of special candidates. We use $\mathcal{T}(B)$ to denote the set of candidates. The construction of $\mathcal{T}(B)$ is described below.

We begin with some preprocessing. We use $P(u, v)$ to denote a shortest path $P(u, v)$ in G from u to v. For each node u and each pair of distinct nodes v and w, $S(u; v, w)$ denotes a u-arborescence in G containing v and w with the smallest number of internal nodes. Both $P(u, v)$ and $S(u; v, w)$ can be computed in polynomial time. Note that then $S(u; u, w)$ is identical to $P(u, w)$.

When $h(B) = 1$, $\mathcal{T}(B)$ consists of only one candidate $P(s, v)$ where v is the head. Similarly, when $h(B) = 2$, $\mathcal{T}(B)$ consists of only one candidate $S(s; v, w)$ where v and w are the two heads. Now, suppose that $h(B) \geq 3$. $\mathcal{T}(B)$ consists of $(h(B)-2)\,n+2$ candidates:

$$\begin{aligned}\mathcal{T}(B) = &\{T_\ell(B, s) : 1 \leq \ell \leq 2\} \cup \\ &\{T_\ell(B, u) : u \in V, 3 \leq \ell \leq h(B)\}\,.\end{aligned}$$

Table 2
Outline of the algorithm $\mathcal{A}$.

```
Algorithm A:
  B ← {s};
  while h(B) > 0,
    construct 𝒯(B);
    find a cheapest T ∈ 𝒯(B);
    B ← B ∪ I(T);
  output a spanning s-arborescence of G⟨B⟩.
```

The candidate $T_1(B, s)$ is the shortest one among all paths $P(s, v)$ where v is the head, and $T_2(B, s)$ is the one with the fewest internal nodes among all $S(s; v, w)$ where v and w are two distinct heads. The construction of each $T_\ell(B, u)$ with $u \in V$ and $3 \le \ell \le h(B)$ proceeds in three steps. In the first step, an edge-weighted graph $G_\ell(B, u)$ is constructed as follows.

Case 1: ℓ is even. Let X denote the set of all heads, and Y be a set of $h - \ell$ dummy nodes disjoint from V. $G_\ell(B, u)$ is the union of the clique on X and the bipartite clique between X and Y. Each edge vw between two heads has weight $c(vw) = |I(S(u; v, w))| - 1$; each other edge e has weight $c(e) = 0$.

Case 2: ℓ is odd and u is a head. Let X denote the set of all heads except v, and Y be a set of $h - \ell$ dummy nodes disjoint from V. $G_\ell(B, u)$ is the union of the clique on X and the bipartite clique between X and Y. Each edge vw between two heads has weight $c(vw) = |I(S(u; v, w))| - 1$; each other edge e has weight $c(e) = 0$.

Case 3: ℓ is odd and u is not a head. Let X denote the set of all heads, Y be a set of $h - \ell + 1$ dummy nodes disjoint from V, and y be a node in Y. $G_\ell(B, u)$ is the union of the clique on X and the bipartite clique between X and Y. Each edge vw between two heads has weight $c(vw) = |I(S(u; v, w))| - 1$; each edge vy has weight $c(vy) = |I(P(u, v))| - 1$; and each other edge e has weight $c(e) = 0$.

Clearly, $|X| + |Y|$ is even and $G_\ell(B, u)$ has a perfect matching. In the second step, we compute a minimum-weight perfect matching $M_\ell(B, u)$ in $G_\ell(B, u)$, and construct a subgraph $G'_\ell(B, u)$ of G as follows. If ℓ is even or if ℓ is odd and u is a head, $G'_\ell(B, u)$ is the union of $S(u; v, w)$ for all matched pairs of heads v and w in $M_\ell(B, v)$. If ℓ is odd and u is not a head, $G'_\ell(B, u)$ is the union of $S(u; v, w)$ for all matched pairs of heads v and w in $M_\ell(B, v)$ and the path $P(v, x)$ for the head x matched with y in $M_\ell(B, u)$. In the third step, we compute a spanning u-arborescence $T_\ell(B, u)$ of $G'_\ell(B, u)$.

Now, we are ready to describe the algorithm $\mathcal{A}$ (Table 2).

In the remaining part of this section, we derive an upper bound on the approximation ratio of the algorithm $\mathcal{A}$, which is stated in the next theorem.

Theorem 3.1. *The approximation ratio of the algorithm $\mathcal{A}$ for SAFIN is at most $1.5H(n-1) - 0.5$.*

To prove the above theorem, we first give a lower bound on the "gain" of each candidate arborescence in $\mathcal{T}(B)$ in terms of the reduction on the number of orphans (or heads).

Lemma 3.2. *Suppose that $h(B) > 0$. Then for any $T \in \mathcal{T}(B)$ containing ℓ heads w.r.t. B,*

$$h(B) - h(B \cup I(T)) \ge \frac{2}{3}\ell.$$

Proof. Let u be the root of T. After the addition of $I(T)$, each orphan w.r.t. B containing a head in T either till survives as a component of $G\langle B \cup I(T)\rangle$ but is not orphan any more, or gets merged into some new (and larger) component of $G\langle B \cup I(T)\rangle$. Furthermore, each new component of $G\langle B \cup I(T)\rangle$ which does not contain u cannot be an orphan. If $u \ne s$, then $\ell \ge 3$ and

$$h(B) - h(B \cup I(T)) \ge \ell - 1 > \frac{2}{3}\ell$$

the reduction in the number of orphan components is at least $\ell - 1 \ge 2\ell/3$ since $\ell \ge 3$. If $u = s$, then the component of $G\langle B \cup I(T)\rangle$ containing u is not an orphan, and

$$h(B) - h(B \cup I(T)) = \ell > \frac{2}{3}\ell.$$

So the lemma holds. ■

The next lemma presents an upper bound on the price of the cheapest candidate in $\mathcal{T}(B)$.

Lemma 3.3. *Suppose that $h(B) > 0$. Then the price of the cheapest arborescence in $\mathcal{T}(B)$ is at most $\frac{opt}{h(B)}$.*

The proof of this lemma is quite involved. We introduce an intermediate structure called legal arborescence. Let T be a u-arborescence in G. Removing u from T results in a collection of arborescences. Each of these arborescences is called a child arborescence of T. T is said to be *legal* with respect to a subset B if (1) T contains at least one (respectively three) head w.r.t. B if $u = s$ (respectively, $u \ne s$), and (2) each child arborescence of T contains at most two heads w.r.t. B, and (3) either $s = u$ or s is not in T. As the first step towards the proof of Lemma 3.3, we first establish a relation between the prices of a legal arborescence and the corresponding candidate arborescence in the next lemma.

Lemma 3.4. *Suppose that $h(B) > 0$. Then for any legal u-arborescence $T \subseteq G$ which contains ℓ heads,*

$$p(T) \geq p(T_\ell(B, u)).$$

Proof. We first show that the lemma holds when $\ell \leq 2$. When $\ell \leq 2$, $u = s$ and $T_\ell(B, s)$ contains exactly ℓ heads. If $\ell = 1$, let v be the head in T. Then,

$$\begin{aligned} p(T) &= |I(T)| - 1 \\ &\geq |I(P(s, v))| - 1 \\ &\geq |I(T_1(B, s))| - 1 \\ &= p(T_\ell(B, s)). \end{aligned}$$

Now, assume $\ell = 2$. Let v and w be the two heads in T. Then,

$$\begin{aligned} p(T) &= \frac{|I(T)| - 1}{2} \\ &\geq \frac{|I(S(s; v, w))| - 1}{2} \\ &\geq \frac{|I(T_2(B, s))| - 1}{2} \\ &= p(T_2(B, s)). \end{aligned}$$

So, the lemma holds when $\ell \leq 2$.

In the rest of the proof, we assume that $\ell \geq 2$. We show that

$$|I(T_\ell(B, u)) \setminus \{u\}| \leq c(M_\ell(B, u)). \tag{1}$$

We only give the proof for the inequality (1) in the case of even ℓ, as the proofs in other cases are similar. Suppose that ℓ is even. Let $v_i w_i$ for $1 \leq i \leq \frac{\ell}{2}$ be the matched pairs of heads in $M_\ell(B, u)$. Then,

$$\begin{aligned} |I(T_\ell(B, u)) \setminus \{u\}| &\leq \left|I(G'_\ell(B, u)) \setminus \{u\}\right| \\ &\leq \sum_{i=1}^{\frac{\ell}{2}} |I(S(u; v_i, w_i)) \setminus \{u\}| \\ &= \sum_{i=1}^{\frac{\ell}{2}} (|I(S(u; v_i, w_i))| - 1) \\ &= c(M_\ell(B, u)). \end{aligned}$$

So, the inequality (1) holds.

Next, we show that

$$|I(T) \setminus \{u\}| \geq c(M_\ell(B, u)). \tag{2}$$

We construct a perfect matching M' in $G_\ell(B, u)$ from T as follows.

Case 1: ℓ is even. Pair up every pair of heads in a same child arborescence of T, pair up the other heads in T arbitrarily, and pair up every head not in T with a unique dummy node.

Case 2: ℓ is odd and u is a head. Pair up every pair of heads in a same child arborescence of T, pair up the other heads in T except u arbitrarily, and pair up every head not in T with a unique dummy node.

Case 3: ℓ is odd and u is not a head. Pair up every pair of heads in a same child arborescence of T, and pairing the other heads in T arbitrarily, leaving exactly one head in T (say x) unpaired. Pair x with y, and pair up every head not in T with a unique dummy node other than y.

We claim that

$$|I(T) \setminus \{u\}| \geq c\left(M'\right).$$

We verify this claim only for Case 1, as the claim in the other two cases can be verified in the same way. Let $v_i w_i$ for $1 \leq i \leq \frac{\ell}{2}$ be the matched pairs of heads in M'. For each $1 \leq i \leq \frac{\ell}{2}$, let T_i be the union of the two paths from u to v_i and w_i respectively in T. Then, these $\frac{\ell}{2}$ arborescences T_i are internally node-disjoint. On the other hand, for each $1 \leq i \leq \frac{\ell}{2}$,

$$|I(T_i)| \geq |I(S(u, v_i, w_i))|.$$

Hence,

$$
\begin{aligned}
|I(T) \setminus \{u\}| &= \sum_{i=1}^{\frac{\ell}{2}} |I(T_i) \setminus \{u\}| \\
&= \sum_{i=1}^{\frac{\ell}{2}} (|I(T_i)| - 1) \\
&\geq \sum_{i=1}^{\frac{\ell}{2}} (|I(S(u, v_i, w_i))| - 1) \\
&= c\left(M'\right).
\end{aligned}
$$

So, our claim is true. Since

$$C\left(M'\right) \geq c\left(M_\ell(B, u)\right),$$

inequality (2) holds.

Finally, we prove the inequality given in the lemma. The two inequalities (1) and (2) imply that

$$|I(T_\ell(B, u)) \setminus \{u\}| \leq |I(T) \setminus \{u\}|.$$

By construction, the number of heads in each $T_\ell(B, u)$ is at least ℓ. If $u = s$, then

$$
\begin{aligned}
p\left((T_\ell(B, s))\right) &\leq \frac{|I(T_\ell(B, s)) \setminus \{s\}|}{\ell} \\
&\leq \frac{|I(T) \setminus \{s\}|}{\ell} \\
&= p(T).
\end{aligned}
$$

So, the lemma holds if $u = s$. Next, assume that $u \neq s$. Then, s is not in T and hence

$$
\begin{aligned}
p\left((T_\ell(B, u))\right) &\leq \frac{|I(T_\ell(B, u)) \setminus \{s\}|}{\ell} \\
&\leq \frac{|I(T_\ell(B, u))|}{\ell} \\
&= \frac{1 + |I(T_\ell(B, u)) \setminus \{u\}|}{\ell} \\
&\leq \frac{1 + |I(T) \setminus \{u\}|}{\ell} \\
&= \frac{|I(T)|}{\ell} \\
&= \frac{|I(T) \setminus \{s\}|}{\ell} \\
&= p(T).
\end{aligned}
$$

So, the lemma also holds when $u \neq s$. ∎

As the second step towards the proof of Lemma 3.3, we establish an upper bound on the price of the cheapest legal arborescence.

Lemma 3.5. *Suppose that $h(B) > 0$. Then, there is a legal arborescence with price at most $\frac{opt}{h(B)}$.*

Proof. We prove the lemma by a decomposition argument. Let T^* be an optimal spanning s-arborescence. The depth of a node in T^* is its hop-distance from s in T^*. Initialize $i = 0$. Repeat the following iteration while T^* contains at least three heads. Increment i by one. Choose a node v with the maximum depth such that the maximal v-arborescence contained in T^* (which is the subgraph of T^* induced by v and all its descendents in T^*) contains at least three heads. Set T_i^* to be the maximal v-arborescence contained in T^*. By the maximum depth of v, no child arborescence of T_i has at most two heads. Hence T_i is legal. Delete T_i^* from T^*. If T^* still contains one head or two heads, then increment i by one and set T_i^* to the whole T^*. Note that T_i^* is also legal since its root is s.

Let $T_1^*, T_2^*, \ldots, T_q^*$ be the sequence of legal arborescences obtained by this construction. Then, they are node-disjoint. In addition, their union contains all heads. For each $1 \leq j \leq q$, let ℓ_j denote the number of heads contained in T_j^*. Then,

$$\sum_{j=1}^{q} \ell_j = h(B),$$

and

$$\sum_{j=1}^{q} \left|I(T_j^*) \setminus \{s\}\right| = \left|I\left(\cup_{j=1}^{q} T_j^*\right) \setminus \{s\}\right| \le opt.$$

Thus,

$$\begin{aligned}\min_{j=1}^{q} p\left(T_j^*\right) &= \min_{j=1}^{q} \frac{\left|I(T_j^*) \setminus \{s\}\right|}{\ell_j}\\ &\le \frac{\sum_{j=1}^{q} \left|I(T_j^*) \setminus \{s\}\right|}{\sum_{j=1}^{q} \ell_j}\\ &\le \frac{opt}{h(B)}.\end{aligned}$$

So, the cheapest one among $T_1^*, T_2^*, \ldots, T_q^*$ has price at most $\frac{opt}{h(B)}$. ■

With the establishment of Lemma 3.4 and Lemma 3.5, we are ready to give the proof for Lemma 3.3. Let T be a cheapest legal arborescence. By Lemma 3.5,

$$p(T) \le \frac{opt}{h(B)}.$$

Let u be the root of T, and ℓ be the number of heads contained in T. By Lemma 3.4,

$$p(T) \ge p(T_\ell(B, u)).$$

Thus,

$$p(T_\ell(B, u)) \le \frac{opt}{h(B)}.$$

Since $T_\ell(B, u) \in \mathcal{T}(B)$, Lemma 3.3 holds.

Finally, we prove Theorem 3.1 by applying Lemma 3.2 and Lemma 3.3. Suppose that the algorithm runs in k iterations. Let $h_0 = n - 1$ which is the number of initial orphan components. For any $1 \le i \le k$, let T_i be the legal arborescence selected at iteration i, and let h_i be the number of orphans just after iteration i. Let $b_i = |I(T_i) \setminus \{s\}|$ and ℓ_i be the number of heads in T_i. Then by Lemma 3.3, for each $1 \le i \le k$,

$$\frac{b_i}{\ell_i} \le \frac{opt}{h_{i-1}}. \quad (3)$$

Since the iteration k is the last iteration, $\ell_k = h_{k-1}$ and consequently,

$$b_k \le opt. \quad (4)$$

By Lemma 3.2, for each $1 \le i \le k-1$,

$$h_{i-1} - h_i \ge 2\ell_i/3. \quad (5)$$

Combining the two inequalities (3) and (3.2), we obtain

$$h_{i-1} - h_i \ge \frac{2}{3}\frac{b_i}{opt} h_{i-1},$$

which implies that

$$\frac{b_i}{1.5opt} \le \frac{h_{i-1} - h_i}{h_{i-1}}.$$

Sum up the above $k - 1$ inequalities, we get

$$\begin{aligned}\frac{1}{1.5opt}\sum_{i=1}^{k-1} b_i &\le \sum_{i=1}^{k-1} \frac{h_{i-1} - h_i}{h_{i-1}}\\ &\le \sum_{i=1}^{k-1} \sum_{j=h_i+1}^{h_{i-1}} \frac{1}{j}\\ &= \sum_{j=h_{k-1}+1}^{h_0} \frac{1}{j}\\ &= H(h_0) - H(h_{k-1})\\ &\le H(n-1) - 1.\end{aligned}$$

Therefore,

$$\sum_{i=1}^{k-1} b_i \leq 1.5\,(H\,(n-1)-1)\cdot opt.$$

Using the inequality (4), we have

$$\sum_{i=1}^{k} b_i \leq 1.5\,(H\,(n-1)-1)\cdot opt + opt$$
$$= (1.5H\,(n-1)-0.5)\cdot opt.$$

Since

$$|B \setminus \{s\}| \leq \sum_{i=1}^{k} c_i,$$

Theorem 3.1 follows.

4. Discussions

By Theorem 2.2, any $\rho \ln n$-approximation algorithm for SAFIN would lead to a $2\rho \ln n$-approximation algorithm for Minimum SCDS. At the expense of higher running time, the $1.5 \ln n$-approximation algorithm for SAFIN presented in this paper can be extended to a $1.35 \ln n$-approximation algorithm for SAFIN by following the approach developed in [9] for Minimum Node Weighted Steiner Tree. On the other hand, it is easy to show that SAFIN is itself as hard as Minimum CDS. Thus, for any constant $0 < \rho < 1$ there is no polynomial-time $\rho \ln n$-approximation for SAFIN unless $NP \subseteq DTIME(n^{o(\ln n)})$. This means that the best possible approximation factor by the approximation algorithm **SCDS** $(\mathcal{A})$ is $2 \ln n + O(1)$. In order to achieve an approximation factor of $\rho \ln n$ with a constant $\rho < 2$, one has to resort to a totally different approach. Any progress towards this improvement would be challenging and exciting.

Acknowledgements

The first author's research was partially supported by the National Natural Science Foundation of China under grant 10671208. The third author's research was partially supported by NSF under grant CNS-0831831. The fourth and sixth authors were supported in part by the National Science Foundation under grants CCF-9208913 and CCF-0728851. The fifth author was supported in part by NSFC (60603003) and XJEDU. This work was done while this author visited University of Texas at Dallas.

References

[1] Khaled M. Alzoubi, Peng-Jun Wan, Ophir Frieder, Message-optimal connected dominating sets in mobile ad hoc networks, in: Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing, Lausanne, Switzerland, June 09–11, 2002.
[2] V. Bharghavan, B. Das, Routing in ad hoc networks using minimum connected dominating sets, in: International Conference on Communication, Montreal, Canada, June 1997.
[3] J. Blum, M. Ding, X. Cheng, Applications of Connected Dominating Sets in Wireless Networks, in: D.-Z. Du, P. Pardalos (Eds.), Handbook of Combinatorial Optimization, Kluwer Academic Publisher, 2004, pp. 329–369.
[4] Yuanzhu Peter Chen, Arthur L. Liestman, Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks, in: Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing, Lausanne, Switzerland, June 09–11, 2002.
[5] M. Cadei, M.X. Cheng, X. Cheng, D.-Z. Du, Connected domination in ad hoc wireless networks, in: Proc. the Sixth International Conference on Computer Science and Informatics, CS&I'2002, 2002.
[6] X. Cheng, X. Huang, D. Li, W. Wu, D.-Z. Du, A polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks, Networks 42 (2003) 202–208.
[7] D.-Z. Du, M.T. Thai, Y. Li, D. Liu, S. Zhu, Strongly connected dominating sets in wireless sensor networks with unidirectional links, in: Proceedings of APWEB, in: LNCS, vol. 3841, 2006, pp. 13–24.
[8] S. Guha, S. Khuller, Approximation algorithms for connected dominating sets, Algorithmica 20 (1998) 374–387.
[9] S. Guha, S. Khuller, Improved methods for approximating node weighted steiner trees and connected dominating sets, Information and Computation 150 (1) (1999) 57–74.
[10] Manki Min, Hongwei Du, Xiaohua Jia, Christina Xiao Huang, Scott C.-H. Huang, Weili Wu, Improving construction for connected dominating set with Steiner tree in wireless sensor networks, Journal of Global Optimization 35 (2006) 111–119.
[11] B. Ramamurthy, J. Iness, B. Mukherjee, Minimizing the number of optical amplifiers needed to support a multi-wavelength optical LAN/MAN, in: Proc. IEEE INFOCOM'97, April 1997, pp. 261–268.
[12] Lu Ruan, Hongwei Du, Xiaohua Jia, Weili Wu, Yingshu Li, Ker-I Ko, A greedy approximation for minimum connected dominating set, Theoretical Computer Science 329 (1–3) (2004) 325–330.
[13] A. Salhieh, J. Weinmann, M. Kochha, L. Schwiebert, Power efficient topologies for wireless sensor networks, in: ICPP'2001, pp. 156–163.
[14] R. Sivakumar, B. Das, V. Bharghavan, An improved spine-based infrastructure for routing in ad hoc networks, in: IEEE Symposium on Computer and Communications, Athens, Greece, June, 1998.
[15] I. Stojmenovic, M. Seddigh, J. Zunic, Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks, in: Proc. IEEE Hawaii Int. Conf. on System Sciences, January 2001.
[16] M.T. Thai, F. Wang, D. Liu, S. Zhu, D.-Z. Du, Connected dominating sets in wireless networks with different transmission ranges, IEEE Transactions on Mobile Computing (TMC) 6 (7) (2007).

[17] M.T. Thai, R. Tiwari, D.-Z. Du, On construction of virtual backbone in wireless ad hoc networks with unidirectional links, IEEE Transactions on Mobile Computing (TMC) 7 (9) (2008) 1098–1109.
[18] P.-J. Wan, K.M. Alzoubi, O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks, in: Proceedings of Infocom'2002.
[19] J. Wu, H.L. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, in: Proceedings of the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, 1999, pp. 7–14.