Approximating Node-Weighted Multicast Trees in Wireless Ad-Hoc Networks

Thomas Erlebach
Department of Computer Science
University of Leicester, UK
te17@mcs.le.ac.uk

Ambreen Shahnaz
Department of Computer Science
University of Leicester, UK
as437@mcs.le.ac.uk

ABSTRACT

Multicast communication in a wireless ad-hoc network can be established using a tree that spans the multicast sender and receivers as well as other intermediate nodes. If the network is modelled as a graph, the multicast tree is a Steiner tree, the multicast sender and receivers correspond to terminals, and other nodes participating in the tree are Steiner nodes. As Steiner nodes are nodes that participate in the multicast tree by forwarding packets but do not benefit from the multicast, it is a natural objective to compute a tree that minimizes the total cost of the Steiner nodes. We therefore consider the problem of computing, for a given node-weighted graph and a set of terminals, a Steiner tree with Steiner nodes of minimum total weight. For graph classes that admit spanning trees of maximum degree at most d, we obtain a $0.775d$-approximation algorithm. We show that this result implies a 3.875-approximation algorithm for unit disk graphs, an $O(1/\alpha^2)$-approximation algorithm for α-unit disk graphs, and an $O(\lambda)$-approximation algorithm for $(\lambda + 1)$-claw-free graphs.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; G.2.2 [**Graph Theory**]: Graph algorithms, Network problems, Trees

General Terms

Algorithms, Theory

Keywords

Wireless ad-hoc networks, multicasting, Steiner tree

1. INTRODUCTION

Wireless ad-hoc networks have received significant attention from researchers in the last few years because of their numerous applications. A wireless ad-hoc network is a multi-hop network without fixed infrastructure. Due to limited resources and the lack of any fixed infrastructure, these networks pose numerous challenges. A simple approach to the study of wireless ad-hoc networks is to model them as unit disk graphs. A unit disk graph is a graph whose nodes correspond to equisized disks in the plane with an edge between two nodes if the corresponding disks intersect [8]. Such a graph is a simplified model of a wireless network consisting of nodes with omnidirectional antennas and equal transmission power. Nodes can communicate directly when they are in each other's transmission range. Since the unit disk graph model is too idealistic, more general graph models that are a better reflection of real wireless ad-hoc networks have been proposed. Kuhn et al. [6] employ *quasi-unit disk graphs* or *α-unit disk graphs*, where nodes are adjacent if their distance is at most α, non-adjacent if their distance is greater than 1, and can be adjacent or non-adjacent otherwise. Schmid and Wattenhofer [10] consider *bounded independence graphs*, i.e. graphs where the independence number of the r-neighborhood of any node is bounded by a polynomial in r. They believe that the model of bounded independence graphs "reflects reality quite well and is appropriate in many situations."

In this paper, we study node-weighted Steiner trees in classes of graphs that admit spanning trees of bounded maximum degree. This includes unit disk graphs, quasi-unit disk graphs and bounded independence graphs. The motivation for our work lies in multicast communication in wireless ad-hoc networks where one node wants to transmit data to multiple receivers via multi-hop transmissions. To support this type of communication, a multicast tree can be used. Considering the graph model of the network, we can refer to the sender and the receivers as *terminals* and a multicast tree corresponds to a *Steiner tree*, i.e., a tree that is a subgraph of the given network and contains all terminals. We are interested in finding a good Steiner tree for a given set of terminals in the graph. In the context of wireless ad-hoc networks, it is meaningful to seek a Steiner tree that minimizes the number of *Steiner nodes*, i.e., the number of nodes that are not terminals but are part of the Steiner tree. The Steiner nodes correspond to wireless nodes that have no direct benefit from taking part in the multicast tree but whose resources are used to forward messages towards the multicast receivers. In other words, these are nodes that are not the intended receivers of a message but only serve as intermediate nodes in transmitting the message to the actual receivers.

We actually consider the weighted version of the problem, where each node of the network has a given weight

IWCMC '09, June 21-24, 2009, Leipzig, Germany.
Copyright © 2009 ACM 978-1-60558-569-7/09/06 ...$5.00.

and the goal is to compute a Steiner tree that minimizes the total weight of the Steiner nodes. This problem variant is motivated by the fact that some nodes may have low battery power or may otherwise be less suitable or willing to forward multicast traffic than others. Such nodes would be assigned higher weight, and a minimum-weight solution would then favor low-weight nodes as Steiner nodes. The unweighted version of the problem, where the goal is to minimize the number of Steiner nodes, is the special case of the weighted version where the terminals have weight 0 and the non-terminals have weight 1.

In this paper, we present a $0.775d$-approximation algorithm for the problem of computing a Steiner tree with minimum total weight of the Steiner nodes for graph classes that admit spanning trees of maximum degree at most d. In particular, this yields a 3.875-approximation algorithm for unit disk graphs, an $O(1/\alpha^2)$-approximation algorithm for α-unit disk graphs, and an $O(\lambda)$-approximation algorithm for $(\lambda + 1)$-claw-free graphs. The latter class of graphs includes the bounded independence graphs.

1.1 Related Work

Many variants of Steiner tree problems have been studied. For the Steiner tree problem in graphs, one distinguishes the edge-weighted version where the goal is to minimize the total weight of the edges of the Steiner tree, and the node-weighted version where the goal is to minimize the total weight of the Steiner nodes. Both versions have also been studied in the unweighted version, where the goal is to minimize the number of edges or the number of Steiner nodes. Note that a Steiner tree has a minimum number of Steiner nodes if and only if it has a minimum number of edges. For the study of approximation algorithms, however, the two objective functions are very different. Researchers have studied the node-weighted Steiner tree problem [5, 3] as well as the edge-weighted Steiner tree problem [7, 4, 11, 14, 9]. So far the best known approximation algorithm for the edge-weighted Steiner tree problem in general graphs is that of Robins and Zelikovsky [9], achieving approximation ratio 1.55. For the node-weighted Steiner tree problem in general graphs, an approximation ratio of $1.35 \ln k$ has been shown by Guha and Khuller [3].

Besides the Steiner tree problem in graphs, there have also been studies of geometric Steiner tree problems. In the Euclidean Steiner tree problem, the terminals are points in Euclidean space, arbitrary points can be used as Steiner nodes, the length of an edge is the Euclidean distance between its endpoints, and the goal is to minimize the total edge length of the tree. The rectilinear Steiner tree problem is the Euclidean Steiner tree problem in the plane with the additional constraint that all edges of the Steiner tree must be horizontal or vertical. It has been proved that the rectilinear Steiner tree problem is NP-hard [2]. This implies that for unit disk graphs, the problem of computing a Steiner tree with a minimum number of edges (or, equivalently, with a minimum number of Steiner nodes) is NP-hard [1].

Xu et al. [13] have considered the Steiner tree problem in unit disk graphs with the goal of minimizing the number of internal nodes of the Steiner tree. The motivation for this objective function is that the internal nodes of the Steiner tree are the nodes that need to actively forward messages, while the leaves are only receivers. They call this problem the *Minimum Steiner Connected Dominating Set* problem and present an algorithm with approximation ratio $2c + 1$, where c is the best known approximation ratio of the edge-weighted Steiner tree problem in graphs, currently $c = 1.55$. A main difference between their objective function and ours is that terminals that are internal nodes of the Steiner tree count towards their objective but not towards ours. Therefore, approximation results for the two problems do not translate to each other. Furthermore, Xu et al. only consider the unweighted version of their problem.

Zou et al. [15] consider the node-weighted Steiner tree problem in unit disk graphs and present a 2.5ρ-approximation algorithm, where ρ is the best known approximation ratio for the edge-weighted Steiner tree problem in arbitrary graphs. With $\rho = 1.55$ [9], this gives approximation ratio 3.875 for node-weighted Steiner trees in unit disk graphs. To obtain this result, Zou et al. define the weight of an edge to be the sum of the weights of the two endpoints and then apply an algorithm for the edge-weighted Steiner tree problem. In this paper we follow the same approach, but our analysis is significantly more general and extends to classes of graphs such as quasi-unit disk graphs and bounded independence graphs.

2. PRELIMINARIES

The Node-Weighted Minimum Steiner Tree (NWMST) problem is defined as follows: Given an undirected graph $G = (V, E)$ with nonnegative weights w_v for $v \in V$ and a subset of nodes $K \subseteq V$ called *terminals*, compute a *Steiner tree* for G and K, i.e., a subgraph T of G that is a tree and contains all the nodes in K. The objective is to minimize the total weight of the vertices of T. We can assume without loss of generality that the terminals have weight 0 (they are present in any solution and their weight increases the objective value of any solution by the same amount), so our goal is to minimize the total weight of the Steiner nodes of T. For a Steiner tree T, we denote its vertex set and edge set by V_T and E_T, respectively. For $v \in V_T$, $\delta_T(v)$ denotes the set of edges in E_T that are incident with v. Furthermore, we denote the set of Steiner nodes of T by $S(T)$ and the total weight of these Steiner nodes (or, equivalently, of all nodes of T) by $w(T)$. Throughout the paper we assume that all leaves of a Steiner tree are terminals, so that all Steiner nodes are internal nodes with degree at least two. (This assumption can be made because Steiner nodes of degree one can simply be removed from a Steiner tree.) If the edges of T are assigned weights according to some weight function c, the total edge weight of T with respect to edge weights c is denoted by $c(T)$.

We say that a class of graphs has *spanning-tree degree bound d* if every connected graph in the class admits a spanning-tree of maximum degree at most d. We first derive our results for classes of graphs with constant spanning-tree degree bound, and then derive results for specific classes of graphs as corollaries. We consider only classes of graphs that are hereditary, i.e., if a graph is in the class, then any induced subgraph of that graph is also in the class.

A graph is a *unit disk graph* if its nodes can be mapped to points in the plane such that two nodes are adjacent if and only if their points have distance at most 1. For $0 < \alpha < 1$, a graph is a *quasi-unit disk graph with parameter α*, or an *α-unit disk graph*, if its nodes can be mapped to points in the plane such that two nodes are (1) adjacent if their points have distance at most α, (2) non-adjacent if their points have distance larger than 1, and (3) can be adjacent or non-adjacent otherwise. A *t-claw* in a graph is an induced subgraph on $t+1$ nodes that is isomorphic to $K_{1,t}$ (the star with one node in the center and t independent leaves). A graph is $(\lambda+1)$*-claw-free* if it does not contain a $(\lambda+1)$-claw or, equivalently, if every node has at most λ independent neighbors. A family of graphs is a family of *bounded independence graphs* if there is a polynomial p such that in any graph in the family, the size of the largest independent set in the r-neighborhood of any node is bounded by $p(r)$, for any r. Note that bounded independence graphs are $(\lambda+1)$-claw-free for $\lambda = p(1)$. Hence, $(\lambda+1)$-claw-free graphs generalize bounded independence graphs.

As NWMST is NP-hard for unit disk graphs and thus also for the more general graph classes we consider, we are interested in algorithms that compute provably good approximate solutions in polynomial time. For $\rho > 1$, an algorithm for NWMST is a *ρ-approximation algorithm* if it runs in polynomial time and always computes a Steiner tree such that the total weight of the vertices is at most ρ times the total weight of the vertices of an optimal Steiner tree.

3. ALGORITHM FOR MINIMIZING THE WEIGHT OF STEINER NODES

3.1 Algorithm Description

Let an instance of NWMST be given by an undirected graph $G = (V, E)$ with node weights w_v for $v \in V$ and a set $K \subseteq V$ of terminals. Our algorithm assigns a nonnegative weight $c(e)$ to each edge e of the graph G. The weight of each edge is set equal to the sum of the weights of the two nodes with which it is incident:

$$\forall (u, v) \in E : \ c(u, v) = w_u + w_v$$

Note that by this definition $c(u, v) = 0$ if $u, v \in K$, since we assume $w_u = 0$ for all $u \in K$.

Our algorithm computes a Steiner tree T for graph G and terminals K with respect to the edge weights c using the algorithm of Robins and Zelikovsky [9]. Then the algorithm outputs the tree T as solution to the given instance of NW-MST.

3.2 Analysis of Approximation Ratio

We can analyze our algorithm as follows. Let T be the Steiner tree computed by the algorithm. Regarding the edge weights c assigned by our algorithm, we view the weight $c(u, v) = w_u + w_v$ of an edge $e = (u, v)$ to be comprised of two parts: one part, denoted $c_u(e) = w_u$, “belongs” to node u, and the other part, denoted $c_v(e) = w_v$, “belongs” to node v. Note that for any edge $e = (u, v)$ we have $c(e) = c_u(e) + c_v(e)$.

Consider an optimal solution to the given instance of NW-MST, i.e. a Steiner tree for graph G and terminal set K in which the total weight of the Steiner nodes is minimum. Denote the weight of the Steiner nodes in this solution by OPT_V. Let us denote the minimum total edge weight (with respect to edge weights c) of a Steiner tree for graph G and terminal set K by OPT_E. As our algorithm computes the Steiner tree T by running the algorithm of Robins and Zelikovsky [9] and then deleting Steiner nodes of degree one, the total edge weight of the Steiner tree T produced by our algorithm satisfies $c(T) \leq 1.55 \cdot OPT_E$. In the next two lemmas we relate OPT_E to OPT_V and $c(T)$ to $w(T)$. This will then allow us to analyze the approximation ratio of our algorithm.

LEMMA 1. *For classes of graphs with spanning-tree degree bound d, $OPT_E \leq d \cdot OPT_V$*

PROOF. Consider a Steiner tree T^* with vertex set V_{T^*} in which the total weight of the Steiner nodes $w(T^*)$ is minimized, i.e., $w(T^*) = OPT_V$. Let G' be the subgraph of G induced by the vertex set V^*. Now consider a spanning tree T' for G' of smallest maximum degree. As the considered class of graphs is hereditary and has spanning-tree degree bound d, T' has maximum degree at most d. This spanning tree T' of G' is a Steiner tree for terminal set K in G with Steiner nodes of degree at most d. As $V_{T^*} = V_{T'}$, we have $w(T^*) = w(T')$. Furthermore, we know that an edge with positive weight is incident to at least one Steiner node. Let us consider a Steiner node v of weight w_v with at most d neighbors. Each edge e incident to v has weight equal to the sum of w_v and the weight of its other endpoint, and the part of the weight that belongs to v is $c_v(e) = w_v$. We have:

$$\sum_{e \in \delta_{T'}(v)} c_v(e) \leq d \cdot w_v$$

This implies:

$$\sum_{v \in S(T')} \sum_{e \in \delta_{T'}(v)} c_v(e) \leq d \cdot \sum_{v \in S(T')} w_v = d \cdot w(T')$$

As the terminals have weight 0 by assumption, the total edge weight of T' is due to the set of Steiner nodes, and we can calculate as follows:

$$\begin{aligned} c(T') &= \sum_{v \in V_{T'}} \sum_{e \in \delta_{T'}(v)} c_v(e) \\ &= \sum_{v \in S(T')} \sum_{e \in \delta_{T'}(v)} c_v(e) \\ &\leq d \cdot w(T') = d \cdot w(T^*) = d \cdot OPT_V \end{aligned}$$

Since $OPT_E \leq c(T')$, we have $OPT_E \leq d \cdot OPT_V$. □

LEMMA 2. $w(T) \leq c(T)/2$

PROOF. The total edge cost $c(T)$ of T is the sum, over all vertices $v \in V_T$, of the sum of $c_v(e)$ over all edges e incident with v:

$$c(T) = \sum_{v \in V_T} \sum_{e \in \delta_T(v)} c_v(e)$$

Since $c_v(e) = 0$ for $v \in K$, we have:

$$c(T) = \sum_{v \in S(T)} \sum_{e \in \delta_T(v)} c_v(e)$$

Furthermore, every Steiner node v of T has at least two incident edges, because all leaves of T are terminals. As $c_v(e) = w_v$ for every edge e incident with a Steiner node v, we have:

$$\sum_{e \in \delta_T(v)} c_v(e) \geq 2 \cdot w_v$$

This yields

$$2 \cdot w(T) = \sum_{v \in S(T)} (2 \cdot w_v) \leq \sum_{v \in S(T)} \sum_{e \in \delta_T(v)} c_v(e) = c(T),$$

and the lemma follows. □

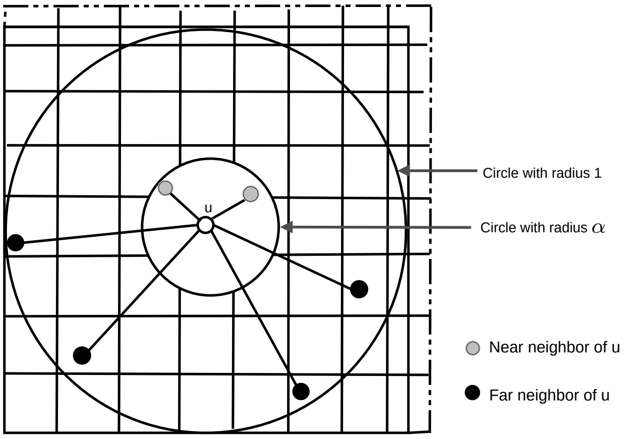

Figure 1: Neighbors of a node in an α-unit disk graph

Theorem 1. *For graph classes with spanning-tree degree bound d, there is a $0.775d$-approximation algorithm for the Node-Weighted Minimum Steiner Tree problem.*

Proof. As the algorithm by Robins and Zelikovsky is a 1.55-approximation algorithm for the edge-weighted Steiner tree problem [9], the total edge cost of the tree T computed by our algorithm is $c(T) \leq 1.55 \cdot OPT_E$. By Lemma 1, we have $OPT_E \leq d \cdot OPT_V$, and by Lemma 2, we have $w(T) \leq c(T)/2$. Combining these inequalities, we get $w(T) \leq (1.55 \cdot d/2) \cdot OPT_V = 0.775d \cdot OPT_V$. □

3.3 NWMST in Unit Disk Graphs and Other Graph Classes

In this section, we apply Theorem 1 to several classes of graphs that are frequently employed to model wireless ad-hoc networks. First, we obtain directly the result for unit disk graphs that was presented by Zou et al. in [15].

Corollary 1. (Zou et al. [15]) There is a 3.875-approximation algorithm for the Node-Weighted Minimum Steiner Tree problem in unit disk graphs.

Proof. As shown by Wu et al. [12], any connected unit disk graph admits a spanning tree in which the degree of every vertex is at most five. By applying Theorem 1 with $d = 5$, we obtain approximation ratio $0.775 \cdot 5 = 3.875$. □

Now we consider quasi-unit disk graphs with parameter α.

Lemma 3. *For every $0 < \alpha < 1$, the class of α-unit disk graphs has spanning-tree degree bound at most $6 + 8/\alpha^2 + 4\sqrt{2}/\alpha$.*

Proof. Consider any connected α-unit disk graph $G = (V, E)$. Let the weight w_{uv} of an edge $uv \in E$ be the Euclidean distance between the positions of nodes u and v, and let T be a minimum spanning tree of G with respect to these edge weights. Consider an arbitrary node u of G. The degree of node u in T can be calculated as the number of neighbors that are at distance at most α from u plus the number of neighbors that have distance greater than α from u. We call the former type of neighbors the *near* neighbors, the latter type the *far* neighbors. We bound the number of neighbors of each type separately. First, consider the near neighbors of u in T. We can use the method of [12] to prove that T can be chosen so that there are at most 5 such neighbors.

Now consider the far neighbors of node u in T. All such neighbors lie in a ring around u that is bounded by circles of radius α and 1 with center u, see Fig. 1. This ring is contained in a square with side length 2 and center u. We cover this square using square *cells* of side length $\alpha/\sqrt{2}$ (and, thus, diameter α). We claim that each cell can contain at most one far neighbor of u in T. If a cell contained two far neighbors v and w of u in T, the weight of T could be decreased by removing one of the edges uv and uw (both of which have length more than α), and adding the edge vw (which has length at most α). Hence, the number of far neighbors of u is bounded by the number of cells needed to cover the ring, which is in turn bounded by the number of cells needed to cover a square of area 2×2. This number of cells is at most $\left\lceil \frac{2}{\frac{\alpha}{\sqrt{2}}} \right\rceil \cdot \left\lceil \frac{2}{\frac{\alpha}{\sqrt{2}}} \right\rceil \leq (\frac{2\sqrt{2}}{\alpha} + 1) \cdot (\frac{2\sqrt{2}}{\alpha} + 1) = \frac{8}{\alpha^2} + \frac{4\sqrt{2}}{\alpha} + 1$.

Considering both types of neighbors of u, we get that u has degree at most $5 + 8/\alpha^2 + 4\sqrt{2}/\alpha + 1 = 6 + 8/\alpha^2 + 4\sqrt{2}/\alpha$ in T. As the argument can be applied to any node u, the claim follows. □

Corollary 2. *For α-unit disk graphs, there is a $0.775 \cdot (6 + 8/\alpha^2 + 4\sqrt{2}/\alpha)$-approximation algorithm for the NW-MST problem.*

We remark that this result also applies to disk graphs with bounded diameter ratio. A disk graph is a graph where each node corresponds to a disk in the plane and two nodes are adjacent if the corresponding disks have a non-empty intersection. The diameter ratio of a disk graph is the ratio between the largest and the smallest diameter of any disk in the graph. If a disk graph has diameter ratio at most D for some constant D, we can assume without loss of generality that the largest disk has diameter 1 and the smallest disk has diameter at least $1/D$. It is easy to see that the graph is then an α-unit disk graph with $\alpha = 1/D$. Hence, Lemma 3 and Corollary 2 imply that disk graphs with diameter ratio at most D have spanning-tree degree bound $O(D^2)$ and admit an $O(D^2)$-approximation algorithm for the NWMST problem.

Next, we consider the class of $(\lambda+1)$-claw-free graphs, for some constant $\lambda > 1$. Recall that bounded independence graphs are a subset of this class of graphs.

Lemma 4. *For every integer $\lambda > 1$, the class of $(\lambda + 1)$-claw-free graphs has spanning-tree degree bound at most $\lambda + 1$.*

Proof. Let $G = (V, E)$ be any connected, $(\lambda + 1)$-claw-free graph. We give an algorithm that constructs a spanning tree of G with maximum degree at most $\lambda + 1$. The algorithm grows a subtree T of G, initially consisting of a single node, into a spanning tree by repeatedly picking a leaf node v and adding a maximal independent set of its neighbors outside T as children of v. More formally, the algorithm can be described as follows:

1. Let T be the subtree of G consisting of an arbitrary node u of G.

2. **while** T is not yet a spanning tree **do**

```
    (i) Pick any leaf $v$ of $T$ that has at least one neighbor
        outside $T$.
    (ii) Compute a maximal independent set $I$ among the
         neighbors of $v$ that are not in $T$.
    (iii) Add all nodes of $I$ as children of $v$ to $T$.
3. return $T$.
```

First, we show that the algorithm computes a spanning tree. The algorithm could only fail to compute a spanning tree if it reaches a situation in which T is not yet a spanning tree, but in Step 2(i) there is no leaf of T with a neighbor outside T. Assume that this happens. Let U be the set of nodes in T and W the set of nodes outside T. Since G is connected, there must be a node $u \in U$ and a node $w \in W$ such that $uw \in E$. The node u cannot be a leaf of T, because we assume that no leaf has an edge to a node in W. Hence, u is an internal node of T. We can assume that u is chosen to be the deepest node in T that is adjacent to w. In an earlier step of the algorithm, when the current tree was a subtree T' of T and u was a leaf of T', the algorithm has added a maximal independent set I of u's neighbors as children of u to the tree. As w was not added to I, w must be adjacent to a node in I, and hence w must be adjacent to a child of w. This is impossible because we have assumed that u is the deepest node in T that is adjacent to w. We have reached a contradiction, and hence the situation where the algorithm fails to compute a spanning tree cannot occur.

Now we analyze the degree of the constructed spanning tree. As G is $(\lambda + 1)$-claw-free, the independent set I computed in Step 2(ii) has cardinality at most λ. Therefore, every node of T has at most λ children and at most one parent. Consequently, the degree of every node of the tree is at most $\lambda + 1$. □

Corollary 3. *There is a $0.775 \cdot (\lambda+1)$-approximation algorithm for the NWMST problem in $(\lambda+1)$-claw-free graphs.*

4. CONCLUSION

Motivated by the problem of computing good multicast trees in wireless ad-hoc networks, we have considered the node-weighted Steiner tree problem in several graph models of wireless networks. Minimizing the weight of the Steiner nodes is meaningful in this setting because it is desirable to minimize the cost of the nodes that forward multicast packets but do not benefit from the multicast transmission themselves. Furthermore, by allowing arbitrary cost values we can model the situation where the nodes have various degrees of willingness or capability to take part in a multicast tree.

We have shown that the simple approach of defining suitable edge weights and then applying an approximation algorithm for the edge-weighted Steiner tree problem yields a good approximation ratio for the node-weighted Steiner tree problem in graph classes that admit spanning trees of small degree. This rather general result has allowed us to obtain algorithms with constant approximation ratio for popular graph models of wireless ad-hoc networks that go well beyond simple unit disk graphs, namely quasi-unit disk graphs and bounded independence graphs (or, even more generally, $(\lambda + 1)$-claw-free graphs).

Our algorithm is centralized, but any distributed algorithm for the edge-weighted Steiner tree problem can be used to obtain a distributed implementation of our algorithm. If the approximation ratio of the distributed algorithm for the edge-weighted Steiner tree problem is ρ, our algorithm yields approximation ratio $\frac{\rho d}{2}$ for graphs with spanning-tree degree bound d.

One interesting question for future work is whether the approximation ratio for the node-weighted Steiner tree problem in unit disk graphs can be improved further and whether even a polynomial-time approximation scheme exists for the problem.

5. REFERENCES

[1] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165–177, 1990.

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

[3] S. Guha and S. Khuller. Improved methods for approximating node weighted Steiner trees and connected dominating sets. *Information and Computation*, 150(1):57–74, 1999.

[4] M. Karpinsky and A. Zelikovsky. New approximation algorithms for the Steiner tree problem. *Journal of Combinatorial Optimization*, 1:47–66, 1997.

[5] P. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted Steiner trees. *Journal of Algorithms*, 19(1):104–115, 1995.

[6] F. Kuhn, R. Wattenhofer, and A. Zollinger. Ad-hoc networks beyond unit disk graphs. *Wireless Networks*, 14(5):715–729, 2008.

[7] G. M. L. Kou and L. Berman. A fast algorithm for Steiner trees. *Acta Informatica*, 15(2):141–145, 1981.

[8] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

[9] G. Robins and A. Zelikovsky. Improved Steiner tree approximation in graphs. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'00)*, pages 770–779, 2000.

[10] S. Schmid and R. Wattenhofer. Algorithmic models for sensor networks. In *Proceedings of the 20th International Parallel and Distributed Processing Symposium (IPDPS)*, 2006.

[11] H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Mathematica Japonica*, 24:573–577, 1980.

[12] W. Wu, H. Du, X. Jia, Y. Li, and S. C.-H. Huang. Minimum connected dominating sets and maximal independent sets in unit disk graphs. *Theoretical Computer Science*, 352(1):1–7, 2006.

[13] C. Xu, Y. Xu, and J. Wu. On the minimization of the number of forwarding nodes for multicast in wireless ad hoc networks. In *Proceedings of ICCNMC'05*, LNCS 3619, pages 286–294. Springer, 2005.

[14] A. Zelikovsky. An 11/6 approximation algorithm for the network Steiner problem. *Algorithmica*, 9:463–470, 1993.

[15] F. Zou, X. Li, D. Kim, and W. Wu. Two constant approximation algorithms for node-weighted Steiner tree in unit disk graphs. In *Proceedings of COCOA'08*, LNCS 5165, pages 278–285. Springer, 2008.