

Approximating Fault-Tolerant Steiner Subgraphs in Heterogeneous Wireless Networks

Ambreen Shahnaz and Thomas Erlebach

Department of Computer Science

University of Leicester

University Road, Leicester LE1 7RH, United Kingdom

Email: {as437|te17}@mcs.le.ac.uk

Abstract—If a set K of nodes in a wireless network want to set up a routing structure that allows them to communicate with each other, one possible approach is to use a Steiner tree that spans all the nodes in K . However, a tree can be disconnected by the failure of a single link, and so it is desirable to employ other routing structures that are fault-tolerant. Furthermore, many real-world wireless networks are heterogeneous, meaning that the suitability of nodes for inclusion in the routing structure varies significantly. Therefore, it is meaningful to assign weights to the nodes and aim to compute a fault-tolerant routing structure of minimum total weight. In this paper, we model this problem as the problem of computing a minimum-weight 2-edge-connected Steiner subgraph spanning a given set of terminals, and we propose a constant-factor approximation algorithm for this problem in wireless networks that are modelled as unit disk graphs or quasi unit disk graphs.

I. INTRODUCTION

There are many settings where a group of nodes in a network want to set up a routing structure that allows efficient communication within the group, for example, multicast communication or a virtual private network. The simplest routing structure that can connect a set of nodes is a tree spanning those nodes, also called a *Steiner tree*. However, in scenarios where links are not fully reliable and may fail, a tree structure has the disadvantage that it can be disconnected by a single link failure. Therefore, it is desirable to establish routing structures that have some amount of fault-tolerance. We refer to subgraphs that span a given set of nodes and that are not necessarily trees as *Steiner subgraphs*. We consider the property of being 2-edge-connected as the criterion for fault tolerance: A 2-edge-connected graph remains connected even if an arbitrary edge (link) fails. Hence, we are interested in determining 2-edge-connected Steiner subgraphs in wireless networks.

Wireless networks are often heterogeneous, i.e., the nodes can have different amounts of computing resources, packet buffer size, remaining battery life, etc. Therefore, the suitability for inclusion in a routing subgraph can vary widely between different nodes. We model this by assuming that each node of the network is assigned a weight that represents its cost for inclusion in the Steiner subgraph: Nodes that are more suitable have smaller weight, and the goal is to compute a Steiner subgraph with the desired fault tolerance property of minimum total weight.

We consider unit disk graphs (UDG) and quasi unit disk graphs (quasi-UDG) as families of graphs representing wireless networks. Therefore, we study the problem of determining minimum-weight 2-edge-connected Steiner subgraphs in UDG and quasi-UDG. We propose a constant-factor approximation algorithm for this problem.

A. Definitions

The node-weighted 2-edge-connected Steiner subgraph problem (NW2ECS) is defined as follows. We are given an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. A nonnegative weight w_v is assigned to each node $v \in V$. We are also given a set of *terminals* $K \subseteq V$. A subgraph $G' = (V', E')$ of G is a *2-edge-connected Steiner subgraph* if it contains all terminals, i.e., $K \subseteq V'$, and there are at least two edge-disjoint paths in G' between any pair of terminals. The objective of NW2ECS is to find a 2-edge-connected Steiner subgraph G' such that the total weight of the nodes in G' , denoted by $w(G')$, is minimized.

Nodes in G' that are not terminals are called *Steiner nodes*. We can assume that the terminals have weight 0 as they are present in any solution and their weight increases the objective value of any solution by the same amount. Therefore, minimizing the total weight of the Steiner subgraph means minimizing the total weight of the Steiner nodes in G' .

We remark that it is easy to check whether a given instance of NW2ECS with graph $G = (V, E)$ and set K of terminals admits a feasible solution. For this, one only needs to check whether there are at least two edge-disjoint paths in G between any pair of vertices in K . This can be done in polynomial time using standard network flow techniques (see, e.g., [1]). Therefore, we assume in the rest of the paper that the given instance of NW2ECS admits at least one feasible solution.

A graph $G = (V, E)$ is a *unit disk graph* (UDG) [2] if each vertex corresponds to a point in the Euclidean plane and there is an edge between two vertices if and only if the Euclidean distance between the corresponding points is at most one. In other words, two vertices are adjacent if unit disks (i.e., topologically closed disks with diameter 1) centered at the corresponding points intersect. UDG are a simplified theoretical model for wireless networks.

For $0 < \alpha \leq 1$, a graph $G = (V, E)$ is an α -*unit disk graph* [3] if each vertex corresponds to a point in the Euclidean plane

and the following holds for every pair u, v of vertices: If the points corresponding to u and v have distance at most α , then there is an edge between u and v . If the points have distance larger than 1, then there is no edge between u and v . If the points have distance larger than α but less than or equal to 1, then the graph may or may not contain an edge between u and v . Graphs that are α -unit disk graphs for some constant α are also called *quasi unit disk graphs*. Quasi unit disk graphs are considered a more realistic model for wireless networks than unit disk graphs [3].

An algorithm for an optimization problem is called a ρ -approximation algorithm, or has approximation ratio ρ , if it runs in polynomial time and always outputs a feasible solution (if there is one) with an objective value that is at most a factor of ρ away from the optimal objective value for the given input. Here, $\rho \geq 1$, and the closer ρ is to 1, the better.

B. Related Work

NW2ECS is related to the classical combinatorial problem known as survivable network design problem or generalized Steiner network problem, defined as follows: Let an undirected graph $G = (V, E)$ with vertex set V and edge set E be given. For every edge $e \in E$, there is a nonnegative weight c_e , and for every pair i, j of vertices in V , there is a connectivity requirement $r_{i,j}$. The objective is to find a minimum-cost subgraph such that there are at least $r_{i,j}$ edge-disjoint paths between i and j , for all $i, j \in V$ [4]. (There are also problem variants concerned with node-disjoint instead of edge-disjoint paths.) The edge-weighted version of the 2-edge-connected Steiner subgraph problem is a special case of the generalized Steiner network problem: $r_{i,j}$ equals 2 if $i, j \in K$ and 0 otherwise. A 2-approximation algorithm for the generalized Steiner network problem was presented by Jain [5]. Hence, there is a 2-approximation algorithm for the edge-weighted version of the 2-edge-connected Steiner subgraph problem for arbitrary graphs, but no known constant-factor approximation algorithm for the node-weighted version.

A geometric version of the survivable network design problem has been studied by Czumaj et al. [6]. Their work yields a polynomial time approximation scheme for both vertex-disjoint and edge-disjoint versions in Euclidean graphs.

All of this previous research work is concerned with edge costs. In wireless networks, it is more meaningful to associate costs with the nodes, and therefore we consider the node-weighted version of the problem.

For the edge-weighted Steiner tree problem, the best known approximation algorithm for arbitrary graphs is due to Robins and Zelikovsky and achieves approximation ratio approximately 1.55 [7], whereas for the node-weighted Steiner tree problem the best known approximation ratio is logarithmic in the number of terminals [8]. Therefore, the node-weighted Steiner tree problem has been studied for restricted graph classes that model wireless networks. For UDG a 3.875-approximation algorithm was presented by Zou et al. [9], and for quasi unit disk graphs an $O(1/\alpha^2)$ -approximation algorithm was presented in [10]. These algorithms work by

assigning suitable edge weights to the given node-weighted graph and then executing an algorithm for the edge-weighted problem. The analysis of the algorithm then relates the node cost of the computed solution to the optimal node cost via the optimal edge cost as a reference value. In this paper, we extend this approach to the case of 2-edge-connected Steiner subgraphs.

Dai and Wu studied fault tolerance in the construction of backbones in a wireless ad hoc network modelled by a unit disk graph [11]. They proposed localized algorithms for backbone construction via a k -connected k -dominating set. According to their definition, a k -connected k -dominating set for a given graph $G = (V, E)$ has the following properties: For every node not in the dominating set, there are k neighbors in the dominating set, and any two nodes in the dominating set are connected by k edge-disjoint paths using only nodes of the dominating set.

C. Our Results

In this paper, we present a constant-factor approximation algorithm for NW2ECS in UDG and in quasi-UDG. We prove the approximation ratio to be at most 12 for UDG and $O(1/\alpha^2)$ for α -unit disk graphs. Our algorithm uses the given node weights to define suitable edge weights and then applies a known 2-approximation algorithm for the edge-weighted 2-edge-connected Steiner subgraph problem. The main ingredient of the analysis is a proof showing that in UDG and quasi-UDG, there always exists an optimal solution of small degree. This allows us to derive approximation results for the node-weighted version of the problem from a known approximation algorithm for the edge-weighted version.

II. COMPUTING 2-EDGE-CONNECTED STEINER SUBGRAPHS

A. Algorithm Description

Consider an instance of NW2ECS 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. As noted earlier, we assume that $w_v = 0$ for all $v \in K$. We assign 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:

$$\forall uv \in E : c(uv) = w_u + w_v$$

Note that $c(u, v) = 0$ if $u, v \in K$. Our algorithm computes a Steiner subgraph G' for graph G and terminals K with respect to edge weights c using the 2-approximation algorithm by Jain [5] for the edge-weighted 2-edge-connected Steiner subgraph problem. Then, it outputs G' as the solution for the given instance of NW2ECS.

Note that we can assume that G' does not contain any Steiner nodes of degree less than 2, because such vertices could be removed from the solution while reducing the cost and maintaining feasibility.

B. Analysis of Approximation Ratio

First, we analyze the algorithm for arbitrary graph classes that have the following property, for some constant d :

- (P_d) Any 2-edge-connected graph in the class contains a spanning 2-edge-connected subgraph of maximum vertex degree d .

In subsequent sections, we will then show that UDG and quasi-UDG satisfy this property.

Assume that the graph $G = (V, E)$ of the given instance of NW2ECS belongs to a class of graphs with property (P_d). Consider an optimal solution to this instance. The weight of the Steiner nodes in this solution is denoted by OPT_V . Let us denote the minimum total edge weight (with respect to edge weights c) of a 2-edge-connected Steiner subgraph for graph G and terminal set K by OPT_E . As our algorithm computes the Steiner subgraph G' by running the 2-approximation algorithm of Jain [5] for the edge-weighted version of the problem, the total edge weight of G' , denoted by $c(G')$, thus satisfies $c(G') \leq 2 \cdot OPT_E$. We analyze the approximation ratio by relating OPT_E to OPT_V and $c(G')$ to $w(G')$, using an adaptation of analogous results for the Steiner tree problem [9], [10].

Lemma 1: For classes of graphs satisfying (P_d), it holds that $OPT_E \leq d \cdot OPT_V$

Proof: Let G^* be the optimal solution to the node-weighted problem. By property (P_d), G^* contains a spanning 2-edge-connected subgraph G_d of maximum node degree d . The total node weight of G_d is equal to that of G^* , as the set of nodes is the same. The total edge weight of G_d is at most $d \cdot OPT_V$, because the weight of each edge is the sum of the weights of its two end vertices and each vertex contributes to the weight of at most d edges. Hence, G_d is a feasible solution of edge weight at most $d \cdot OPT_V$, and the optimal solution with respect to edge weights cannot have larger weight. ■

Lemma 2: $w(G') \leq c(G')/2$

Proof: As every Steiner node of G' has degree at least two in G' and $c(uv) = w_u + w_v$ for all edges uv , we have that every Steiner node contributes its weight once to the sum of all node weights but at least twice to the sum of all edge weights. Hence, $c(G') \geq 2w(G')$. ■

Theorem 1: For classes of graphs satisfying (P_d), the algorithm described in Section II-A is a d -approximation algorithm for NW2ECS.

Proof: Combining the previous lemmas, we get that $w(G') \leq c(G')/2 \leq OPT_E \leq d \cdot OPT_V$. ■

Before we consider UDG and quasi-UDG, we state the following auxiliary lemma that follows directly from Menger's theorem.

Lemma 3: If there are at least two edge-disjoint paths between u and v , and at least two edge-disjoint paths between v and w , then there are also at least two edge-disjoint paths between u and w .

Proof: Recall that by Menger's theorem (see, e.g., [1]), for any pair of distinct nodes x and y , the size (number of edges) of the minimum cut between x and y is equal to the maximum number of edge-disjoint paths between x and y .

Consider any cut C that separates u and w . The node v is connected to each of the nodes u and w via two edge-disjoint paths.

- Case 1: C separates u and v , i.e., v is located on the side of C where w is located. We know that there are at least two edge-disjoint paths between u and v . So, there are at least two edges going across C , which means that the size of C is at least 2.
- Case 2: v is on the side of C where u lies. There are two disjoint paths between v and w , and C is a cut separating these two nodes. Again, the size of C is at least 2.

In either case, there are at least two edges in the cut C . As the choice of C was arbitrary, we have that any cut that separates u and w has size at least 2. By Menger's theorem, this implies that there are at least 2-edge-disjoint paths between u and w . ■

If there are two edge-disjoint paths between two nodes u and v , we also say that u and v are in a cycle.

C. Unit Disk Graphs

Lemma 4: Every 2-edge-connected unit disk graph has a 2-edge connected spanning subgraph of maximum degree at most 12.

Proof: Let $G = (V, E)$ be a 2-edge-connected unit disk graph. We show how to construct a 2-edge-connected spanning subgraph G' of small degree. First, we determine a minimum spanning tree T (with respect to edge weights that represent the Euclidean distance between the endpoints of the edge). Then, we add edges to T so that it becomes 2-edge-connected. The construction can be stated more formally as follows:

- (i) Compute a minimum spanning tree T with respect to edge weights given by Euclidean distance, and initialize $G' = T$.
- (ii) For all edges $e \in E$ that are not in T , in order of non-decreasing length:
 - if $G' \cup \{e\}$ puts at least one bridge edge from G' into a cycle, then add e to G' .

Here, an edge of G' is a bridge edge (or simply a bridge) if the removal of the edge disconnects G' . An edge is a bridge if and only if it is not contained in any cycle. Putting a bridge edge into a cycle means turning the edge from a bridge edge into an edge that is not a bridge.

We observe that after the construction has considered an edge uv in the for-loop, either it has added the edge or there were already two edge-disjoint paths between u and v in G' . In either case, we have that after an edge uv has been processed, u and v are in a cycle.

It is easy to see that if G is 2-edge-connected, the graph G' produced by the construction is 2-edge-connected. It remains to analyze the maximum node degree of the final G' . Consider an arbitrary node u . We divide the surrounding area of u into six equal 60° sectors. We choose the sectors in such a way that none of the neighbors of u in G lies on the boundary between two sectors. Note that each sector contains at most one node that is a neighbor of u in the spanning tree T . (If

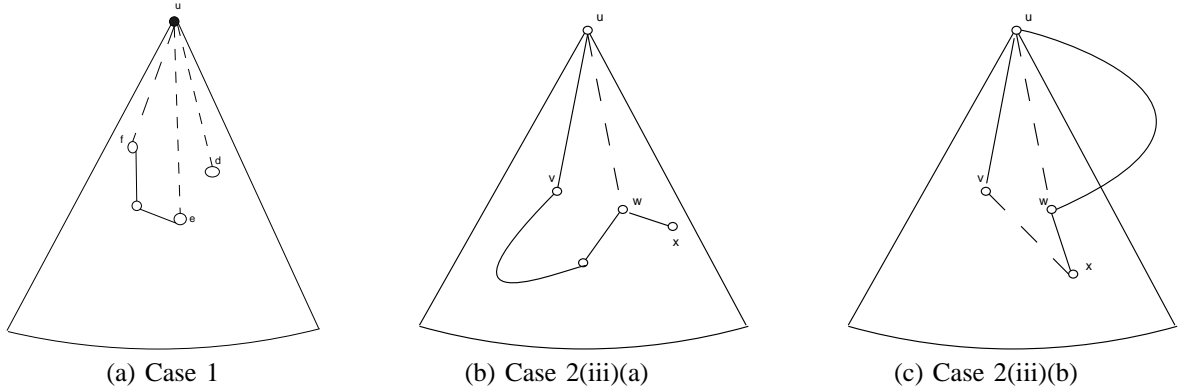


Fig. 1. Illustration of cases in the proof of Lemma 4. Edges in T are drawn solid, other edges dashed.

there were two neighbors of u in the same sector, one of the two edges joining u and these neighbors could be replaced by the shorter edge between these two neighbors to give a spanning tree of smaller cost, a contradiction.) The degree of u in the final G' is the sum of the number of adjacent nodes of u in all six sectors. We claim that u has at most 2 neighbors in each sector in G' . To prove this, consider the following cases.

Case 1: Consider a 60° sector of u that does not contain a spanning tree neighbor of u . Assume that there are two nodes f and d in the sector such that the construction has added edges between u and these two nodes to G' . See Figure 1(a). Let e be a third node in the sector that the construction considers. If ef and ed are in T , then $ufedu$ is a cycle in G' containing e and u , so eu is not added.

Now assume that at least one of the two edges (ef and ed) is not in T . Let us assume ef is not in T . (The case that ed is not in T is analogous.) The construction adds edges in order of increasing length, so ef is considered before eu . This means that e and f are already in a cycle when eu is considered. Furthermore, f and u are in a cycle because the construction has added the edge fu and there is also a path connecting f and u in T . By Lemma 3, if u and f are in a cycle and f and e are in a cycle, then u and e are also in a cycle. Therefore, the construction does not add e as third neighbor of u .

Case 2: u has one neighbor v in the sector such that $uv \in T$. Assume that the construction has added w as a second neighbor of u in the sector. Note that the distance from u to w is at least as large as the distance from u to v , because otherwise the edge vu would not be in any minimum spanning tree. (It would be the longest edge in the cycle $uvwu$.) Assume x is another node in the sector and the construction considers the edge xu .

- (i) If xv and xw are in T : Then $xvwvx$ is a cycle. So, the construction does not add xu .
- (ii) If xw is not in T (and xv may or may not be in T): Before considering xu , the construction has considered xw . From then on, u and w are in a cycle. Furthermore, w and x are in a cycle. Therefore, by Lemma 3, u and x are in a cycle, and the edge xu is not added by the construction.

- (iii) If xw is in T and xv is not in T : There can be two different ways in which w is connected to u, v via a path in T . (Furthermore, if x lies on that path then x and u are in a cycle and xu does not get added to G' by the construction. Hence, we assume in the following that x does not lie on the path in T between w and u, v .)

- (a) The path in T from w to u, v reaches v before u . See Figure 1(b). Observe that u and v are in a cycle as uw has been added to G' . Moreover, the construction considers vx before ux is considered. Therefore, v and x are already in a cycle when the construction considers ux . By Lemma 3, if u and v are in a cycle and v and x are in a cycle, then u and x are also in a cycle. Thus, the construction does not add xu .
- (b) The path in T from w to u, v reaches u before v . See Figure 1(c). vx is considered before ux . If the construction adds vx , then u and x lie on the cycle consisting of the edge vx and the path in T between v and x , so ux does not get added. If vx is not added, this shows that uw is not a bridge. (If uw was a bridge, the addition of vx would have put it into a cycle, and hence the construction would have added vx .) Hence, v and x are in a cycle, and u and v are in a cycle. By Lemma 3, u and x are in a cycle, and the construction does not add ux .

We have shown that in each of the six sectors, it is impossible that the construction adds an edge to a node that would become a third neighbor to u in that sector. Therefore, the final G' is a 2-edge-connected spanning subgraph in which each node has at most 12 neighbors. \blacksquare

By applying Theorem 1 with $d = 12$, we obtain the following corollary.

Corollary 1: There is a 12-approximation algorithm for NW2ECS in unit disk graphs.

D. α -Unit Disk Graphs

Lemma 5: For every $0 < \alpha < 1$, every 2-edge-connected α -UDG has a 2-edge-connected spanning subgraph of node degree at most $14 + 16/\alpha^2 + 8\sqrt{2}/\alpha$.

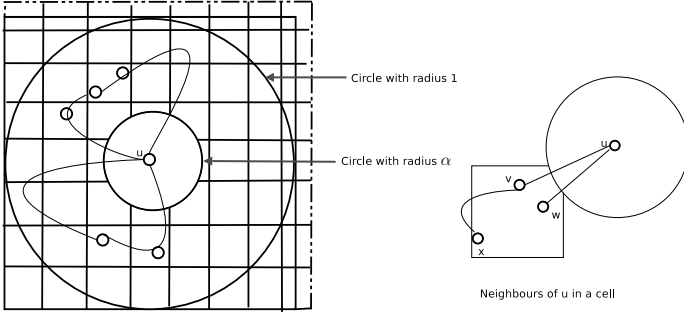


Fig. 2. Neighbors of u in an α -unit disk graph

Proof: Apply the same construction as in the proof of Lemma 4 to produce a 2-edge-connected spanning subgraph G' . To analyze the neighborhood of a node in an α -UDG, we use the same framework as in [10]. The neighbors of a node u are classified as *near* (within distance at most α from u) and *far* (distance greater than α) neighbors. The proof of Lemma 4 can be used to show that u has at most 12 near neighbors in the final G' . In the remainder of the proof, we consider only far neighbors. To bound the number of far neighbors, we cover a 2×2 square containing all neighbors of u using squares of side length $\alpha/\sqrt{2}$ called *cells*, see Figure 2. Note that any two nodes in the same cell have distance at most α and hence are adjacent. We show that the number of far neighbors of u in the final G' is at most twice the number of cells. After computing a minimum spanning tree T , there is at most one neighbor of u in every cell. We show that if a cell contains a neighbor of u in T , then at most one more node in that cell can be added as a neighbor of u by our construction, and otherwise, at most two neighbors can be added to u .

Consider three nodes v, w, x in a cell and assume that in the minimum spanning tree, there is no neighbor of u in this cell. Assume that v and w have already been added as neighbors of u in this cell and we are now considering the edge ux . We claim that the construction would not add x as third neighbor of u . To show this, we can use arguments analogous to Case 1 of the proof of Lemma 4: If vx and wx are both in T , then we have a cycle $uvxwu$ and therefore, xu does not get added to T . If at least one of vx and wx is not in T , assume that vx is not in T . Then the algorithm considers vx before xu is considered, so by the time xu is considered v and x are already in a cycle. By Lemma 3, if u and v are in a cycle and v and x are in a cycle, then u and x are also in a cycle. Therefore, the construction does not add x as third neighbor of u .

Now consider the case that u has one neighbor v' in T in the cell, a second node w' has been added as neighbor of u in this cell, and we are now considering the edge ux' for a third node x' in this cell. Here, the same arguments apply as we have described in Case 2 of the proof of Lemma 4, and it again follows that ux' is not added.

Thus, we get that in every cell, u has at most two far neighbors in the end. Therefore, the number of far neighbors

of u is at most twice the number of cells, and the number of cells is at most

$$\left\lceil \frac{2}{\frac{\alpha}{\sqrt{2}}} \right\rceil \cdot \left\lceil \frac{2}{\frac{\alpha}{\sqrt{2}}} \right\rceil \leq \left(\frac{2\sqrt{2}}{\alpha} + 1 \right) \cdot \left(\frac{2\sqrt{2}}{\alpha} + 1 \right) = \frac{8}{\alpha^2} + \frac{4\sqrt{2}}{\alpha} + 1.$$

The total number of *near* and *far* neighbors of u is at most $12 + 16/\alpha^2 + 8\sqrt{2}/\alpha + 2 = 14 + 16/\alpha^2 + 8\sqrt{2}/\alpha$. ■

By applying Theorem 1 with $d = 14 + 16/\alpha^2 + 8\sqrt{2}/\alpha$ we obtain the following corollary.

Corollary 2: There is an approximation algorithm with approximation ratio $14 + 16/\alpha^2 + 8\sqrt{2}/\alpha = O(1/\alpha^2)$ for NW2ECS in α -unit disk graphs.

III. CONCLUSION

Motivated by the computation of fault-tolerant routing structures in wireless networks, we have considered the node-weighted 2-edge-connected Steiner subgraph problem and presented a constant-factor approximation algorithm for unit disk graphs and quasi unit disk graphs. The key in the analysis was to show that for the considered classes of graphs, there exists an optimal solution of small node degree.

For the node-weighted Steiner tree problem, constant-factor approximation algorithms have also been found for the class of $(\lambda + 1)$ -claw-free graphs, for constant λ [10]. These are graphs in which each node has at most λ neighbors that are pairwise non-adjacent. In the future we plan to extend our approximation results for NW2ECS to that class of graphs as well. Furthermore, it would be interesting to study other variants of fault tolerance requirements, e.g., k -edge-connected Steiner subgraphs for $k > 2$ or k -vertex-connected Steiner subgraphs.

REFERENCES

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. New Jersey: Prentice Hall, 1993.
- [2] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, no. 1-3, pp. 165-177, 1990.
- [3] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Ad-hoc networks beyond unit disk graphs," *Wireless Networks*, vol. 14, no. 5, pp. 715-729, 2008.
- [4] M. Grötschel, C. L. Monma, and M. Stoer, "Design of survivable networks," *Handbook in Operations Research and Management Science*, pp. 617-671, 1995.
- [5] K. Jain, "A factor 2 approximation algorithm for the generalized steiner network problem," *Combinatorica*, vol. 21, no. 1, pp. 39-60, 2001.
- [6] A. Czumaj, A. Lingas, and H. Zhao, "Polynomial-time approximation schemes for the Euclidean survivable network design problem," in *Proceedings of ICALP'02*, ser. LNCS 2380. Springer, 2002, pp. 973-984.
- [7] 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)*, 2000, pp. 770-779.
- [8] S. Guha and S. Khuller, "Improved methods for approximating node weighted Steiner trees and connected dominating sets," *Information and Computation*, vol. 150, no. 1, pp. 57-74, 1999.
- [9] 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*, ser. LNCS 5165. Springer, 2008, pp. 278-285.
- [10] T. Erlebach and A. Shahnaz, "Approximating node-weighted multicast trees wireless ad hoc networks," in *Proceedings of IWCMC'09*, 2009, pp. 639-643.
- [11] F. Dai and J. Wu, "On constructing k -connected k -dominating set in wireless ad hoc and sensor networks," *Parallel and Distributed Computing*, vol. 66, pp. 947-958, 2006.