Fault-Tolerant Topology Control for All-to-One and One-to-All Communication in Wireless Networks

Feng Wang, *Member*, *IEEE*, My T. Thai, *Member*, *IEEE*, Yingshu Li, *Member*, *IEEE*, Xiuzhen Cheng, *Member*, *IEEE*, and Ding-Zhu Du, *Member*, *IEEE*

Abstract—This paper introduces the problem of fault-tolerant topology control for all-to-one and one-to-all communication in static wireless networks with asymmetric wireless links. This problem is important in both theoretical and practical aspects. We investigate two approaches, namely, the *minimum-weight-based approach* and the *augmentation-based approach*, to address this problem. Furthermore, we prove that the minimum-weight-based approach has a k-approximation algorithm for all-to-one fault-tolerant topology control, where k is the number of node-disjoint paths. When $k = 1$, this approach solves the minimum power sink tree problem. To the best of our knowledge, this paper is the first to study the fault-tolerant topology control for all-to-one and one-to-all communication in asymmetric static wireless networks and is also the first to demonstrate that the minimum power sink tree problem has a polynomial time optimal solution.

Index Terms—Wireless networks, topology control, fault tolerance, k-InConnectivity, k-OutConnectivity.

1 Introduction

In recent years, wireless networks have been increasingly deployed for both civil and military applications [1]. In most cases, wireless networks are deployed under a harsh environment; thus, it is not surprising that wireless nodes and links experience frequent failures. Such node or link failures often have a significant impact on the performance and reliability of wireless networks and upper level applications. Therefore, ensuring fault tolerance becomes a very important issue in wireless networks. On the other hand, power optimization to prolong the network lifetime is a fundamental problem in wireless networks since wireless nodes are mostly driven by limited power. Topology control [2] has been proposed to save the power consumption of nodes: Each node, instead of using its maximal transmission power, sets its power to a certain level so that the global topology satisfies a certain constraint. The combined problem, providing fault tolerance while minimizing the node power consumption in wireless networks, is very challenging.

In wireless networks, the communication models can be categorized into four classes:

1. all to all, which represents end-to-end communication of every pair of nodes in the network;
2. one to one, which represents the communication from a given source node to a given destination node;
3. all to one, which indicates the communication from all nodes to a given (root) node; and
4. one to all, which indicates the communication from the root to all the other nodes.[1]

Most of the previous studies focused on the fault-tolerant topology control for first two communication models. For example, the authors of [4], [5], and [6] proposed approximations for constructing minimum power k node-disjoint paths between any two nodes, that is, the all-to-all communication model. Srinivas and Modiano [7] gave an optimal solution to construct k node-disjoint paths between the given source and destination, that is, the one-to-one communication model.

Little effort has been put on the all-to-one or one-to-all fault-tolerant topology control, although they are practical in wireless networks. For example, the main objective of sensor networks is to collect data from distributed sensors to the sink; therefore, it is necessary and sufficient to construct k node-disjoint paths from all sensors to the sink. On the other hand, the sink could send control messages to

- F. Wang is with the Mathematical Sciences and Applied Computing Department, Arizona State University, Mail Code: 0852, PO Box 37100, Phoenix, AZ 85069-7100. E-mail: Feng.Wang.4@asu.edu.
- M.T. Thai is with the Department of Computer and Information Sciences and Engineering, University of Florida, CSE Building, Room E301, University of Florida, PO Box 116120, Gainesville, FL 32611-6120. E-mail: mythai@cise.ufl.edu.
- Y. Li is with the Department of Computer Science, Georgia State University, 34 Peachtree Street, Atlanta, GA 30303. E-mail: yli@cs.gsu.edu.
- X. Cheng is with the Department of Computer Science, The George Washington University, 801 22nd St. NW, Suite 704, Washington, DC 20052. E-mail: cheng@gwu.edu.
- D.-Z. Du is with the Department of Computer Science and Engineering, Erik Jonsson School of Engineering & Computer Science, The University of Texas at Dallas, PO Box 830688; MS EC31, Richardson, TX 75083-0688. E-mail: dzdu@utdallas.edu.

Manuscript received 12 Dec. 2005; revised 24 June 2007; accepted 16 July 2007; published online 1 Aug. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0367-1205.
Digital Object Identifier no. 10.1109/TMC.2007.70743.

1. Clearly, if the links are symmetric, all-to-one and one-to-all fault-tolerant topology control are the same. In [3], the symmetric wireless network was studied.

all the other sensors via broadcast, where the traffic is sent from one source to all other nodes in the network. Hence, finding a topology that has k node-disjoint paths from all nodes to one root node and/or has k node-disjoint paths from one root node to all the other nodes is indeed meaningful and practical.

Algorithms for all-to-all fault-tolerant topology control can be applied here, but such approaches consume more power than necessary since not every node needs to be k-connected to every other node. Thus, it is important to find solutions for constructing a topology containing k node-disjoint paths from one specific node to all other nodes while consuming minimum power. We call this problem the *all-to-one* k-*fault-tolerant* problem. Similarly, we need to solve the *one-to-all* k-*fault-tolerant* problem, that is, constructing the topology containing k node-disjoint paths from all other nodes to one specific node with minimum power, and the *concurrent all-to-one* and *one-to-all* k-*fault-tolerant* problem, that is, constructing the topology with minimum power k node-disjoint paths from one-to-all and all-to-one. To the best of our knowledge, our work is the first one to investigate these three problems.

The minimum power broadcast problem [8] is a special case of the one-to-all k-fault-tolerant problem. The works in [9], [10], and [11] proved that the minimum power broadcast is NP-hard, so the one-to-all k-fault-tolerant problem is also NP-hard. The hardness of all-to-one k-fault tolerance is still an open problem. Due to the computational complexity of these problems, approximation algorithms need to be investigated.

In this paper, we investigate two general approaches, namely, the *Minimum-Weight-based (MW) approach* and the *Augmentation-Based (AB) approach*. Under each approach, we propose and analyze three algorithms for the *all-to-one* k-*fault-tolerant*, *all-to-one* k-*fault-tolerant*, and *concurrent all-to-one and one-to-all* k-*fault-tolerant* problems, respectively. The main idea of the MW approach is to utilize the existing minimum weight algorithms and the main idea of AB is to use *free* links as much as possible. Among other results, we show that the Minimum Weight k-INconnected (MW-IN) algorithm is a k-approximation and the Augmentation-Based k-INconnected (AB-IN) algorithm is a $(k+1)$-approximation for the all-to-one k-fault-tolerant problem. The Minimum Weight k-OUTconnected (MW-OUT) algorithm is a Δ^--approximation and the Augmentation-Based k-OUTconnected (AB-OUT) algorithm is a $2\Delta^-$-approximation for the one-to-all k-fault-tolerant problem, where Δ^- is the maximum out-degree in the minimum power one-to-all k-connected topology. An important result of our MW-IN algorithm is that the problem of finding a minimum power topology in which there exists a path from every node to the root node (known as the minimum power sink tree problem) can be solved in polynomial time and any minimum weight sink tree algorithm is such a solution.

We compare the average performance of the MW approach and the AB approach through simulations. The results show that the former performs better for the all-to-one k-fault-tolerant problem and the one-to-all k-fault-tolerant problem, whereas the latter performs better for the concurrent all-to-one and one-to-all k-fault-tolerant problem.

The rest of this paper is organized as follows: Section 2 describes the related work. Section 3 discusses the network model we use and gives the formal problem definition. In Section 4, we present three MW algorithms for the three corresponding problems and analyze their approximation ratios. The AB algorithms are presented and analyzed in Section 5. In Section 6, we show the simulation results. Section 7 concludes this paper.

2 Related Work

Previous work related to fault-tolerant topology control in static wireless networks can be categorized into three classes: 1) finding all-to-all k-node connected subgraphs consuming minimum total power, 2) finding all-to-all k-node connected subgraphs consuming minimum maximal power, that is, the maximal possible power used by each node is minimized, and 3) finding one-to-one (also called source-destination) k-node connected subgraphs consuming minimum total power. Our work is distinguished from previous work in that we investigate the all-to-one and one-to-all k-fault-tolerant topology control problem. Below, we briefly review each of these classes.

Minimum total power for all-to-all k-fault tolerance. The works in [13], [14], and [15] adopt the algorithm BICONNECTED-KR proposed in [16] that constructs a minimum weight 2-connected subgraph and analyze its approximation ratio for minimum power all-to-all 2-connectivity problem. The work in [15] gave the best approximation ratio of 4.

The main idea of the two distributed algorithms for all-to-all 2-connectivity [15], [4] construct a minimum spanning tree first and then augment it to be 2-connected. The work in [15] gave the best approximation ratio of 8.

Hajiaghayi et al. [4] analyzed a linear programming (LP) approach and proved that its approximation ratio is no better than $O(\frac{n}{k})$. In the same paper, the authors also analyzed the k-approximation minimum weight k-connectivity algorithm and proved its approximation ratio of $8k$ for the all-to-all k-fault-tolerant problem. Jia et al. [5] proposed a $3k$-approximation algorithm that first constructs the $(k-1)$th nearest neighbor graph and then augments it to be k-connected using the existing minimum weight k-connected subgraph algorithm.

The only localized algorithm is Fault-Tolerant Cone-Based Topology Control (FCBTC) [12], which generalized the well-known Cone-Based Topology Control (CBTC) [17], [18]. Its approximation ratio is proved in [4] to be $O(\frac{n}{k})$.

Minimum maximal power for all-to-all k-fault tolerance. The works in [19] and [6] fall in the second category, and they gave solutions for minimum maximal power consumption for 2-vertex and k-vertex connectivity, respectively. Both papers proposed greedy algorithms in the sense that, at each iteration, the edge with the minimum weight is chosen until the subgraph becomes 2- or k-connected. Li and Hou [6] also proposed a localized implementation of the centralized algorithm and proved its optimality with regard to minimizing maximal power consumption among all localized algorithms.

TABLE 1
Comparison of Fault-Tolerant Topology Control Algorithms

Problem	Reference	Scheme	Approximation Ratio
all-to-all	Centralized	Hajiaghayi *et al.* (LP) [4]	$O(\frac{n}{k})$
		Hajiaghayi *et al.* [4]	8(k-1)
		Jia *et al.* [5]	3k
	Localized	Bahramgiri *et al.* (FCBTC) [12]	$\geq O(\frac{n}{k})$
one-to-one	Centralized	Srinivas *et al.* [7]	Optimal
all-to-one	Centralized	this paper	k
one-to-all	Centralized	this paper	Δ^-

Minimum total power for source-destination (S-D) k-fault tolerance. Srinivas and Modiano [7] proposed an algorithm called Source Transmit Power Selection (STPS) based on the observation that each internal vertex on the k vertex-disjoint S-D paths has only one outgoing edge and only the source node has more than k outgoing edges. Thus, their algorithm tries every possible power setting for the source node and then applies the minimum weight k vertex-disjoint S-D path algorithm [20] for each setting and picks the one with minimum power consumption. Table 1 summarizes the work on fault-tolerant topology control in wireless networks.

There are also related works addressing the construction of minimum weight one-to-all k vertex-disjoint paths. Frank and Tardos [21] proved that it has an optimal solution by defining a new LP relaxation for the problem and then mapping it to the minimum cost submodular flow problem. The authors of [22] and [23] gave faster implementations for the minimum cost submodular flow problem. In [16] and [24], the authors investigated the construction of k vertex-disjoint paths from one node to all other nodes in undirected graphs. We use these minimum weight algorithms as building blocks for our algorithms.

3 Problem Definitions

In this paper, we use the following common network model: A wireless network consists of N nodes, each of which is equipped with an omnidirectional antenna with a maximal transmission range of r_{max}. The power required for a node to attain a transmission range of r is Cr^{α}, where C is a constant and α is the *power attenuation exponent*, usually chosen between two and four. For any two nodes u and v, there exists a link from u to v if the distance $d(u,v) \leq r_u$, where r_u is the transmission range for node u, determined by its power level. If the links are asymmetric, the existence of a link from u to v does not guarantee the existence of a link from v to u. In this paper, we consider asymmetric links and assume that the wireless network is static, namely, the nodes in the network are stationary.

Given the coordination of the nodes in the plane and the transmission power of the nodes, the network can be mapped into a *cost graph* $G = (V, E, c)$, where V denotes the set of wireless nodes, E denotes the set of wireless links induced by the transmission power, and the weight c for a given edge (u, v) is computed as $Cd(u,v)^{\alpha}$, where d is the distance. By this mapping, an asymmetric wireless network is represented by a directed graph.

A wireless network has an important feature called *Wireless Multicast Advantage (WMA)* because of its broadcast media. WMA can be utilized to save power. For a node to send data to multiple nodes in its transmission range, instead of sending data multiple times, it only needs to send it once, and all nodes in its transmission range can receive the same data. In light of WMA, the power and weight are different in wireless networks.[2] Weight is link-based, whereas power is node-based. The power and weight are defined as follows: Given a cost graph $G = (V, E, c)$, let $p(u)$ be the power assignment of node u, $w(uv)$ be the cost of an edge uv, $c(G)$ be the weight of G, and $p(G)$ be the power of G. Then, we have

- $p(u) = max_{uv \in E} w(uv)$,
- $c(G) = \sum_{e \in E} w(e)$, and
- $p(G) = \sum_{v \in V} p(v)$.

Traditional problems in graph theory are link-based, with the goal of minimizing total weight. However, wireless networks call for node-based algorithms to minimize the total power consumption.

In the following, we give formal definitions for the all-to-one and one-to-all fault-tolerant problem.

Definition 3.1. All-to-one k-fault-tolerant problem. *Given the cost graph of a network and root node r, find the power assignment of each node such that there exist k node-disjoint paths from every other node to r in the induced spanning subgraph and the total node power assignment is minimized. This is also called the minimum power k-inconnectivity problem.*

Definition 3.2. One-to-all k-fault-tolerant problem. *Given the cost graph of a network and root node r, find the power assignment of each node such that there exist k node-disjoint paths from r to every other node in the induced spanning subgraph and the total node power assignment is minimized. This is also called the minimum power k-outconnectivity problem.*

Definition 3.3. Concurrent all-to-one and one-to-all k-fault-tolerant problem. *Given the cost graph of a network and root node r, find the power assignment of each node such that the induced spanning subgraph is both k-inconnected and k-outconnected and the total node power assignment is minimized. This is also called the minimum power k-bothconnectivity problem.*

2. In this paper, we will use cost and weight and power and energy interchangeably.

In this paper, we focus on investigating node connectivity. Our solutions for node connectivity can be applied directly to link connectivity. In the following, we will use *vertex* in the context of a graph and use *node* in the context of a network.

4 Minimum-Weight-Based Approach

The main idea of the MW approach is to construct a minimum weight spanning subgraph that satisfies a certain connectivity constraint. We propose three algorithms called MW-IN, MW-OUT, and Minimum Weight k-BOTHconnected (MW-BOTH) for minimum power k-inconnectivity, minimum power k-outconnectivity, and minimum power k-bothconnectivity problems, respectively. Then, we analyze their approximations.

4.1 Minimum-Weight-Based Algorithms

All three algorithms use an algorithm proposed by Frank and Tardos [21], which constructs a minimum weight k-outconnected subgraph for a directed graph as a building block.[3]

MW-IN is designed based on the following observation: If we first reverse the direction of edges of a given graph D to produce a new graph D', then construct the minimum weight k-outconnected subgraph S' of D', and then reverse the edge of S', we get a minimum weight k-inconnected subgraph S of the original graph D.

Let us call the algorithm constructing a minimum weight k-outconnected subgraph in [21] FT. MW-IN(D, k, r) constructs a k-inconnected subgraph as follows:

- Construct D' by reversing the direction of each edge in D and keep the weight of each edge the same.
- $S' = FT(D', k, r)$.
- Reverse the direction of each edge in S' to get S, where S is the k-inconnected spanning subgraph rooted at r.

The design of MW-OUT and MW-BOTH is similar. MW-OUT is the same as FT. MW-BOTH first applies MW-IN on D, then applies MW-OUT on D, and finally unions the output of MW-IN and MW-OUT. Note that the output of MW-BOTH does not necessarily have the minimum weight.

The dominating step of MW algorithms is the step to apply the FT algorithm. Thus, the time complexity of MW algorithms is the same as that of the FT algorithm, which is $O(n^9 logn)$, where n is the node size.

4.2 Theoretical Analysis

In this section, we give the theoretical analysis of the three algorithms. Theorems 1 through 3 give the approximation ratios of MW-IN, MW-OUT, and MW-BOTH, respectively.[4]

Lemma 1. *For directed graph $D = (V, E, c)$, $p(D) \leq c(D)$.*

Proof. By definition, $p(u) = max_{uv \in E} c(uv) \leq \sum_{uv \in E} c(uv)$ and $p(D) = \sum_{u \in V} p(u) \leq \sum_{u \in V} \sum_{uv \in E} c(uv) = c(D)$. □

Lemma 2. *For directed graph $D = (V, E, c)$, $c(D) \leq \Delta^- p(D)$, where Δ^- is the maximum outgoing degree in D.*

Proof. For each vertex u, $\sum_{uv \in E} c(uv) \leq \Delta^- p(u)$; thus, $c(D) = \sum_{u \in V} \sum_{uv \in E} c(uv) \leq \sum_{u \in V} \Delta^- p(u) = \Delta^- p(D)$. □

Lemma 3 ([21]). *Suppose that a digraph $D = (V, E)$ is k-outconnected from a node r and D is critical, where critical means that D is no longer k-outconnected from r if any edge in D is deleted. Then, the in-degree for every $v \in V - r$ is k.*

Lemma 4. *Suppose that a digraph $D = (V, E)$ is k-inconnected to a node r and D is critical. Then, the out-degree for every $v \in V - r$ is k.*

Proof. We first prove that the reversal of the direction of each edge of a critically k-inconnected graph D is a critically k-outconnected graph D' by contradiction. Suppose that D' is not critically k-outconnected. Then, we can at least remove an edge e; $D' - e$ is still k-outconnected. By reversing the edges of $D' - e$, we get a k-inconnected graph whose edge set is a subset of E; thus, D is not critical anymore. Hence, we know that D' is critically k-outconnected.

From Lemma 3, we know that the in-degree of each vertex in D' except r is k. Actually, the in-degree of a vertex in D' is the out-degree of the corresponding vertex in D; thus, the out-degree of every vertex in D except r is also k. □

Theorem 1. *Let D_{MW-IN} be the output of algorithm MW-IN and $D_{popt-in}$ be a k-inconnected subgraph with optimal power. Then, $p(D_{MW-IN}) \leq kp(D_{popt-in})$.*

Proof. $D_{popt-in}$ is a k-inconnected graph. By deleting all uncritical edges, we can construct a critically k-inconnected subgraph of $D_{popt-in}$. Let us call it $D_{critical}$. From Lemma 4, we know that the out-degree for every $v \in V - r$ in $D_{critical}$ is k. Thus, we have

$$\begin{aligned} p(D_{MW-IN}) &\leq c(D_{MW-IN}) \\ &\leq c(D_{critical}) \\ &\leq kp(D_{critical}) \\ &\leq kp(D_{popt-in}). \end{aligned}$$

The first inequality follows Lemma 1. The second inequality is true because D_{MW-IN} is the k-inconnected subgraph with minimum weight and $D_{critical}$ is also a k-inconnected subgraph. The third inequality follows Lemma 2, and the last inequality is true because $D_{critical}$ is a subgraph of $D_{popt-in}$. □

Corollary 1. *The minimum power sink tree problem has an optimal solution. Any minimum weight sink tree algorithm produces a minimum power sink tree.*[5]

Proof. From Theorem 1, we know that, if $k = 1$, D_{MW-IN} is a sink tree and $p(D_{MW-IN}) \leq p(D_{popt-in})$. Since $p(D_{popt-in})$ is optimal, $p(D_{MW-IN})$ is optimal. □

Theorem 2. *Let D_{MW-OUT} be the output of the algorithm MW-OUT and $D_{popt-out}$ be the k-outconnected subgraph with optimal power. Then, $p(D_{MW-OUT}) \leq \Delta^- p(D_{popt-out})$, where Δ^- is the minimum maximum out-degree of any critically k-outconnected subgraph of $D_{popt-out}$.*

3. The main idea of the algorithm FT is described in the Appendix.

4. The first two lemmas have been proved in [5]; we rewrite them here for completeness.

5. In [25], it has been proved that a sink tree with minimum maximal power can be calculated, while, in our corollary, we state that a sink tree with minimum total power can be calculated.

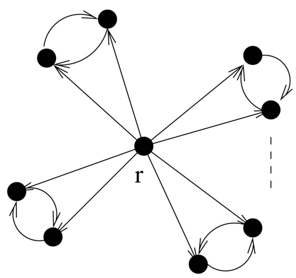


Fig. 1. The out-degree of a minimum power critically k-outconnected graph can be $n-1$. In the figure, the graph is critically 2-outconnected from r, and it consumes minimum power. The out-degree of r is $n-1$.

Proof. This proof is similar to the proof of Theorem 1. Let $D_{critical}$ be a critically k-outconnected subgraph of $D_{popt-out}$.

$$\begin{aligned} p(D_{MW-OUT}) &\le c(D_{MW-OUT}) \\ &\le c(D_{critical}) \\ &\le \Delta^- p(D_{critical}) \\ &\le \Delta^- p(D_{popt-out}). \end{aligned}$$

Fig. 1 shows that Δ^- can be as large as $n-1$, where n is the number of vertices. Note that, for the third inequality, we cannot replace Δ^- with k even though, for a critically k-outconnected graph, the incoming edges to each vertex are k. This is because, in calculating the power of a vertex in a directed graph, only the weight of its outgoing edges counts, whereas the weight of incoming edges does not contribute. This is the main reason that causes the different approximation ratios of the minimum power k-inconnectivity problem and the minimum power k-outconnectivity problem in a directed graph. □

Theorem 3. *Let $D_{MW-BOTH}$ be the output of algorithm MW-BOTH and D_{popt} be the k-bothconnected subgraph with optimal power. Then, $p(D_{MW-BOTH}) \le (k+\Delta^-)p(D_{popt})$.*

Proof. Since $D_{MW-BOTH} = D_{MW-IN} \cup D_{MW-OUT}$,

$$\begin{aligned} p(D_{MW-BOTH}) &\le p(D_{MW-IN}) + p(D_{MW-OUT}) \\ &\le kp(D_{popt-in}) + \Delta^- p(D_{popt-out}) \\ &\le kp(D_{popt}) + \Delta^- p(D_{popt}) \\ &\le (k+\Delta^-)p(D_{popt}). \end{aligned}$$

□

In this section, we present the MW approach and its three proposed algorithms. We also analyze the approximation ratios of these algorithms. Among other results, we show that the minimum power sink tree (1-inconnectivity) problem has an optimal solution, and MW-IN is a k-approximation for the minimum power k-inconnectivity problem for all $k \ge 2$.

5 AB Approach

The main idea of the AB approach is to first construct a graph that consumes small total power and then augment it to satisfy a certain connectivity requirement. In this section, we present three algorithms called AB-IN, AB-OUT, and Augmentation-Based k-BOTHconnected (AB-BOTH) for minimum power k-inconnectivity, minimum power k-outconnectivity, and minimum power k-bothconnectivity, respectively. We first introduce the definitions of the i nearest outgoing/incoming neighbor graph and k-outconnected/k-inconnected augmentation.

Definition 5.1. i **nearest outgoing neighbor graph** D_i^-. *Given $G=(V,E)$, $D_i^-=(V,E')$ is constructed as follows: For each $v \in V$, sort its outgoing edges in nondecreasing order by weight and then choose the first i edges in E'.*

The i nearest incoming neighbor graph D_i^+ can be defined accordingly.

Definition 5.2. k**-outconnected augmentation.** *Given $D=(V,E)$ and its subgraph H, F is called the k-outconnected augmentation to H if $H \cup F$ is the k-outconnected spanning subgraph of D.*

k-inconnected augmentation can be defined accordingly.

5.1 Augmentation-Based Algorithms

In this section, we present our designs for three algorithms belonging to the AB approach: AB-IN, AB-OUT, and AB-BOTH.

AB-IN is designed based on the following observation: For any k-inconnected subgraph, the out-degree of each vertex is at least k. Since the power of a node is the maximum weight of all out-edges incident to this node, the power assignment for a node in any k-inconnected subgraph is at least the weight of the edge whose weight is the kth smallest among all edges. This power assignment causes all the edges with a weight less than the kth smallest weight to appear in any k-inconnected subgraph, including the one with minimum power. Because these are the *free* edges, we want to use as many such edges as possible. Hence, the main idea of AB-IN is to first construct a $(k-1)$-nearest outgoing neighbor graph, then augment it to be k-inconnected by setting the weight of edges in the nearest neighbor graph to zero and apply algorithm MW-IN proposed in Section 4 to find the optimal weight k-inconnected subgraph. AB-IN is similar to the algorithm proposed by Jia et al. [5]. The main differences are that 1) we handle directed graphs, whereas they deal with undirected graphs, and 2) we construct a k-inconnected subgraph, whereas their goal is to construct a k-vertex connected subgraph, that is, there exists k-vertex disjoint paths between any pair of vertices.

AB-IN is illustrated in Algorithm 1.

Algorithm 1. Augmentation-Based k-INconnected AB-IN(D,k,r)

1: INPUT: (D,k,r) where $D=(V,E,c)$ is a directed graph of network topology and r is the root.
2: OUTPUT: a spanning subgraph that is k-inconnected to r
3: Construct $(k-1)$-nearest neighbor graph D_{k-1}^-;
4: Construct D' from D by setting the weight of edges in D_{k-1}^- to zero;
5: Reverse the direction of each edge in D';
6: $H' = FT(D',k,r)$;
7: Reverse the direction of each edge in H'. Let us call the new graph H. H is a spanning subgraph that is k-inconnected to r;

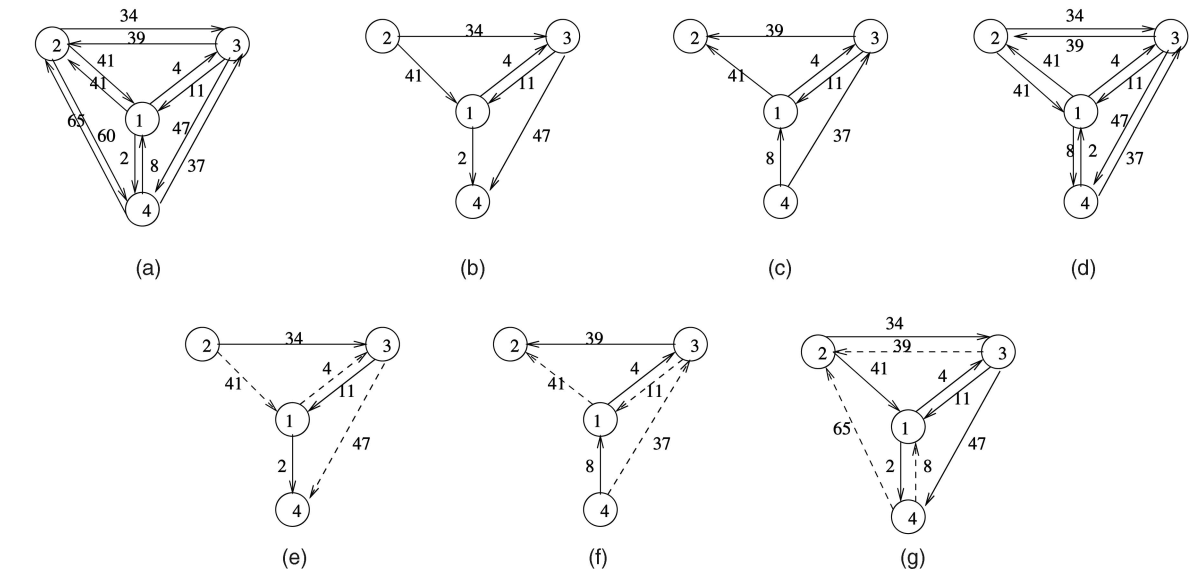


Fig. 2. An example illustrating the MW and AB algorithms. (a) The original topology, where node 4 is the root node. (b), (c), and (d) The 2-inconnected, 2-outconnected, and 2-bothconnected subgraphs constructed by MW-IN, MW-OUT, and MW-BOTH, respectively. (e), (f), and (g) The outputs from AB-IN, AB-OUT, and AB-BOTH, respectively. (e) The solid line is the first nearest outgoing neighbor graph, and the dashed line is its 2-inconnected augmentation. (f) The solid line is the first nearest incoming neighbor graph, and the dashed line is its 2-outconnected augmentation. (g) The solid line is the 2-inconnected subgraph in (e), and the dashed line is its 2-outconnected augmentation.

Similarly, AB-OUT gives a k-outconnected subgraph by first constructing a $(k-1)$-nearest incoming neighbor graph D^+_{k-1} and then augmenting it to be k-outconnected. Note that, since the power assignment of a node is decided by the weight of its outgoing edges instead of its incoming edges, the edges in the $(k-1)$-nearest incoming neighbor graph are not necessarily *free* anymore.

Since a k-bothconnected subgraph is also a k-inconnected subgraph, the main idea of AB-BOTH is to first construct a k-inconnected subgraph and then augment it to be k-outconnected as well. The AB-BOTH algorithm is illustrated in Algorithm 2.

Algorithm 2. Augmentation-Based k-BOTHconnected AB-BOTH (D, k, r)

1: INPUT: (D, k, r), where $G = (V, E, c)$ is a directed graph of network topology and r is the root.
2: OUTPUT: a spanning subgraph where there exist k-vertex disjoint paths between every vertex in $V - r$ and r.
3: $H = AB - IN(D, k, r)$;
4: Construct D' from D by setting the weight of edges in H to zero;
5: $S = FT(D', k, r)$;
6: output $H \cup S$, which is a spanning subgraph that is both k-outconnected from r and k-inconnected to r.

The dominating step of AB algorithms is the step to apply the FT algorithm. Thus, the time complexity of AB algorithms is the same as that of the FT algorithm, which is $O(n^9 logn)$, where n is the node size.

Fig. 2 gives an example illustrating the six algorithms of the two approaches for $k = 2$.

5.2 Theoretical Analysis

In this section, we give the theoretical analysis for AB-IN, AB-OUT, and AB-BOTH. Theorems 4 through 6 give the approximation ratios of these three algorithms.

Lemma 5. *Let $D_{popt-in}$ be the k-inconnected subgraph with optimal power. Then, $p(D^-_{k-1}) \le p(D_{popt-in})$.*

Proof. This is true because D^-_{k-1} is the $(k-1)$-nearest outgoing neighbor graph; its out-degree is $k-1$. The minimum out-degree of $D_{popt-in}$ is at least k because $D_{popt-in}$ is a k-inconnected subgraph. Hence, $p(D^-_{k-1}) \le p(D_{popt-in})$. □

Lemma 6. *Let F_{opt-in} be the minimum weight k-inconnected-augmentation to D^-_{k-1}. Then, $p(F_{opt-in}) \le kp(D_{popt-in})$.*

Proof. Let $D_{critical}$ be a subgraph of $D_{popt-in}$ that is critically k-inconnected to r. We have

$$\begin{aligned} c(D_{critical}) &\le kp(D_{critical}) \\ &\le kp(D_{popt-in}). \end{aligned}$$

The first inequality follows Lemma 4. The second inequality is true because $D_{critical}$ can be constructed by deleting uncritical edges from $D_{popt-in}$; thus, $p(D_{critical}) \le p(D_{popt-in})$. $D_{critical}$ is critically k-inconnected; thus, it contains a k-inconnected augmentation to D^-_{k-1}, since F_{opt-in} is the minimum weight k-inconnected augmentation to D^-_{k-1}, and we have $c(F_{opt-in}) \le c(D_{critical})$. Combining the above results, we have

$$\begin{aligned} p(F_{opt-in}) &\le c(F_{opt-in}) \\ &\le c(D_{critical}) \\ &\le kp(D_{popt-in}). \end{aligned}$$

□

Theorem 4. *Let D_{AB-IN} be the output of AB-IN. Then, $p(D_{AB-IN}) \le (k+1)p(D_{popt-in})$.*

Proof. From Lemmas 5 and 6, we have $p(D_{AB-IN}) \le p(D^-_{k-1}) + p(F_{popt-in}) \le (k+1)p(D_{popt-in})$. □

TABLE 2
Summary of Theoretical Results for the MW Approach and the AB Approach

Problem	MW	AB
minimum power k-inconnectivity	k	k+1
minimum power k-outconnectivity	Δ^-	$2\Delta^-$
minimum power k-bothconnectivity	$k+\Delta^-$	$1+k+\Delta^-$

Theorem 5. *Let D_{AB-OUT} be the output of algorithm AB-OUT and $D_{popt-out}$ be the k-outconnected subgraph with optimal power. Then, $p(D_{AB-OUT}) \leq 2\Delta^- p(D_{popt-out})$, where Δ^- is the maximum out-degree of $D_{popt-out}$.*

Proof. Let $F_{opt-out}$ be the optimal weight k-outconnected augmentation to D^+_{k-1}:

$$\begin{aligned} p(D^+_{k-1}) &\leq c(D^+_{k-1}) \\ &\leq c(D_{popt-out}) \\ p(F_{opt-out}) &\leq c(F_{opt-out}) \\ &\leq c(D_{popt-out}). \end{aligned}$$

The second inequality is true because $D_{popt-out}$ contains an augmentation to D^+_{k-1}. Then, we have

$$\begin{aligned} p(D^+_{k-1} \cup F_{opt-out}) &\leq p(D^+_{k-1}) + p(F_{opt-out}) \\ &\leq 2c(D_{popt-out}) \\ &\leq 2\Delta^- p(D_{popt-out}). \end{aligned}$$

The last inequality follows Lemma 2. □

Theorem 6. *Let $D_{AB-BOTH}$ be the output of algorithm AB-BOTH and D_{popt} be the k-bothconnected subgraph with optimal power. Then, $p(D_{AB-BOTH}) \leq (1+k+\Delta^-)p(D_{popt})$.*

Proof. Let $F_{opt-out}$ be the optimal weight k-outconnected augmentation to D_{AB-IN}:

$$\begin{aligned} p(D_{AB-BOTH}) &\leq p(D_{AB-IN}) + p(F_{opt-out}) \\ &\leq (1+k)p(D_{popt-in}) + \Delta^- p(D_{popt-out}) \\ &\leq (1+k)p(D_{popt}) + \Delta^- p(D_{popt}) \\ &\leq (1+k+\Delta^-)p(D_{popt}). \end{aligned}$$

□

We summarize our results about approximation ratios for both approaches in Table 2, where Δ^- is the maximum out-degree in a minimum power k-outconnected subgraph. The theoretical analysis compares the worst cases. In the next section, we present simulation results to compare their performance in average cases.

6 SIMULATIONS

In this section, we evaluate and compare the performance of the MW approach and the AB approach in terms of the total power of the constructed k-inconnected, k-outconnected, and k-bothconnected spanning subgraphs. Moreover, we illustrate the effects of k to the total power consumption.

To set up the simulation environment, we randomly generate various numbers of nodes in a fixed area and construct a complete cost graph from these nodes by setting the weight of each edge uv as $Cd^2(uv)$, and C is randomly chosen between (0.5, 1.5). For each node size, we run the simulation for 100 times.

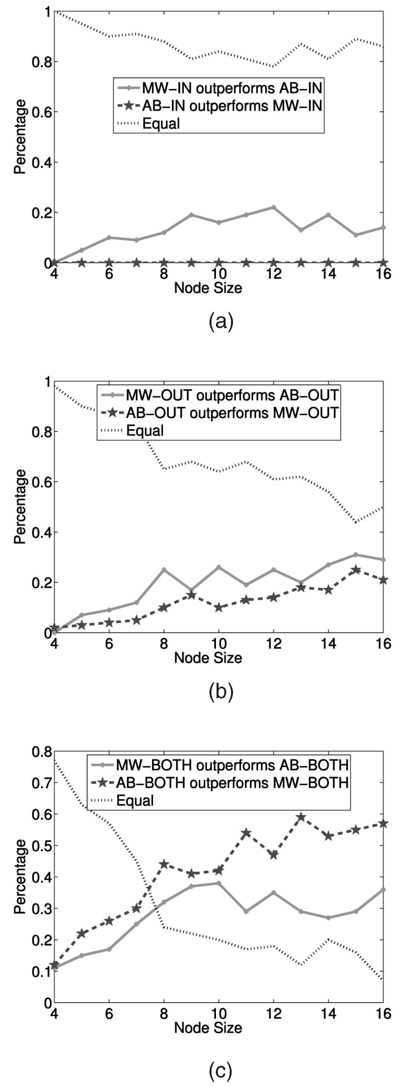
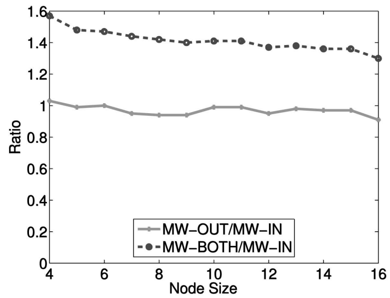


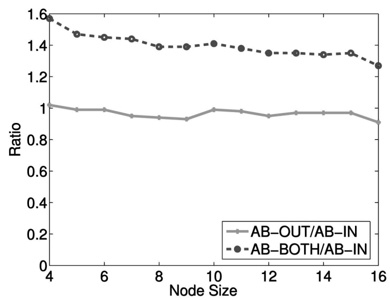
Fig. 3. Total power consumption of the MW approach versus the power of the AB approach. (a) The power consumption of MW-IN versus the power consumption of AB-IN. (b) The power consumption of MW-OUT versus the power consumption of AB-OUT. (c) The power consumption of MW-BOTH versus the power consumption of AB-BOTH.

6.1 Power

In this section, we show the total power consumption of both approaches and the power required for different constraints. Fig. 3 shows the comparison of the total power consumption for both approaches. Over 100 runs for each node size, we calculate the percentage of times when the MW approach outperforms the AB approach, as illustrated by the solid line, the percentage of times when the AB approach outperforms the MW approach, as illustrated by the dashed line with star, and the percentage of times when both approaches use the same power, as illustrated by the dotted line. As we can see in Fig. 3a, MW-IN always uses power less than or equal to that of AB-IN, and under most of



(a)



(b)

Fig. 4. Comparison of power needed for k-inconnectivity, k-outconnectivity, and k-bothconnectivity. (a) MW approach. (b) AB approach.

the cases (more than 80 percent of the times), MW-IN and AB-IN are equally good. This suggests that MW-IN is a good approximation for the minimum power k-inconnectivity problem. This observed result is contrary to our expectation that, under most cases, AB-IN should outperform MW-IN. We speculate that this might suggest that MW-IN is an optimal solution for the all-to-one k-fault-tolerant problem.

Fig. 3b shows that AB-OUT outperforms MW-OUT sometimes, but overall, MW-OUT performs slightly better than AB-OUT for the one-to-all k-fault-tolerant problem. In Fig. 3c, we observe that there is a steep fall of the times when the two approaches perform equally as the node size increases. As in our expectation, AB-BOTH shows better performance in general than MW-BOTH. For example, when the node size is 14, AB-BOTH outperforms MW-BOTH 55 percent of the time. The figure also demonstrates the trend that AB-BOTH outperforms MW-BOTH most of the time as the node size gets larger. This suggests that AB-BOTH is a better approximation for the all-to-one and one-to-all k-fault-tolerant problem.

In Fig. 4, we compare the total power consumption of the constructed k-inconnected, k-outconnected, and k-bothconnected subgraphs. The dotted line shows the ratio of the power for k-outconnectivity over the power for k-inconnectivity, and the dashed line shows the ratio of the power for k-bothconnectivity over the power for k-inconnectivity. As can be seen, the power required for k-inconnectivity is almost the same as the power for k-outconnectivity in both approaches, and the power to maintain a k-bothconnected topology is about 1.4 times the power to maintain a k-inconnected topology.

Table 3 shows the effects of k to the total power consumption. This is the average result over 100 runs for node size 10. We use the power consumption with $k = 2$ as the base. As we expected, the total power increases as k increases, which means that more power is needed to construct a more fault-tolerant topology. The power needed for $k = 4$ is more than double the power needed for $k = 2$. By comparing the power increment as k increases, we notice that the increment is slowest for inconnectivity and the increment for outconnectivity grows fastest. Also, the AB approach performs slightly better than the MW approach in terms of the amount of power increase as k increases.

In general, we find that the MW approach outperforms the AB approach for all-to-one fault-tolerant and one-to-all fault-tolerant problems, and the AB approach suits all-to-one and one-to-all fault tolerance better. We also notice that the two approaches have similar performance in the following aspects: 1) The total power needed for k-outconnectivity is almost the same as the total power needed for k-inconnectivity, which is 2/3 of the total power needed for k-bothconnectivity. 2) The power needed is around 70 percent of the total weight for k-inconnectivity and k-outconnectivity. 3) The power is more than doubled for $k = 4$ compared with that for $k = 2$ for all connectivity requirements.

7 Conclusion and Discussion

In this paper, we introduced the problems of all-to-one, one-to-all, and all-to-one and one-to-all k-fault-tolerant topology control and studied their hardness. We proposed two approaches, namely, the *MW approach* and the *AB approach*, for solving these problems. For each approach, we proposed three algorithms for the three corresponding problems. In addition, we gave a theoretical analysis for all six algorithms and compared their performance using simulation.

Although the proposed algorithms have good approximation ratios, they have high time complexity since their time complexity is dominated by the time complexity of the FT algorithm, which is $O(n^9 logn)$. Hence, the practicality of the proposed algorithms in wireless networks is limited. In the following, we briefly discuss an efficient

TABLE 3
The Effects of k to the Total Power Consumption

Approach	IN			OUT			BOTH		
	k=2	k=3	k=4	k=2	k=3	k=4	k=2	k=3	k=4
MW	1	1.48	2.05	1	1.57	2.31	1	1.52	2.12
AB	1	1.47	2.05	1	1.54	2.28	1	1.50	2.09

heuristic algorithm for constructing a k-inconnected spanning subgraph, which can be practically implemented in wireless networks. Given $G=(V,E)$ and r as the root, first construct a minimum weight sink tree H of G. For each node $v \in V-r$, repeat the following steps until the number of node-disjoint (v,r)-paths in H is equal to k: 1) delete all intermediate nodes on the (v,r)-paths in H from G to produce G', 2) find the shortest (v,r)-paths P_{vr} in G', 3) augment H with P_{vr}. H is the k-inconnected spanning subgraph.

The time complexity of this algorithm is $O(kn^3)$ because the time complexity of calculating the shortest path using Dijastra's algorithm is $O(n^2)$, and it is iterated $k*n$ times. The distributed implementation of this algorithm is trivial because it is well-known that every centralized algorithm has a distributed implementation [26].

Appendix

Description of the FT Algorithm

Since our algorithm is mainly based on the FT algorithm, we briefly explain the main idea of algorithm FT as proposed in [21]. Given $D=(V,E,c)$, define $x : E \rightarrow \{0,1\}$ as follows:

$$x(e) = \begin{cases} 1 & \text{if } e \in \text{a minimum weight } k\text{-outconnected} \\ & \text{subgraph,} \\ 0 & \text{otherwise.} \end{cases}$$

Let $O(A)$ be the set of all incoming edges to node set A, where $A \subseteq V-r$. Let $O_x(A)=\sum_{uv\in O(A)} x(uv)$.

The minimum weight k-outconnectivity problem has the following LP relaxation:

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & O_x(A) \geq k \text{ for any } A \subseteq V-r \\ & 0 \leq x \leq 1. \end{array} \quad (1)$$

An intersecting supermodular [27] $p : 2^V \rightarrow Z \cup \{-\infty\}$ is defined as follows:

$$p(A) = \begin{cases} k & \text{if } A \subseteq V-r \text{ and } A \neq \emptyset, \\ 0 & \text{if } A = \emptyset, \\ -\infty & \text{otherwise.} \end{cases}$$

Now, we can rewrite the LP relaxation of the minimum weight k-outconnected subgraph problem as follows:

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & O_x(A) \geq p(A) \text{ for any } A \subseteq V-r \\ & 0 \leq x \leq 1. \end{array} \quad (2)$$

This problem is an intersecting supermodular flow problem, and the integer solution is the optimal one [27]. Since the integer value of x can only be 0 or 1, by solving this LP, we can get the solution for the minimum weight k-outconnected problem.

Acknowledgments

This work is supported in part by the US National Science Foundation Faculty Early Career Development (CAREER) Award under Grant CCF-0545667 and the Key Program of the National Natural Science Foundation of China under Grants 60533110 and 60473075.

References

[1] I.F. Akyildiz, W. Su, S. Yogesh, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Comm. Magazine*, 2002.

[2] R. Rajaraman, "Topology Control and Routing in Ad Hoc Networks: A Survey," *SIGACT News*, vol. 33, no. 2, pp. 60-73, 2002.

[3] F. Wang, K. Xu, M. Thai, and D.-Z. Du, "Fault Tolerant Topology Control for One-to-All Communications in Symmetric Wireless Networks," *Int'l J. Sensor Networks*, special issue on theoretical and algorithmic aspects in sensor networks, vol. 2, nos. 3/4 pp. 163-168, 2007.

[4] M. Hajiaghayi, N. Immorlica, and V.S. Mirrokni, "Power Optimization in Fault-Tolerant Topology Control Algorithms for Wireless Multi-Hop Networks," *Proc. ACM MobiCom*, 2003.

[5] X. Jia, D. Kim, P. Wan, and C. Yi, "Power Assignment for K-Connectivity in Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM*, 2005.

[6] N. Li and J.C. Hou, "FLSS: A Fault-Tolerant Topology Control Algorithm for Wireless Networks," *Proc. ACM MobiCom*, 2004.

[7] A. Srinivas and E. Modiano, "Minimum Energy Disjoint Path Routing in Wireless Ad-Hoc Networks," *Proc. ACM MobiCom*, 2003.

[8] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," *Proc. IEEE INFOCOM*, 2000.

[9] O. Egecioglu and T.F. Gonzalez, "Minimum-Energy Broadcast in Simple Graphs with Limited Node Power," *Proc. IASTED Int'l Conf. Parallel and Distributed Computing and Systems (PDCS '01)*, 2001.

[10] M. Cagalj, J.-P. Hubaux, and C. Enz, "Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues," *Proc. ACM MobiHoc*, 2002.

[11] D. Li, X. Jia, and H. Liu, "Energy Efficient Broadcast Routing in Static Ad Hoc Wireless Networks," *IEEE Trans. Mobile Computing*, vol. 3, 2004.

[12] M. Bahramgiri, M. Hajiaghayi, and V. Mirrokni, "Fault-Tolerant and 3-Dimensional Distributed Topology Control Algorithms in Wireless Multi-Hop Networks," *Proc. 11th IEEE Int'l Conf. Computer Comm. and Networks (ICCCN '02)*, 2002.

[13] E.L. Lloyd, R. Liu, M.V. Marathe, R. Ramanathan, and S.S. Ravi, "Algorithmic Aspects of Topology Control Problems for Ad Hoc Networks," *Proc. ACM MobiHoc*, 2002.

[14] E.L. Lloyd, R. Liu, M.V. Marathe, R. Ramanathan, and S.S. Ravi, "Algorithmic Aspects of Topology Control Problems for Ad Hoc Networks," *Mobile Network Application*, vol. 10, nos. 1-2, pp. 19-34, 2005.

[15] G. Calinescu and P.-J. Wan, "Range Assignment for High Connectivity in Wireless Ad Hoc Networks," *Proc. Second Int'l Conf. Ad-Hoc Networks and Wireless (ADHOC-NOW '03)*, 2003.

[16] S. Khuller and B. Raghavachari, "Improved Approximation Algorithms for Uniform Connectivity Problems," *Proc. 27th Ann. ACM Symp. Theory of Computing (STOC '95)*, 1995.

[17] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed Topology Control for Wireless Multihop Ad-Hoc Networks," *Proc. IEEE INFOCOM*, 2001.

[18] L. Li, J.Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "Analysis of a Cone-Based Distributed Topology Control Algorithm for Wireless Multi-Hop Networks," *Proc. 20th Ann. ACM Symp. Principles of Distributed Computing (PODC '01)*, 2001.

[19] R. Ramanathan and R. Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment," *Proc. IEEE INFOCOM*, 2000.

[20] J.W. Suurballe, "Disjoint Paths in a Network," *Networks*, pp. 125-145, 1974.

[21] A. Frank and E. Tardos, "An Application of Submodular Flows," *Linear Algebra and Its Applications*, vols. 114/115, pp. 329-348, 1989.

[22] H.N. Gabow, "A Representation for Crossing Set Families with Applications to Submodular Flow Problems," *Proc. Fourth Ann. ACM/SIGACT-SIAM Symp. Discrete Algorithms (SODA '93)*, 1993.

[23] L.K. Fleischer, S. Iwata, and S.T. McCormick, "A Faster Capacity Scaling Algorithm for Minimum Cost Submodular Flow," *Math. Programming*, vol. 92, pp. 119-139, 2002.

[24] J. Cheriyan, T. Jord, and Z. Nutov, "Approximating K-Out-connected Subgraph Problems," *Proc. Int'l Workshop Approximation Algorithms for Combinatorial Optimization (APPROX '98)*, pp. 77-88, 1998.

[25] Y. Li, X. Cheng, and W. Wu, "Optimal Topology Control for Balanced Energy Consumption in Wireless Networks," *J. Parallel Distributed Computing*, vol. 65, no. 2, pp. 124-131, 2005.

[26] N. Lynch, *Distributed Algorithms.* Morgan Kaufmann, 1996.

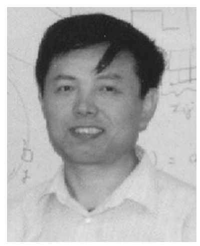
[27] J. Edmonds and R. Giles, "A Min-Max Relation for Submodular Functions on Graphs," *Annals of Discrete Math.*, vol. 1, pp. 185-204, 1977.

Feng Wang received the BS degree from Wuhan University in 1996, the MS degree from Beijing University in 1999, and the PhD degree from the University of Minnesota, Twin Cities, in 2005, all in computer science. She is currently an assistant professor in the Department of Mathematical Sciences and Applied Computing, Arizona State University. Her research interests include wireless networks, network security, computational biology, and combinatorial optimization. She is a member of the IEEE.

My T. Thai received the BS degree in computer science and the BS degree in mathematics from Iowa State University in 1999 and the PhD degree in computer science from the University of Minnesota, Twin Cities, in 2005. She is now an assistant professor in the Department of Computer and Information Sciences and Engineering, University of Florida. Her main research interests include combinatorics, algorithms, wireless networks, and computational biology. Her work has covered many areas of wireless networks and computational biology, including routing protocols, coverage in sensor networks, broadcast tree, virtual backbone, group testing, and nonunique probe selection. She is a member of the IEEE.

Yingshu Li received the BS degree in computer science from the Beijing Institute of Technology, China, in 2001, and the MS and PhD degrees in computer science from the University of Minnesota, Twin Cities, in 2003 and 2005. She is currently an assistant professor in the Department of Computer Science, Georgia State University, Atlanta. Her research interests include wireless networks, optimization theory, and computational biology. She is a member of the IEEE.

Xiuzhen Cheng received the MS and PhD degrees in computer science from the University of Minnesota, Twin Cities, in 2000 and 2002, respectively. She is an assistant professor in the Department of Computer Science, George Washington University. Her research interests include mobile ad hoc and sensor networking and approximation algorithm design and analysis. She is a member of the IEEE.



Ding-Zhu Du received the MS degree from the Chinese Academy of Sciences in 1982 and the PhD degree from the University of California at Santa Barbara under the supervision of Professor Ronald V. Book in 1985. Before settling at the University of Texas at Dallas, he worked as a professor in the Department of Computer Science and Engineering, University of Minnesota. He also worked at the Mathematical Sciences Research Institute, Berkeley, for one year, in the Department of Mathematics, Massachusetts Institute of Technology for one year, and in the Department of Computer Science, Princeton University for one and a half years. Currently, he has published more than 140 journal papers and several books. He is the editor-in-chief of the *Journal of Combinatorial Optimization* and is also on the editorial boards for several other journals. Thirty PhD students have graduated under his supervision. He is a member of the IEEE.