Throughput Optimization in Mobile Backbone Networks

Emily M. Craparo, *Member*, *IEEE*, Jonathan P. How, *Senior Member*, *IEEE*, and
Eytan Modiano, *Senior Member*, *IEEE*

Abstract—This paper describes new algorithms for throughput optimization in a mobile backbone network. This hierarchical communication framework combines mobile backbone nodes, which have superior mobility and communication capability, with regular nodes, which are constrained in mobility and communication capability. An important quantity of interest in mobile backbone networks is the number of regular nodes that can be successfully assigned to mobile backbone nodes at a given throughput level. This paper develops a novel technique for maximizing this quantity in networks of fixed regular nodes using mixed-integer linear programming (MILP). The MILP-based algorithm provides a significant reduction in computation time compared to existing methods and is computationally tractable for problems of moderate size. An approximation algorithm is also developed that is appropriate for large-scale problems. This paper presents a theoretical performance guarantee for the approximation algorithm and also demonstrates its empirical performance. Finally, the mobile backbone network problem is extended to include mobile regular nodes, and exact and approximate solution algorithms are presented for this extension.

Index Terms—Wireless sensor networks, mobile communication systems.

✦

1 INTRODUCTION

DETECTION and monitoring of spatially distributed phenomena often necessitates the distribution of sensing platforms. For example, multiple mobile robots can be used to explore an area of interest more rapidly than a single mobile robot [1], and multiple sensors can provide simultaneous coverage of a relatively large area for an extended period of time [2]. However, in many applications, the data collected by these distributed platforms is best utilized after it has been aggregated, which requires communication among the robotic or sensing agents. This paper focuses on a hierarchical network architecture called a *mobile backbone network*, in which mobile agents are deployed to provide long-term communication support for other agents in the form of a fixed *backbone* over which end-to-end communication can take place. Mobile backbone networks can be used to model a variety of multiagent systems. For example, a heterogeneous system composed of air and ground vehicles conducting ground measurements in a cluttered environment can be appropriately modeled as a mobile backbone network, as can a team of mobile robotic agents deployed to collect streams of data from a network of stationary sensor nodes.

Previous work has focused on optimal placement of mobile backbone nodes in networks of fixed regular nodes, with the objective of providing permanent communication support for the regular nodes [3]. Existing techniques, while exact, suffer from intractable computation times, even for problems of modest size. Furthermore, mobility of regular nodes has not been adequately addressed. This paper provides tractable solutions to the important problem of maximizing the number of regular nodes that achieve a desired level of throughput. It also describes a new mobile backbone network optimization problem formulation that models regular node mobility, and it provides tractable solutions to this problem, including the first known approximation algorithm with computation time that is polynomial in both the number of regular nodes and the number of mobile backbone nodes.

2 BACKGROUND AND PROBLEM STATEMENT

Mobile backbone networks were described by Rubin et al. [4] and Xu et al. [5] as a solution to the scalability issues inherent in mobile ad hoc networks. Noting that most communication bandwidth in single-layer large-scale mobile networks is dedicated to packet forwarding and routing overhead, they proposed a multilayer hierarchical network architecture, as is currently used in the Internet. Srinivas et al. [6] defined two types of nodes: regular nodes, which have limited mobility and communication capability, and mobile backbone nodes, which have greater communication capability than regular nodes and which can be placed at arbitrary locations in order to provide communication support for the regular nodes.

Srinivas et al. [6] formulated the *connected disk cover* (CDC) problem, in which many mobile backbone nodes with fixed communication ranges are deployed to provide communication support for a set of fixed regular nodes. The goal of the CDC problem is to place the minimum number of mobile backbone nodes such that each regular node is covered by at least one mobile backbone node and all

---

- E.M. Craparo is with the Department of Operations Research, Naval Postgraduate School, Monterey, CA 93940. E-mail: emilyc@alum.mit.edu.
- J.P. How and E. Modiano are with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Ave., Room 33-326, Cambridge, MA 02139.
  E-mail: {jhow, modiano}@mit.edu.

mobile backbone nodes are connected to each other. Thus, the CDC problem takes a discrete approach to modeling communication, in that two nodes can communicate if they are within communication range of each other, and otherwise cannot.

This paper uses a more sophisticated model of communication similar to that described by Srinivas and Modiano [3]. We assume that the throughput (data rate) that can be achieved between a regular node and a mobile backbone node is a *monotonically nonincreasing* function of both the *distance* between the two nodes and the *number* of other regular nodes that are also communicating with that particular mobile backbone node and thus causing interference. While our results are valid for any throughput function that is monotonically nonincreasing in both distance and cluster size, it is useful to gain intuition by considering a particular example. One such example is the throughput function resulting from the use of a Slotted Aloha communication protocol in which all regular nodes are equally likely to transmit. In this protocol, the throughput τ between regular node i and mobile backbone node j is given by

$$\tau(A_j, d_{ij}) = \frac{1}{|A_j|} \left(1 - \frac{1}{|A_j|}\right)^{(|A_j|-1)} \left(\frac{1}{d_{ij}^\alpha}\right), \qquad (1)$$

where $|A_j|$ is the number of regular nodes assigned to mobile backbone node j, d_{ij} is the distance between regular node i and mobile backbone node j, and α is the path loss exponent. As noted in [3], one can use the fact that

$$\left(1 - \frac{1}{c}\right)^{c-1} \approx \frac{1}{e}$$

for $c > 0$ to obtain a simpler expression for the Slotted Aloha throughput function. In this simplified expression, the throughput is given by

$$\tau(A_j, d_{ij}) \approx \frac{1}{e \cdot |A_j| \cdot d_{ij}^\alpha}, \qquad (2)$$

where e is the base of the natural logarithm.

Building upon this continuous throughput model, we pose the mobile backbone network optimization problem as follows: given a set of N regular nodes distributed in a plane, our goal is to *place* K mobile backbone nodes, which can occupy arbitrary locations in the plane, while simultaneously *assigning* the regular nodes to the mobile backbone nodes, such that the effectiveness of the resulting network is maximized. In this work, the effectiveness of the resulting network is measured by the number of regular nodes that achieve throughput at least τ_{min}, although other formulations (such as that which maximizes the aggregate throughput achieved by all regular nodes) are possible. The particular choice of objective in this work is motivated by applications such as control over a network, in which a minimum throughput level is required, or sensing applications in which sensor measurements are of a particular (known) size. Thus, our objective is to maximize the number of regular nodes that achieve throughput at least τ_{min}.

Each regular node can be assigned to a single mobile backbone node, and it is assumed that regular nodes assigned to one mobile backbone node encounter no interference from regular nodes assigned to other mobile backbone nodes (e.g., because each "cluster" composed of a mobile backbone node and its assigned regular nodes operates at a different frequency than other clusters). We also assume that there is no need for the mobile backbone nodes to be "connected" to one another. This assumption models the case in which mobile backbone nodes serve to provide a satellite uplink for regular nodes; this is the case, for instance, in hastily formed networks that operate in disaster areas [7]. This assumption is also valid for the case in which the mobile backbone nodes are known to be powerful enough to communicate effectively over the entire problem domain. For cases in which the problem domain is so large that mobile backbone nodes have difficulty communicating with each other, it would be necessary to develop algorithms to ensure connectivity between the mobile backbone nodes (see [6], for example).

We seek to provide the best possible throughput on a permanent basis; therefore, once the mobile backbone nodes have been placed at their optimal positions, no improvement can be obtained by moving further. Thus, our model represents a "one-time" network design problem and is also suitable for cases in which mobile backbone nodes are deployable, but cannot move once they have been deployed. This is in contrast to the *message ferrying* problem, in which regular nodes have a finite amount of data available to transmit, and mobile backbone nodes must move around the network and collect data [8], [9], [10], [11].

We assume that the positions of regular nodes are known with complete accuracy, e.g., because the regular nodes are equipped with GPS. The problem of dealing with error in regular node position estimates is a topic of future research.

As posed, the mobile backbone network optimization problem is quite difficult. Consider a simplification in which the problem is decomposed into two parts: mobile backbone node placement and regular node assignment. Because the mobile backbone nodes can be placed anywhere in the plane, the problem of finding an optimal placement is a hard nonconvex optimization problem even when a simple heuristic technique is used to solve the assignment portion of the problem. Similarly, given a placement of mobile backbone nodes, the assignment portion of the problem is also nontrivial. An exhaustive enumeration of all K^N possible assignments is impractical, and naive assignment techniques, such as that of assigning each regular node to the nearest mobile backbone node, can perform quite badly [3].

Although the problem considered in this paper is similar to that encountered in cellular network optimization, the approaches taken herein differ from those in the cellular literature. Some approaches to cellular network optimization take base station placement to be given, then optimize over user assignment and transmission power to minimize total overall interference [12], [13], [14], [15]. Others assume a simple heuristic for the assignment subproblem and proceed to choose base station locations from among a restricted set [16], [17]. In contrast, we seek to optimize the network simultaneously over mobile backbone node placement and regular node assignment, without assuming variable transmission power capability on the part of the regular nodes and without limiting the placement of the mobile backbone nodes.
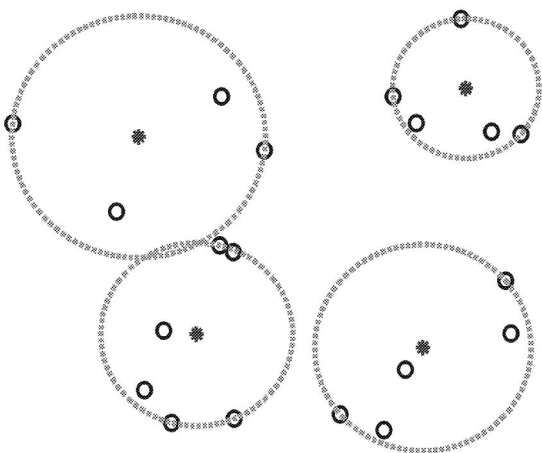
Fig. 1. A typical example of an optimal mobile backbone network. Mobile backbone nodes, indicated by $*$, are placed such that they provide communication support for regular nodes, shown as $\circ$. Each regular node is assigned to one mobile backbone node. Dashed lines indicate the radius of each cluster of nodes.

A typical example of an optimal solution to the simultaneous placement and assignment problem is shown in Fig. 1 for a group of regular nodes denoted by $\circ$. The mobile backbone nodes, denoted by $*$, have been placed, and regular nodes have been assigned to them such that the number of regular nodes that achieve throughput τ_{min} is maximized. This example is typical of an optimal solution in that the clusters of regular nodes and mobile backbone nodes are of relatively small size, and the regular nodes are distributed intelligently among the mobile backbone nodes, with fewer regular nodes being allocated to mobile backbone nodes with larger cluster radii. In this example, all regular nodes have been successfully assigned to mobile backbone nodes.

In the problem considered in Fig. 1, regular nodes are stationary, and their positions are given as problem data, as has been assumed in previous work [3]. New solutions to this problem are described in Section 3. In Section 4, we consider an extension to this problem in which the placement of regular nodes is also a decision variable. That is, the goal in Section 4 is to *place* both N regular nodes and K mobile backbone nodes, while simultaneously *assigning* regular nodes to mobile backbone nodes, such that the effectiveness of the overall network is maximized.

3 STATIONARY REGULAR NODES

This section describes the mobile backbone network optimization problem in which regular nodes are stationary. Denoting the problem data as $\mathbf{R}_i$ (the locations of the regular nodes, $i = 1, \ldots, N$), τ_{min} (the desired minimum throughput level), and τ (the throughput function); and the decision variables as $\mathbf{M}_i$ (the selected locations of the mobile backbone nodes, $i = 1, \ldots, K$) and A (the assignment of regular nodes to mobile backbone nodes), this optimization problem can be stated as:

$$\max_{\mathbf{M}, A} F_\tau(\mathbf{R}, \mathbf{M}, A, \tau_{min})$$

$$\text{subject to } \mathbf{M}_i \in \mathbb{R}^2, i = 1, \ldots, K$$

$$A \in \mathbf{A},$$

where $\mathbf{A}$ is the set of valid assignments (i.e., those in which each regular node is assigned to at most one mobile backbone node), and $F_\tau(\mathbf{R}, \mathbf{M}, A, \tau_{min})$ is the number of regular nodes that achieve throughput level τ_{min}, given node placements $\mathbf{R}$ and $\mathbf{M}$, assignment A, and throughput function τ.

As discussed in Section 2, this problem is quite difficult. Fortunately, it is possible to solve a simpler problem that nonetheless yields an optimal solution to the original problem. A key insight discussed in [3], [18] is that although the mobile backbone nodes can occupy arbitrary locations, they can be restricted to a small number of locations without sacrificing optimality. For throughput functions that are monotonically nonincreasing in distance, each mobile backbone node can be placed at the *1-center* of its assigned regular nodes in an optimal solution.

The 1-center location for a set of regular nodes is the location that minimizes the maximum distance from the mobile backbone node to any of the regular nodes in the set. Consider a feasible solution to the mobile backbone network optimization problem, i.e., a solution in which K mobile backbone nodes are placed anywhere in the plane, each regular node is assigned to at most one mobile backbone node, and each assigned regular node achieves throughput at least τ_{min}. Let A_k denote the set of regular nodes assigned to mobile backbone node k, for $k = 1, \ldots, K$, and let r_k denote the distance from mobile backbone node k to the most distant regular node in A_k. By our assumption that the solution is feasible, we know that $\tau(|A_k|, r_k) \geq \tau_{min}$. Now, modify the solution such that the mobile backbone node k is placed at the 1-center of the set A_k, leaving the assignment A_k unchanged. By definition of the 1-center, the distance from every regular node in A_k to mobile backbone node k is no more than r_k. In particular, if the distance from the mobile backbone node to the most distant regular node in A_k is now denoted by r'_k, we know that $\tau(|A_k|, r'_k) \geq \tau(|A_k|, r_k) \geq \tau_{min}$, since τ is a nonincreasing function of distance. Thus, the modified solution in which the mobile backbone node is placed at the 1-center of its assigned regular nodes is feasible and has the same objective value as the original solution. Repeating the argument for the remaining mobile backbone nodes $1, \ldots, K$, we can see that restricting the feasible set of mobile backbone node locations to the set of 1-center locations of all subsets of regular nodes does not reduce the maximum objective value that can be obtained.

Fortunately, although there are 2^N possible subsets of N regular nodes, there are only $O(N^3)$ distinct 1-center locations [19]. Although a particular 1-center location may correspond to multiple subsets of regular nodes, it is uniquely defined by the regular nodes that are most distant from it in all of these sets. Each 1-center location either coincides with a regular node, lies at the center of the diameter described by two regular nodes, or lies at the circumcenter of three regular nodes [20]. Thus, there are at most $\binom{N}{1} + \binom{N}{2} + \binom{N}{3}$ distinct 1-center locations, and they can be efficiently enumerated through enumeration of the possible sets of "defining" regular nodes. Moreover, the maximum communication radius associated with each possible mobile backbone node location is easy to compute. This radius, which we will denote as the *1-center radius*, is the distance from the 1-center location to any of the defining regular nodes. For 1-center locations defined by the
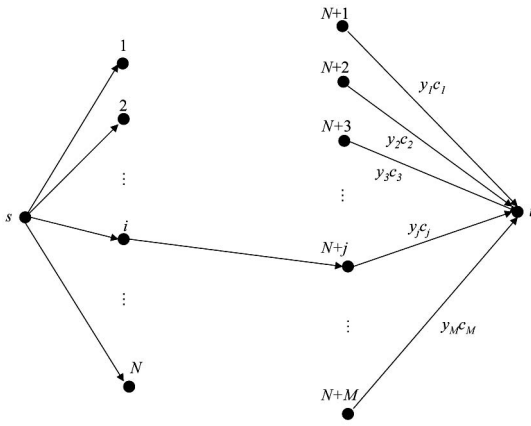
Fig. 2. The network design problem corresponding to the joint placement and assignment problem for mobile backbone networks. Unlabeled arc capacities are equal to 1.

diameter between two nodes or the circumcircle of three nodes, the 1-center radius is simply the radius of the associated circle. For 1-center locations defined by a single regular node, the associated 1-center radius is zero.[1]

The insight that mobile backbone nodes can be restricted to a relatively small number of locations without sacrificing optimality of the overall solution allows the mobile backbone network optimization problem to be simplified. The problem becomes

$$\max_{\mathbf{M},A} F_\tau(\mathbf{R}, \mathbf{M}, A, \tau_{min})$$

$$\text{subject to } \mathbf{M}_i \in \mathbf{C}(\mathbf{R}), i = 1, \ldots, K$$

$$A \in \mathbf{A},$$

where $\mathbf{C}(\mathbf{R})$ denotes the set of 1-center locations of the regular nodes, and $|\mathbf{C}| = O(N^3)$.

3.1 Mixed-Integer Linear Program (MILP) Approach

A primary contribution of this work is the development of a single optimization problem that simultaneously solves the mobile backbone node placement and regular node assignment problems. This is accomplished through the formulation of a *network design* problem. In network design problems, a given network (represented by a directed graph) can be augmented with additional arcs for a given cost, and the goal is to intelligently "purchase" a subset of these arcs in order to achieve a desired flow characteristic [21].

The network design problem that produces an optimal solution to the mobile backbone network optimization problem is constructed as follows: A source node, s, is connected to each node in the set of nodes $N = \{1, \ldots, N\}$ (see Fig. 2). N represents the set of regular nodes. The arcs connecting s to $i \in$ N are of unit capacity. Each node $i \in$ N is, in turn, connected to a subset of the nodes in $M = \{N+1, \ldots, N+M\}$, where M is $O(N^3)$. M represents the set of possible mobile backbone node locations, i.e., the 1-

1. In practice, we consider the 1-center radius of such locations to have a small positive value ε in order to assure boundedness of the throughput function; this modification does not impact the solution as long as ε is such that the throughput achieved by the regular node does not cross the threshold of τ_{min}, and no additional regular nodes fall within a distance of ε of the mobile backbone node.

center locations of the subsets of regular nodes. Node $i \in$ N is connected to node $N + j \in$ M if, and only if, regular node i is within the 1-center radius of location j. The arc connecting i to $N + j$ is of unit capacity. Finally, each node in M is connected to the sink, t. The capacity of the arc connecting node $N + i \in$ M to the sink is the product of a binary variable y_i, which represents the decision of whether to place a mobile backbone node at location i, and a constant c_i, which is the maximum number of regular nodes that can be assigned to a mobile backbone node at location i at the desired throughput level τ_{min}. The quantity c_i can be efficiently computed in one of two ways. For an easily inverted throughput function such as the approximate Slotted Aloha function described by (2), one can simply take the inverse of the expression with respect to cluster size, evaluate the inverse at the desired minimum throughput level τ_{min}, and take the floor of the result to obtain an integer value for c_i. For the throughput function given by (2), we have

$$c_i = \left\lfloor \frac{1}{e \cdot \tau_{min} \cdot r_i^\alpha} \right\rfloor, \tag{3}$$

where r_i is the 1-center radius associated with 1-center location i. If the throughput function cannot easily be inverted with respect to cluster size, as is the case with the exact Slotted Aloha throughput function given in (1), one can perform a search for the largest cluster size $c_i \leq N$ such that $\tau(c_i, r_i) \geq \tau_{min}$. For example, a binary search for c_i would involve $O(\log(N))$ evaluations of the function τ. In either case, the resulting value of c_i is the maximum number of regular nodes that can be assigned to the mobile backbone node at location i, such that each of these regular nodes achieves throughput at least τ_{min}.

Denote the set of nodes for the network design problem by $\mathcal{N}$ and the set of arcs by $\mathcal{A}$. If K mobile backbone nodes are available to provide communication support for N regular nodes at given locations, and a throughput level τ_{min} is specified, the goal of the network design problem is to select K arcs incident to the sink and a feasible flow $\mathbf{x}$ such that the net flow through the graph is maximized.

The network design problem can be solved via the following mixed-integer linear program (MILP), which we denote as the Network Design MILP:

$$\max_{\mathbf{x},\mathbf{y}} \sum_{i=1}^N x_{si} \tag{4a}$$

$$\text{subject to } \sum_{i=1}^M y_i = K, \tag{4b}$$

$$\sum_{j:(i,j)\in\mathcal{A}} x_{ij} = \sum_{l:(l,i)\in\mathcal{A}} x_{li} \quad i \in \mathcal{N} \setminus \{s,t\}, \tag{4c}$$

$$x_{ij} \geq 0 \qquad \forall(i,j) \in \mathcal{A}, \tag{4d}$$

$$x_{ij} \leq 1 \qquad \forall(i,j) \in \mathcal{A} : j \in \mathcal{N} \setminus \{t\}, \tag{4e}$$

$$x_{(N+i)t} \leq y_i c_i \qquad i \in \{1, \ldots, M\}, \tag{4f}$$

$$y_i \in \{0, 1\} \quad i \in \{1, \ldots, M\}, \tag{4g}$$

$$x_{i(N+j)} \leq y_j \quad \forall(i, N+j) \in \mathcal{A} : i \in \{1, \ldots, N\}. \tag{4h}$$

The objective of the Network Design MILP is to maximize the flow $\mathbf{x}$ through the graph (4a). The constraints state that K arcs (mobile backbone node locations) must be

selected (4b), flow through all internal nodes must be conserved (4c), arc capacities must be observed (4d)-(4f), and y_i is binary for all i (4g). Constraint (4h) is a valid inequality that decreases computation time by reducing the size of the feasible set in the LP relaxation [22]. Note that, for a given specification of the **y** vector, all flows **x** are integer in all basic feasible solutions of the resulting (linear) maximum flow problem.

This network design problem exactly corresponds to the mobile backbone network optimization problem as posed in this section. The geometry of the mobile backbone network problem is described by the arcs connecting node sets N and M, while both the throughput function and the desired minimum throughput level are captured in the capacities of the arcs connecting nodes in M to the sink, t. Any feasible placement of mobile backbone nodes and assignment of regular nodes is associated with a feasible solution to the network design problem with the same objective value; likewise, any integer-feasible solution to the network design problem yields a feasible placement and assignment in the mobile backbone network optimization problem, such that the number of regular nodes assigned is equal to the volume of flow through the network design graph. The equivalence of these two problems is formally stated in Theorem 1.

Theorem 1. *Given an instance of the mobile backbone network design problem, the corresponding Network Design MILP yields an optimal solution to the mobile backbone node placement and regular node assignment problems.*

Proof. The proof of Theorem 1 can be found in Appendix A.□

A solution to problem (4) provides both a placement of mobile backbone nodes and an assignment of regular nodes to mobile backbone nodes. Mobile backbone nodes are placed at locations for which $y_i = 1$, and regular node i is assigned to the mobile backbone node at location j if $x_{i(N+j)} = 1$. The number of regular nodes assigned is equal to the volume of flow through the graph.

We make the following observations about this algorithm:

Remark 1. If K mobile backbone nodes are available and the goal is to assign as many regular nodes as possible such that a desired minimum throughput is achieved for each assigned regular node, the MILP problem in (4) needs only to be solved once for the desired throughput value and with a fixed value of K. However, we also note that the Network Design MILP can be used as a subroutine in solving the *maximum fair placement and assignment* (MFPA) problem considered in [3], in which the objective is to maximize the minimum throughput achieved by any regular node, such that all regular nodes are assigned. To solve the MFPA problem, it is necessary to solve the Network Design MILP $O(\log(N))$ times for different throughput values in order to find the maximum throughput value, such that all regular nodes can be assigned. There are at most $O(N^4)$ possible values for the minimum throughput achieved by any regular node; searching among these throughput values via binary search would require $O(\log(N))$ solutions of the Network Design MILP.

Remark 2. It should be noted that the worst-case complexity of mixed-integer linear programming is exponential in

TABLE 1
Average Computation Times for the MILP-Based and Search-Based Algorithms, for Various Numbers of Regular (N) and Mobile Backbone Nodes (K) in the Maximum Fair Placement and Assignment (MFPA) Problem

N	K	MILP Algorithm	Search-based Algorithm
3	2	.92 sec	3.12 sec
4	2	1.24 sec	14.97 sec
5	2	1.35 sec	45.15 sec
6	2	1.37 sec	1 min 28 sec
6	3	1.45 sec	5 min 44 sec
8	3	1.95 sec	33 min
10	3	2.23 sec	2 hr 6 min
15	5	5.06 sec	5 hr 4 min

All models were formulated in GAMS 22.9 and solved using ILOG CPLEX 11.2.0 on a 3.16 GHz Intel Xeon CPU with 3.25 GB of RAM.

the number of binary variables. However, this approach performs well in practice, and simulation results indicate that it compares very favorably with the search-based approach developed in [3] for the MFPA problem (see Table 1). Note that while the computation time of the search-based algorithm increases very rapidly with the problem size, the MILP-based algorithm remains computationally tractable for problems of practical scale.

3.2 Approximation Algorithm

Table 1 indicates that the MILP formulation described by (4) provides an optimal solution in tractable time for moderately sized problems. However, this method is demonstrated to scale poorly with problem size. Moreover, we have shown that the network design problem on a network of the general form shown in Fig. 2 is NP-hard [23]. Therefore, an approximation algorithm with computation time that is polynomial in the number of regular nodes and the number of mobile backbone nodes is desirable. This section describes such an algorithm.

The primary insight on which the approximation algorithm is based is the fact that the maximum number of regular nodes that can be assigned is a *submodular* function of the set of mobile backbone node locations selected. Given a finite ground set $D = \{1, \ldots, d\}$, a set function $f(S)$ defined for all subsets S of D is said to be submodular if it has the property that

$$f(S \cup \{i, j\}) - f(S \cup \{i\}) \le f(S \cup \{j\}) - f(S),$$

for all $i, j \in D, i \ne j$ and $S \subset D \setminus \{i, j\}$ [24]. In the context of the network design problem, this means that the maximum flow through the network is a submodular function of the set of arcs incident to the sink that are selected.

It has been shown [25] that for maximization of a nondecreasing submodular set function f, where $f(\emptyset) = 0$, greedy selection of elements yields a performance guarantee of $1 - (1 - \frac{1}{P})^P > 1 - \frac{1}{e}$, where P is the number of elements to be selected from the ground set and e is the base of the natural logarithm. This means that if an exact algorithm selects P elements from the ground set and produces a solution of value OPT, a greedy selection of P elements (i.e., selection via a process in which element i is selected if it is the element that maximizes $f(S \cup \{i\})$, where

S is the set of elements already selected) produces a solution of value at least $(1 - (1 - \frac{1}{P})^P) \cdot OPT$.

For the network design problem considered in this paper, $P = K$ (the number of mobile backbone nodes that are to be placed), and OPT is the number of regular nodes that are assigned in an optimal solution. Note that greedy selection of K arcs amounts to solving at most $O(N^3 K)$ linear maximum flow problems on graphs with at most $N + K + 2$ nodes. Thus, the computation time of the greedy algorithm is polynomial in the number of regular nodes and the number of mobile backbone nodes. Furthermore, each of the maximum flow problems solved by the greedy algorithm is solved over a bipartite graph with node sets $N \cup \{t\}$ and $\{s\} \cup K$, where K is the set of nodes from M whose outgoing arcs are selected. Because maximum flow problems can be solved even more efficiently in bipartite networks than in general networks [21], the greedy algorithm is thus highly efficient. Further computational efficiency can result from exploitation of max-flow/min-cut duality [23].

The submodularity of the network design objective is formally stated in Lemma 1:

Lemma 1. *If G is a graph in the network design problem described in Section 3.1, the maximum flow that can be routed through G is a submodular function of the set of arcs selected.*

Proof. The proof of Lemma 1 can be found in Appendix B.□

Lemma 1 implies that greedy selection of mobile backbone node locations (i.e., selection via a process that maximizes the total number of regular nodes assigned for each incremental selection of a new mobile backbone node location) yields provably good solution to the overall placement and assignment problem.

Given a network design graph G, K mobile backbone nodes, and M possible mobile backbone node locations, and denoting by f the maximum flow through G as a function of the set of mobile backbone node locations selected, this greedy selection process is described by Algorithm 1. Algorithm 1 begins with an empty set of selected mobile backbone node locations, S, and incrementally adds K elements to this set. In each of K rounds of selection, $M - |S|$ maximum flow problems are solved on graphs consisting of nodes s, t, N, S and each of the $M - |S|$ nodes belonging to set $M \setminus S$. In each round of selection, the node from set $M \setminus S$ that maximizes the total flow through G is added to set S, and this process continues until $|S| = K$. Algorithm 1 then returns set S. The performance of Algorithm 1 is described by Theorem 2.

Algorithm 1.
$S \leftarrow \emptyset$
$maxflow \leftarrow 0$
for $k = 1$ to K **do**
  for $m = 1$ to M **do**
    if $f(S \cup \{m\}) \geq maxflow$ **then**
      $maxflow \leftarrow f(S \cup \{m\})$
      $m^* \leftarrow m$
    end if
  end for
  $S \leftarrow S \cup \{m^*\}$
end for
return S

Theorem 2. *Algorithm 1 returns a solution S such that $f(S) \geq \lceil (1 - \frac{1}{e}) \cdot f(S^*) \rceil$, where S^* is the optimal solution to the network design problem on G.*

Proof. This follows from the fact that all maximum flows through G are integer, and from the observation that the maximum flow that can be routed through G is a submodular function of S, the set of arcs that are selected. □

3.3 Experimental Evaluation of Approximation Algorithm

As described in Section 3.2, greedy selection of mobile backbone node locations results in assignment of at least $\lceil (1 - (1 - \frac{1}{K})^K) \cdot OPT \rceil \geq \lceil (1 - \frac{1}{e}) \cdot OPT \rceil$ regular nodes, where K is the number of mobile backbone nodes that are to be placed and OPT is the number of regular nodes assigned by an exact algorithm (such as the MILP algorithm described in Section 3.1) [25]. While such *worst-case* performance guarantees are quite useful, it is also worthwhile to examine the *typical* performance of the approximation algorithm on many problems.

To this end, we have performed computational experiments on a number of problems of various degrees of complexity. Regular node locations were generated randomly in a finite 2D area, and a moderate throughput value was specified (i.e., one high enough such that there was no trivial selection of mobile backbone node locations that would result in assignment of all regular nodes). Results were averaged over a number of trials for each problem dimension.

Fig. 3 shows the performance of the approximation algorithm relative to the exact (MILP) algorithm. In Fig. 3a, the average percentage of regular nodes assigned by the exact algorithm that are also assigned by the approximation algorithm is plotted, along with the theoretical lower bound of $\lceil (1 - \frac{1}{e}) \cdot OPT \rceil$, for various problem sizes. In this figure, a data point at 100 percent would mean that, on average, the approximation algorithm assigned as many regular nodes as the exact algorithm for that particular problem size. As the graph shows, the approximation algorithm consistently exceeds the theoretical performance guarantee and achieves nearly the same level of performance as the exact algorithm for all problem sizes considered.

Fig. 3b shows the computation time required for each of these algorithms, plotted on a logarithmic axis. As the figure shows, the computation time required for the approximation algorithm scales gracefully with problem size. The average computation time of the approximation algorithm was about 15 seconds for $N = 100$ and $K = 14$, whereas the MILP algorithm took nearly 12 minutes to solve a problem of this size. The significant improvement in computation time achieved by the approximation algorithm makes it appropriate for some real-time applications, while the exact algorithm is a promising candidate for one-time design problems involving significant costs.

Both the MILP algorithm and the approximation algorithm were formulated in GAMS 22.9 and solved using ILOG CPLEX 11.2.0 on a 3.16 GHz Intel Xeon CPU with 3.25 GB of RAM.
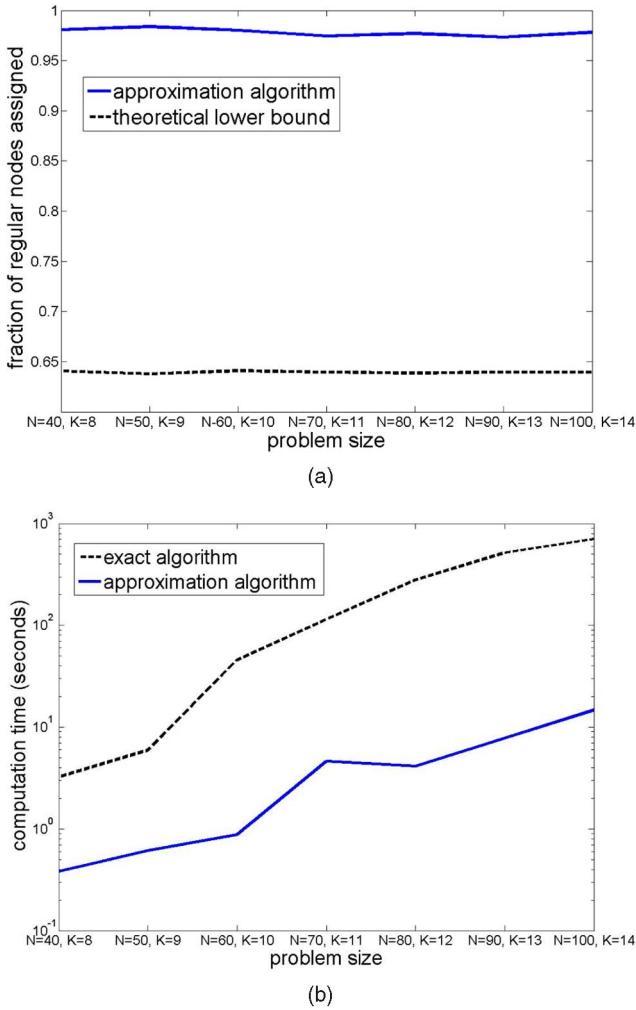
(a)

(b)

Fig. 3. Comparison of the exact and approximation algorithms developed in this section. Although the MILP-based exact algorithm developed in this section significantly outperforms existing techniques in terms of required computation time, our experiments indicate that the greedy approximation algorithm achieves nearly the same level of performance with an even greater reduction in computation time. (a) Performance of the approximation algorithm developed in this paper, relative to an exact solution technique, in terms of number of regular nodes assigned at the given throughput level. (b) Computation time of the approximation algorithm and the exact (MILP) algorithm for various problem sizes. Due to the large range of values represented, a logarithmic scale is used.

4 JOINT PLACEMENT OF REGULAR AND MOBILE BACKBONE NODES

As described in Section 3, existing problem formulations in the study of mobile backbone networks have assumed that the locations of regular nodes are fixed a priori and that only the locations of mobile backbone nodes are variable [18], [3], [6]. This assumption is reasonable for some applications, such as scenarios that involve mobile agents extracting data from a fixed sensor network. However, in many applications, the locations of both regular nodes and mobile backbone nodes can be controlled. For example, a heterogeneous system composed of air and ground vehicles conducting ground measurements in an urban environment can be appropriately modeled as a mobile backbone network: the ground vehicles are well-positioned to make

observations of phenomena at ground level, but their movement and communication are hindered by surrounding obstacles. Air vehicles, on the other hand, are poorly equipped to observe events on the ground but can easily move above ground obstacles and communicate.

This section develops a modeling framework and solution technique that are appropriate for problems of this nature. We assume that L candidate regular node locations are available a priori, perhaps selected by heuristic means or due to logistical constraints. Each of N regular nodes ($N \leq L$) must occupy one of these locations, and no two regular nodes can be assigned to the same location. Given an initial location and a mobility constraint, each regular node is capable of reaching a subset of the other locations. There are K mobile backbone nodes ($K \leq N$) that can be placed anywhere, a throughput function τ is specified, and a desired minimum throughput τ_{min} is given.

Given these assumptions, the goal of this section is to *place* both the regular nodes and mobile backbone nodes while simultaneously *assigning* regular nodes to mobile backbone nodes in order to maximize the number of regular nodes that are successfully assigned and achieve the desired minimum throughput level τ_{min}, under the given throughput function τ.

Denoting the problem data as $\mathbf{I}_i$ (the initial locations of the regular nodes, $i = 1, \ldots, N$) and $r(\mathbf{I}_i)$ (the locations reachable from each of the initial regular node locations); and the decision variables as $\mathbf{F}_i$ (the final locations of the regular nodes), $\mathbf{M}_i$ (the selected locations of the mobile backbone nodes, $i = 1, \ldots, K$), and A (the assignment of regular nodes to mobile backbone nodes), this optimization problem is:

$$\max_{\mathbf{F}, \mathbf{M}, A} F_\tau(\mathbf{F}, \mathbf{M}, A, \tau_{min})$$
$$\text{subject to } \mathbf{F}_i \in r(\mathbf{I}_i), i = 1, \ldots, N$$
$$\mathbf{M}_i \in \mathbf{C}(\mathbf{I}), i = 1, \ldots, K$$
$$A \in \mathbf{A},$$

where $F_\tau(\mathbf{F}, \mathbf{M}, A, \tau_{min})$ is the number of regular nodes that achieve throughput level τ_{min}, given node placements $\mathbf{F}$ and $\mathbf{M}$, assignment A, and throughput function τ. $\mathbf{A}$ denotes the set of valid assignments, while $\mathbf{C}$ denotes the 1-centers of candidate regular node locations.

4.1 Network Design Formulation

Optimal simultaneous placement and assignment of regular nodes and mobile backbone nodes is again achieved through the solution of a network design problem.

The network design graph over which this optimization takes place is schematically represented in Fig. 4. This graph is similar to that shown in Fig. 2, with the exception that sets of nodes L = $\{N + 1, \ldots, N + L\}$ and L'= $\{N + L + 1, \ldots, N + 2L\}$ are added. These nodes represent locations to which regular nodes may move. Node $i \in$ N is connected to node $N + j \in$ L if, and only if, regular node i can reach sensing location j under its mobility constraint. Each node $N + i \in$ L is connected to node $N + L + i \in$ L' via an arc of unit capacity. This enforces the constraint that at most one regular node can occupy each location. Finally, node $N + L + i \in$ L' is connected to node $N + 2L + j \in$ M if, and only if, sensing location i is within the 1-center radius of 1-center
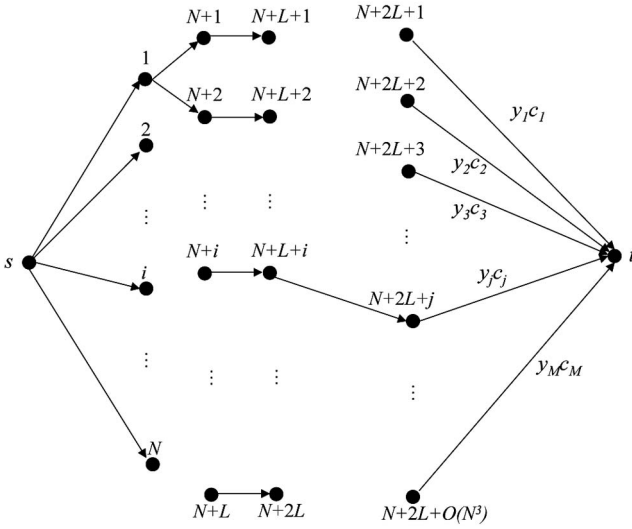
Fig. 4. The network design problem corresponding to the joint placement and assignment problem for mobile backbone networks, with regular node mobility. Unlabeled arc capacities are equal to 1. For clarity, not all arcs and nodes are shown.

j. Finally, each node in M is connected to the sink t, and the capacity of the arc connecting $N + 2L + i \in$ M to t is again the product of y_i and c_i.

This network design problem exactly describes the mobile backbone network optimization problem with mobile regular nodes. The arcs connecting node sets N and L reflect the mobility constraints of the regular nodes, while the geometric aspects of the mobile backbone node placement problem are described by the arcs connecting node sets L' and M. As in Section 3, both the throughput function and the desired minimum throughput level are captured in the capacities of the arcs connecting nodes in M to the sink, t. Any feasible placement and assignment of regular and mobile backbone nodes is associated with a feasible solution to the network design problem; likewise, any feasible solution to the network design problem yields a feasible placement and assignment in the mobile backbone network optimization problem.

Denote the set of nodes in the network design graph by $\mathcal{N}$ and the set of arcs by $\mathcal{A}$. If K mobile backbone nodes are available and a minimum throughput level is specified, the goal of the network design problem is to select K arcs from $\{N + 2L + 1, \ldots, N + 2L + M\}$ and a feasible flow $x_{ij}, (i, j) \in \mathcal{A}$ such that the $s - t$ flow is maximized. This problem can be solved via the following MILP:

$$\max_{\mathbf{x,y}} \sum_{i=1}^{N} x_{si} \tag{5a}$$

$$\text{subject to} \sum_{i=1}^{M} y_i = K, \tag{5b}$$

$$\sum_{j:(i,j)\in\mathcal{A}} x_{ij} = \sum_{l:(l,i)\in\mathcal{A}} x_{li} \quad i \in \mathcal{N} \setminus \{s, t\}, \tag{5c}$$

$$x_{ij} \geq 0 \quad \forall (i, j) \in \mathcal{A}, \tag{5d}$$

$$x_{ij} \leq 1 \quad \forall (i, j) \in \mathcal{A} : j \in \mathcal{N} \setminus \{t\}, \tag{5e}$$

$$x_{(N+2L+i)t} \leq y_i c_i \quad i \in \{1, \ldots, M\}, \tag{5f}$$

$$y_i \in \{0, 1\} \quad i \in \{1, \ldots, M\}, \tag{5g}$$

$$x_{(N+L+i)(N+2L+j)} \leq y_j \quad \forall i \in \{1, \ldots, L\},$$
$$(N + L + i, N + 2L + j) \in \mathcal{A}, \tag{5h}$$

where the constraints state that K arcs (mobile backbone node locations) must be selected (5b), flow through all internal nodes must be conserved (5c), arc capacities must be observed (5d)-(5f), and y_i is binary for all i (5g). Constraint (5h) is again a valid inequality included to reduce computation time by strengthening the LP relaxation [22].

Fig. 5 shows an example of a solution to the simultaneous placement and assignment problem with regular node movement. The regular nodes, initially in positions indicated by •, are able to move to other locations (∘) within their radii of motion, indicated by shaded circles. This initial configuration is shown in Fig. 5a. In an optimal solution to this problem, shown in Fig. 5b, the regular nodes have moved such that they are grouped into compact clusters for which the mobile backbone nodes can provide an effective communication infrastructure. The clusters are relatively balanced, in that the clusters with larger radii tend to have fewer regular nodes, while the more compact clusters can accommodate more regular nodes and still achieve the desired minimum throughput. In this example, all regular nodes have been successfully assigned to mobile backbone nodes.

This algorithm is designed to maximize the number of regular nodes that are assigned at throughput level τ_{min}. If, instead, the goal is to achieve the best possible minimum throughput such that *all* regular nodes are assigned to a mobile backbone node (i.e., to solve the MFPA problem), it is necessary to solve the MILP problem in (5) $O(\log(L))$ times for different throughput values (which result in different values for $c_1, \ldots, c_M$ in the network design problem).

In this paper, all candidate regular node locations are considered to be equally valuable—that is, a regular node that transmits to a mobile backbone node from location i contributes as much to the objective as a mobile backbone node transmitting from location j. However, this formulation can easily be modified to model variably valuable sensing locations by replacing the objective function with

$$\max_{\mathbf{x,y}} \sum_{i=1}^{L} v_i x_{N+i,N+L+i},$$

where v_i is the value of location i.

4.2 Approximation Algorithm

While the MILP-based algorithm described in Section 4.1 provides an exact solution to the mobile backbone network optimization problem, its worst-case computation time is again exponential in the number of binary variables. Fortunately, the submodularity property described in Section 3.2 also holds for graphs of the form shown in Fig. 4:

Lemma 2. *If G is a graph in the network design problem described in Section 4.1, the maximum flow that can be routed through G is a submodular function of the set of arcs selected.*

Proof. The proof of Lemma 2 is similar to that of Lemma 1, with the exception that the reduction to a bipartite matching problem is replaced with a reduction to a node-disjoint path problem on a tripartite graph. □

(a)   (b)
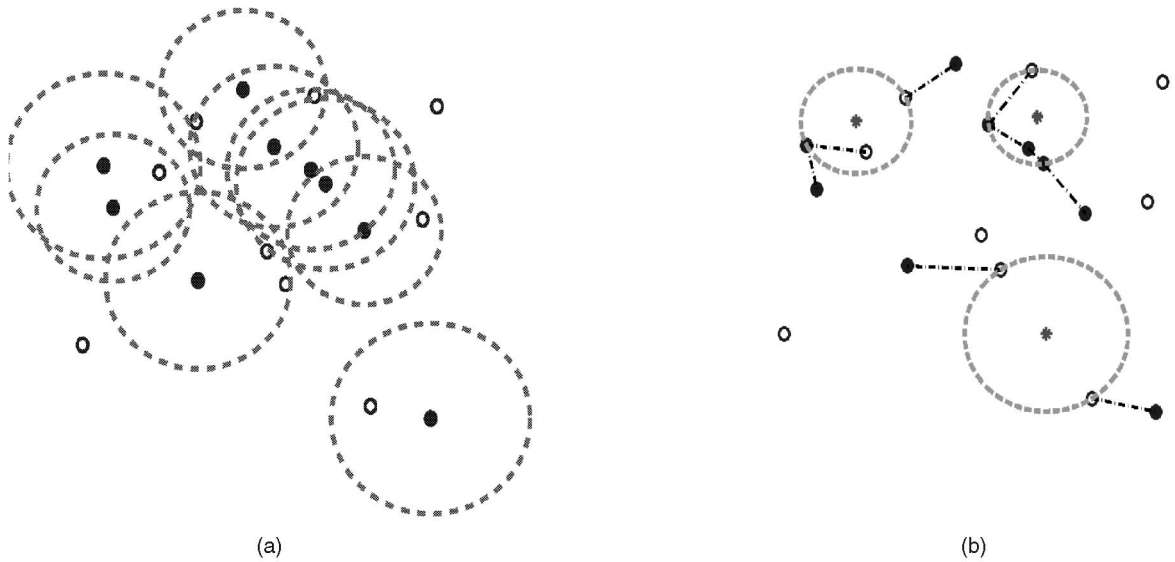
Fig. 5. A small example of mobile backbone network optimization with limited regular node movement. Open circles represent possible regular node locations, and filled circles are the initial locations of the regular nodes. Shaded circles in (a) indicate the possible radius of motion of each regular node. In (b), mobile backbone nodes are placed such that they provide communication support for the regular nodes. Each regular node is assigned to at most one mobile backbone node. Dotted lines indicate regular node motion in this optimal solution. Dashed circles indicate the communication radius of each cluster of nodes. In this example, all regular nodes have been successfully assigned to mobile backbone nodes. (a) Initial regular node placement, with radius of motion for each regular node. (b) An optimal placement of regular and mobile backbone nodes.

Lemma 2 motivates consideration of a greedy algorithm for the problem of maximizing the number of regular nodes that achieve throughput level τ_{min}. Given a network design graph G of the form shown in Fig. 4, with K mobile backbone nodes and M possible mobile backbone node locations, and denoting by f the maximum flow through G as a function of the set of mobile backbone node locations selected, this greedy algorithm is described by Algorithm 2. Theorem 3 describes the performance of Algorithm 2.

Algorithm 2.
  $S \leftarrow \emptyset$
  $maxflow \leftarrow 0$
  for $k = 1$ to K **do**
    for $m = 1$ to M **do**
      if $f(S \cup \{m\}) \geq maxflow$ **then**
        $maxflow \leftarrow f(S \cup \{m\})$
        $m^* \leftarrow m$
      end if
    end for
    $S \leftarrow S \cup \{m^*\}$
  end for
  return S

Theorem 3. *Algorithm 2 returns a solution S such that $f(S) \geq \left\lceil (1 - \frac{1}{e}) \cdot f(S^*) \right\rceil$, where S^* is the optimal solution to the network design problem on G.*

Proof. This follows from the fact that all maximum flows through G are integer, and from the observation that the maximum flow that can be routed through G is a submodular function of S, the set of arcs that are selected.   □

Thus, Algorithm 2 is an approximation algorithm with approximation guarantee $1 - \frac{1}{e}$. Additionally, because each round of greedy selection consists of solving a polynomial

number of maximum flow problems on graphs with at most $N + 2L + K + 2$ nodes, and there are K rounds of selection, the running time of Algorithm 2 is polynomial in the number of regular nodes, the number of locations, and the number of mobile backbone nodes. Furthermore, all network flow problems solved by Algorithm 2 are formulated on bipartite graphs, for which highly efficient algorithms exist [21].

The performance of Algorithm 2 relative to the exact (MILP) algorithm developed in this section is shown in Fig. 6. Again, the approximation algorithm exhibits excellent empirical performance, achieving results comparable to those of the exact algorithm with a great reduction in computation time. The results shown in Fig. 6 were obtained from models formulated in GAMS 22.9 and solved using ILOG CPLEX 11.2.0 on a 3.16 GHz Intel Xeon CPU with 3.25 GB of RAM.

5   CONCLUSIONS

This work has described new algorithms for solving the problem of mobile backbone network optimization. Exact MILP-based techniques and the first known approximation algorithms with computation time polynomial in the number of regular nodes and the number of mobile backbone nodes were described.

Based on simulation results, we conclude that the MILP-based approach provides a considerable computational advantage over existing techniques for mobile backbone network optimization. This approach has been successfully applied to a problem in which a maximum number of regular nodes are to be assigned to mobile backbone nodes at a given level of throughput, and to a related problem from the literature in which all regular nodes are to be assigned to a mobile backbone node such that the minimum throughput achieved by any regular node is maximized.
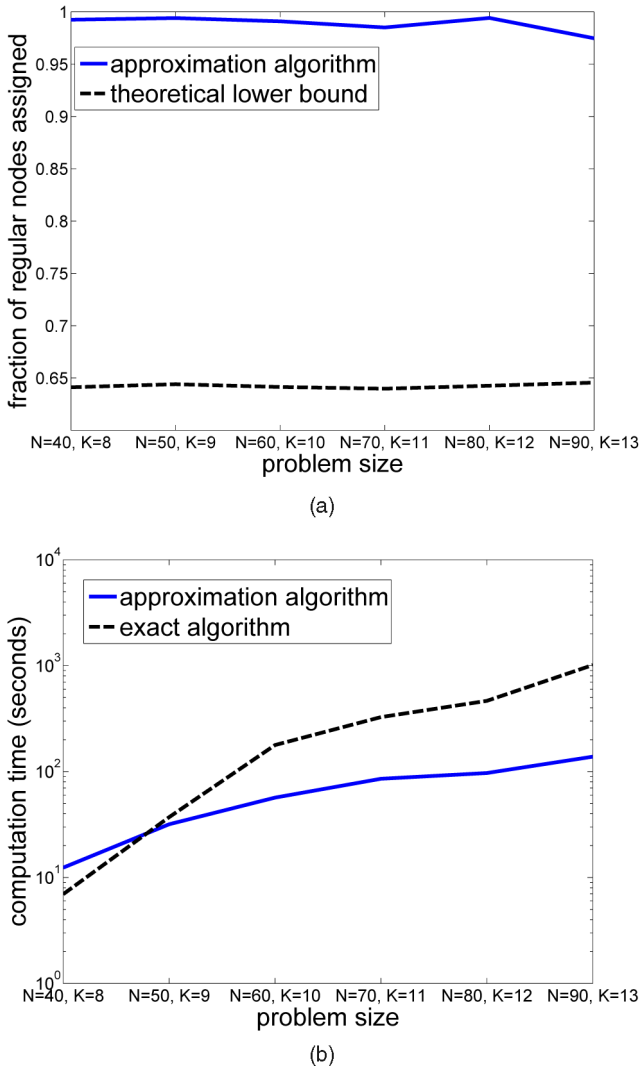
(a)

(b)

Fig. 6. Comparison of the exact and approximation algorithms developed in this section. On average, the approximation algorithm greatly exceeded its performance guarantee, achieving nearly the same level of performance as the exact algorithm for all problem sizes considered. In all cases, $L = 1.5$ N. (a) Performance of the approximation algorithm developed in this section, relative to an exact solution technique, in terms of number of regular nodes assigned at the given throughput level. (b) Computation time of the approximation algorithm and the exact (MILP) algorithm for various problem sizes. Due to the large range of values represented, a logarithmic scale is used.

For cases in which an MILP approach is impractical due to constraints on computation time, the greedy approximation algorithms developed in this paper present viable alternatives. The approximation algorithms carry the benefit of a theoretical performance guarantee, and simulation results indicate that they perform very well for the problem of assigning a maximum number of regular nodes such that each assigned regular node achieves a minimum throughput level.

APPENDIX A

PROOF OF THEOREM 1

Proof. Given a feasible solution to the original problem (i.e., a solution in which K mobile backbone nodes are placed

anywhere in the plane, each regular node is assigned to at most one mobile backbone node, and every assigned regular node achieves throughput at least τ_{min}), a feasible solution to the corresponding network design problem and its associated Network Design MILP (4) can be constructed as follows:

Let A_k denote the set of regular nodes assigned to mobile backbone node k, for $k = 1, \ldots, K$. Calculate the 1-center of set A_k and denote its location by l_k and its 1-center radius by r_k. Note that although some 1-center locations may coincide, each will be distinct in the set M. By our assumption that we are given a feasible solution to the original problem, we know that $A_i \cap A_j = \emptyset \; \forall i, j \in \{1, \ldots, K\}, i \neq j$. Therefore, the defining regular nodes of each 1-center must be distinct, and l_i is distinct from l_j in the set M.

Assume without loss of generality that the network design graph is labeled such that nodes $N + 1, \ldots, N + K$ correspond to the locations $l_1, \ldots, l_K$. Set binary variables $y_1, \ldots, y_K$ equal to 1, and set the remaining binary variables $y_{K+1}, \ldots, y_M$ equal to 0. Note that constraints 4b and 4g in the Network Design MILP are now satisfied.

Next, for each regular node i that is a member of set A_k for some k, set x_{si} equal to 1. Set x_{sj} equal to 0 for each regular node j for which $\nexists k : j \in A_k$. For $k = 1, \ldots, K$, and for all regular nodes i such that $i \in A_k$, set $x_{i(N+k)}$ equal to 1 (recall that an arc exists between every such pair of nodes $(i, N + k)$ by definition of the 1-center and by construction of the network design problem). Set $x_{j(N+k)}$ equal to 0 for all regular nodes j such that $j \notin A_k$ (if an arc exists between node j and node $N + k$). For all arcs terminating at nodes $N + K + 1, \ldots, N + M$, set the flow x equal to 0. Note that flows for all arcs terminating in node sets N and M have now been assigned such that constraints 4d and 4e are satisfied. Furthermore, flow conservation (constraint 4c) is now satisfied for nodes $1, \ldots, N$: regular nodes that are assigned to a mobile backbone node have one unit of incoming flow and one unit of outgoing flow (since every regular node is assigned to at most one mobile backbone node by our assumption that the original solution was feasible), while regular nodes that are not assigned to a mobile backbone node have no incoming or outgoing flow.

Finally, consider the arcs connecting nodes $N + 1, \ldots, N + M$ to the sink. For $k = 1, \ldots, K$, the arc from node $N + k$ to node t has capacity c_k, since $y_k = 1$. The remaining arcs have zero capacity, since $y_k = 0$ for $k = K + 1, \ldots, M$. Set the flows $x_{(N+k)t}$ equal to $|A_k|$ for $k = 1, \ldots, K$, and set the flows $x_{(N+k)t}$ equal to 0 for $k = K + 1, \ldots, M$. By definition of c_k and by our assumption that all assigned regular nodes achieve throughput at least τ_{min} in the original solution, constraint (4f) is satisfied. Constraint (4d) is also now satisfied for all arcs. Furthermore, constraint (4c) is satisfied for nodes $N + 1, \ldots, N + M$; node $N + k$ has $|A_k|$ units of flow incoming and outgoing for $k = 1, \ldots, K$, and zero units of flow incoming and outgoing for $k = K + 1, \ldots, M$. Thus, all constraints are satisfied, and the objective value of this solution is equal to the number of regular nodes that were assigned in the original solution, i.e., $\sum_{k=1}^{K} |A_k|$.

Now, assume that we are given a feasible solution to the network design problem and its associated MILP. Furthermore, assume that all flows in this solution are integer. (By virtue of the total unimodularity of the constraint matrix in the maximum flow problem and the integrality of the right-hand side vector, all basic feasible solutions of the linear program induced by any feasible choice of the $\mathbf{y}$ vector are integer.)

Since all flows are integer, flows along arcs terminating in node sets N and M are either 0 or 1. Again, assume without loss of generality that the nodes are labeled such that $y_1, \ldots, y_K$ are equal to 1, and $y_{K+1}, \ldots, y_M$ are equal to 0. Thus, no flow traverses nodes $N + K + 1, \ldots, N + M$.

To construct a feasible solution to the mobile backbone network optimization problem, first place mobile backbone nodes at the K locations $l_1, \ldots, l_K$ corresponding to nodes $N + 1, \ldots, N + K$ in the network design graph. Next, for each mobile backbone node $k = 1, \ldots, K$, let A_k be the set of regular nodes for which $x_{i(N+k)} = 1$. Assign the regular nodes in set A_k to regular node k. Note that each regular node is assigned to at most one mobile backbone node, since at most one unit of flow can traverse each node $1, \ldots, N$. Furthermore, note that each assigned regular node achieves throughput at least τ_{min} by definition of the arc capacity c_k. The number of regular nodes assigned in this solution is equal to $\sum_{k=1}^{K} |A_k|$. For each mobile backbone node k, $|A_k|$ is equal to the flow traversing node $N + k$. Since no flow traverses nodes $N + K + 1, \ldots, N + M$, the total flow through the graph is equal to $\sum_{k=1}^{K} |A_k|$. Thus, the two solutions have the same objective value.

We have shown that for every feasible solution to the original problem, there is a feasible solution to the corresponding network design problem and its associated MILP with the same objective value. Thus, although the restriction of mobile backbone nodes to the 1-center locations of regular nodes may exclude an uncountable number of optimal solutions to the original placement and assignment problem, it does not exclude *all* optimal solutions. Furthermore, we have shown that for every integer solution to the network design problem and its associated MILP (including the optimal solution), there is a feasible solution to the original problem with the same objective value. Thus, the MILP formulation cannot produce a solution with a higher objective value than is possible in the original problem. Therefore, the MILP formulation can be used to obtain a solution to the original problem that achieves the same optimal objective value that was possible in the original problem.   □

APPENDIX B

PROOF OF LEMMA 1

Proof. We begin by restating the submodularity condition as follows:

$$f^*(S \cup \{i, j\}) + f^*(S) \leq f^*(S \cup \{i\}) + f^*(S \cup \{j\}), \quad (6)$$

where f^* is the maximum flow through G, as a function of the set of selected arcs. Next, we note that for a fixed
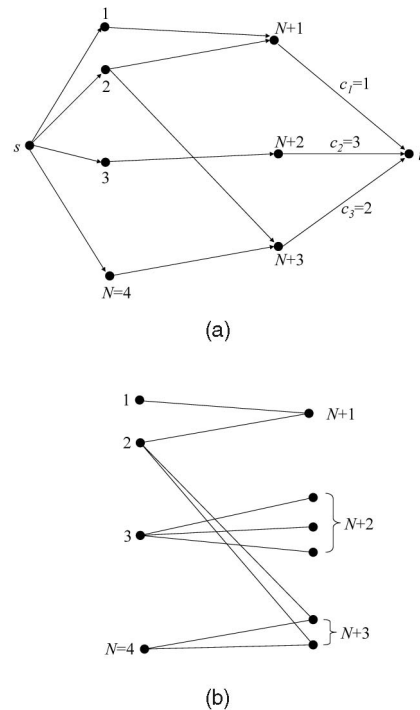

Fig. 7. An example of conversion from a maximum flow problem to an equivalent bipartite matching problem, for $N = 4, M = 3$. (a) A graph over which a maximum flow problem can be formulated. Unlabeled arc capacities are equal to one. (b) A bipartite matching problem that is equivalent to the maximum flow problem above.

selection of arcs S, the problem of finding the maximum flow through G can be expressed as an equivalent maximum matching problem on a bipartite graph with node sets L and R.[2] This is accomplished as follows: node set L in the bipartite matching problem is simply node set N in the maximum flow problem. Node set R is derived from node set M in the maximum flow problem, with one modification: if the arc from node $N + i \in$ M to t has outgoing capacity c_i, then R contains c_i copies of node $N + i$, each of which is connected to the same nodes in L as the original node $N + i$. Thus, each node $N + i$ in the maximum flow problem becomes a *set* of nodes $N + i$ in the bipartite matching problem, and the cardinality of this set is equal to c_i. An example of this reformulation is shown in Fig. 7.

For any feasible flow in the original graph, there is a corresponding matching in the bipartite graph with cardinality equal to the volume of flow; likewise, for any feasible matching in the bipartite graph, there is a corresponding flow of volume equal to the cardinality of the matching. Therefore, the volume of the maximum flow through the original graph is equal to the cardinality of a maximum matching in the bipartite graph.

The graphs expressing the relation in (6) are shown in Fig. 8a: the sum of the maximum flows through the left two graphs must be less than or equal to the sum of the maximum flows through the right two graphs.

2. A set of edges in a graph is a *matching* if no two edges share a common end node. A *maximum matching* is a matching of maximum cardinality [26].
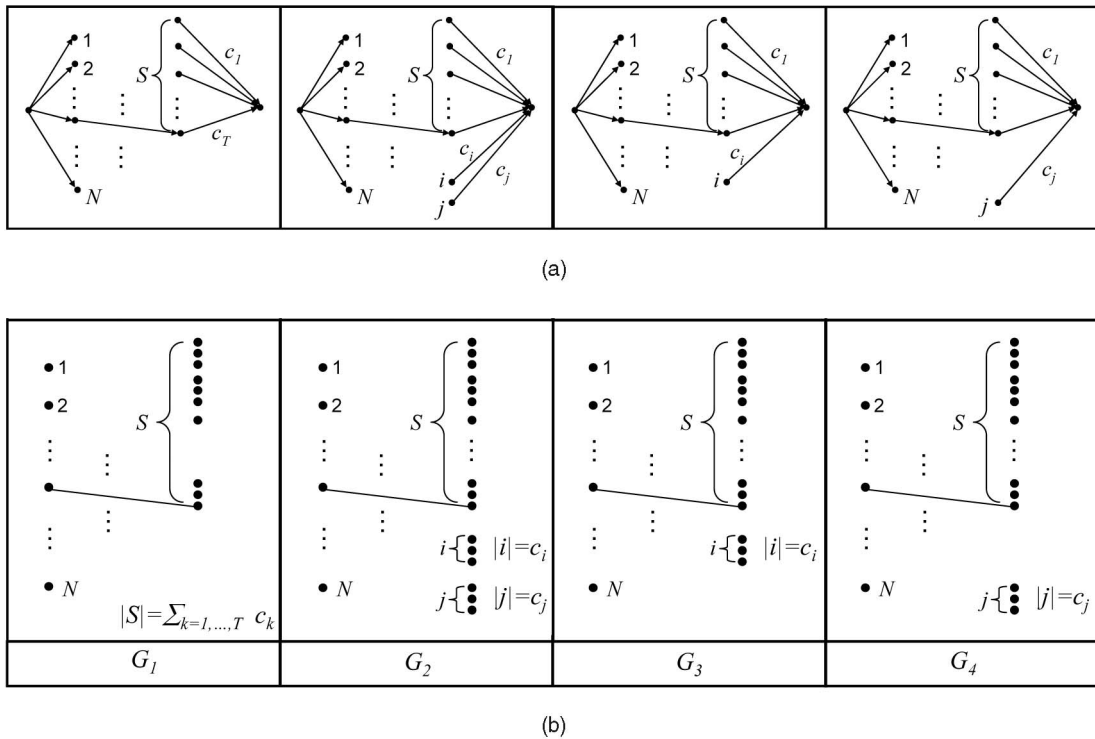
(a)

(b)

Fig. 8. Schematic representation of the graphs involved in the proof of the submodularity condition. The top graphs relate to the original maximum flow problem, while the bottom graphs are their equivalent reformulations in the bipartite matching problem. For clarity, not all arcs are shown.

Converting these maximum flow problems into their equivalent bipartite matching problems, we obtain the condition that the sum of the cardinalities of maximum matchings in bipartite graphs G_1 and G_2 in Fig. 8b is at most the sum of the cardinalities of maximum matchings in G_3 and G_4.

Consider a maximum matching M_1 in graph G_1, and denote its cardinality by N_s. This means that N_s nodes from set S are covered by matching M_1. Note that M_1 is a feasible matching for G_2 as well, since all arcs in G_1 are also present in G_2.

It is a property of bipartite graphs that if a matching Q is feasible for a graph H, then there exists a maximum matching Q^* in H such that all of the nodes covered by Q are also covered by Q^* [26]. Denote such a maximum matching for matching M_1 in graph G_2 by M_2, and note that N_s nodes from set S are covered by M_2. Denote the number of nodes covered by M_2 in node sets i and j by N_i and N_j, respectively. Then, the total cardinality of these maximum matchings for graphs G_1 and G_2 is equal to $2N_s + N_i + N_j$.

Now consider the matching obtained by removing the edges incident to node set j from M_2. Note that this matching is feasible for graph G_3, and its cardinality is $N_s + N_i$. Likewise, the matching obtained by removing the edges incident to node set i from M_2 is feasible for graph G_4, and its cardinality is $N_s + N_j$. Since these matchings are feasible (but not necessarily optimal) for G_3 and G_4, the sum of the cardinalities of maximum matchings for these graphs must be at least $2N_s + N_i + N_j$. This establishes the submodularity property for the matching problem as well as for the maximum flow problem. □

REFERENCES

[1] W. Burgardt, M. Moorstt, D. Fox, R. Simmons, and S. Thrun, "Collaborative Multi-Robot Exploration," *Proc. IEEE Int'l Conf. Robotics and Automation,* Apr. 2000.
[2] R. Balasubramanian, S. Ramasubramanian, and A. Efrat, "Coverage Time Characteristics in Sensor Networks," *Proc. IEEE Int'l Conf. Mobile Adhoc and Sensor Systems (MASS '06),* Oct. 2006.
[3] A. Srinivas and E. Modiano, "Joint Node Placement and Assignment for Throughput Optimization in Mobile Backbone Networks," *Proc. IEEE INFOCOM,* Apr. 2008.
[4] I. Rubin, A. Behzadm, R. Zhang, H. Luo, and E. Caballero, "TBONE: A Mobile-Backbone Protocol for Ad Hoc Wireless Networks," *Proc. IEEE Aerospace Conf.,* vol. 6, 2002.
[5] K. Xu, X. Hong, and M. Gerla, "Landmark Routing in Ad Hoc Networks with Mobile Backbones," *J. Parallel and Distributed Computing,* vol. 63, no. 2, pp. 110-122, 2003.
[6] A. Srinivas, G. Zussmann, and E. Modiano, "Construction and Maintenance of Wireless Mobile Backbone Networks," *IEEE/ACM Trans. Networking,* vol. 17, no. 1, pp. 239-252, Feb. 2009.
[7] B. Steckler, B. Bradford, and S. Urrea, "Hastily Formed Networks for Complex Humanitarian Disasters—After Action Report and Lessons Learned from the Naval Postgraduate School's Response to Hurricane Katrina," http://faculty.nps.edu/dl/HFN/documents/NPS_Katrina_AAR-LL_04-MAY-06.pdf, Sept. 2005.
[8] D. Jea, A.A. Somasundara, and M.B. Srivastava, "Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks," *Proc. IEEE/ACM Int'l Conf. Distributed Computing in Sensor Systems (DCOSS '05),* June 2008.
[9] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks," *Proc. IEEE Int'l Workshop Sensor Network Protocols and Applications (SNPA '03),* May 2003.

[10] M.M. Bin Tariq, M. Ammar, and E. Zegura, "Message Ferry Route Design for Sparse Ad Hoc Networks with Mobile Nodes," *Proc. ACM MobiHoc,* May 2006.

[11] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," *Proc. ACM MobiHoc,* May 2004.

[12] S.V. Hanly, "An Algorithm for Combined Cell-Site Selection and Power Control to Maximize Cellular Spread Spectrum Capacity," *IEEE J. Selected Areas in Comm.,* vol. 13, no. 7, pp. 1332-1340, Sept. 1995.

[13] G.J. Foschini and Z. Milzanic, "A Simple Distributed Autonomous Power Control Algorithm and Its Convergence," *IEEE Trans. Vehicular Technology,* vol. 42, no. 4, pp. 641-646, Nov. 1993.

[14] R. Yates and C.Y. Huang, "Integrated Power Control and Base Station Assignment," *IEEE Trans. Vehicular Technology,* vol. 44, no. 3, pp. 638-644, Aug. 1995.

[15] R. Mathar and T. Niessen, "Optimum Positioning of Base Stations for Cellular Radio Networks," *Wireless Networks,* vol. 6, no. 6, pp. 421-428, Dec. 2000.

[16] C. Glaßer, S. Reith, and H. Vollmer, "The Complexity of Base Station Positioning in Cellular Networks," *Discrete Applied Mathematics,* vol. 148, no. 1, pp. 1-12, 2005.

[17] E. Amaldi, A. Capone, and F. Malucelli, "Radio Planning and Coverage Optimization of 3G Cellular Networks," *Wireless Networks,* vol. 14, pp. 435-447, 2008.

[18] E. Craparo, J. How, and E. Modiano, "Optimization of Mobile Backbone Networks: Improved Algorithms and Approximation," *Proc. Am. Control Conf.,* June 2008.

[19] F. Preparta and M. Shamos, *Computational Geometry: An Introduction.* Springer-Verlag, 1985.

[20] P. Agarwal and M. Sharir, "Efficient Algorithms for Geometric Optimization," *ACM Computing Surveys,* vol. 30, pp. 412-458, 1998.

[21] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[22] G. Cornuèjols, "Valid Inequalities for Mixed Integer Linear Programs," *Math. Programming,* vol. 112, no. 1, pp. 3-44, 2006.

[23] E. Craparo, "Cooperative Exploration under Communication Constraints," doctoral thesis, Massachusetts Inst. of Technology, Sept. 2008.

[24] D. Bertsimas and R. Weismantel, *Optimization over Integers,* p. 88. Dynamic Ideas, 2005.

[25] G. Nemhauser and L. Wolsey, "Maximizing Submodular Set Functions: Formulations and Analysis of Algorithms," *Studies on Graphs and Discrete Programming,* P. Hansen, ed., pp. 279-301, North-Holland, 1981.

[26] A.S. Asratian, T.M.J. Denley, and R. Häggkvist, *Bipartite Graphs and Their Applications.* Cambridge Univ. Press, 1998.

Emily M. Craparo is a graduate of the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology (MIT) and received the SB, SM, and PhD degrees in aeronautics and astronautics in 2002, 2004, and 2008, respectively. Following a National Research Council postdoctoral fellowship, she joined the Department of Operations Research at the Naval Postgraduate Research School in Monterey, California. Her research interests include design and management of complex systems, particularly networked systems that operate in information-rich environments. She is a member of the IEEE.

Jonathan P. How received the BASc degree from the University of Toronto in 1987 and the SM and PhD degrees in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT) in 1990 and 1993, respectively. He then studied for two years at MIT as a postdoctoral associate for the Middeck Active Control Experiment (MACE) that flew on-board the space shuttle Endeavour in March 1995. He is currently the Richard Cockburn Maclaurin professor of aeronautics and astronautics in the Department of Aeronautics and Astronautics at MIT. Prior to joining MIT in 2000, he was an assistant professor in the Department of Aeronautics and Astronautics at Stanford University. He was the planning and control lead for the MIT DARPA Urban Challenge team that placed fourth in the 2007 race at Victorville, California. His current research interests include robust coordination and control of autonomous vehicles in dynamic uncertain environments. Also, he is currently an associate editor for the *IEEE Control Systems Magazine* and has also served as an associate editor for the *AIAA Journal of Guidance, Control, and Dynamics.* He was the recipient of the Institute of Navigation Burka Award in 2002 and the Boeing Special Invention Award in 2008. He is the Raymond L. Bisplinghoff fellow for the MIT Aeronautics/Astronautics Department (2006-2008), an associate fellow of the AIAA, and a senior member of the IEEE.

Eytan Modiano received the BS degree in electrical engineering and computer science from the University of Connecticut at Storrs in 1986, and the MS and PhD degrees, both in electrical engineering, from the University of Maryland at College Park in 1989 and 1992, respectively. Since 1999, he has been on the faculty at the Massachusetts Institute of Technology (MIT), where he is currently an associate professor. He was a US Naval Research Laboratory fellow between 1987 and 1992 and a National Research Council postdoctoral fellow during 1992-1993. Between 1993 and 1999, he was with the MIT Lincoln Laboratory where he was the project leader for MIT Lincoln Laboratory's Next Generation Internet (NGI) project. His research interest is in communication networks and protocols with emphasis on satellite, wireless, and optical networks. He was the corecipient of the Sigmetrics 2006 Best Paper Award for the paper "Maximizing Throughput in Wireless Networks via Gossiping" and the WiOpt 2005 Best Student Paper Award for the paper "Minimum Energy Transmission Scheduling Subject to Deadline Constraints." Also, he is currently an associate editor for the *IEEE/ACM Transactions on Networking* and has served as an associate editor for the *IEEE Transactions on Information Theory.* He has served as a guest editor for the *IEEE JSAC* special issue on WDM network architectures, the *Computer Networks* journal special issue on broadband Internet access, the *Journal of Communications and Networks* special issue on wireless ad-hoc networks, and for the *IEEE Journal of Lightwave Technology* special issue on optical networks. He was the technical program cochair for IEEE WiOpt 2006, IEEE INFOCOM 2007, and ACM MobiHoc 2007. He is a senior member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.