Optimization of Mobile Backbone Networks: Improved Algorithms and Approximation

Emily M. Craparo, Jonathan P. How, and Eytan Modiano

***Abstract*— This paper presents new algorithms for throughput optimization in mobile backbone networks. This hierarchical sensing approach combines mobile backbone nodes, which have superior mobility and communication capability, with regular nodes, which are constrained in mobility and communication capability but which can sense the environment. 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 optimizing this quantity 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 approximation algorithm has a theoretical performance guarantee and is demonstrated to perform well in practice.**

I. 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 [6], and multiple sensors can provide simultaneous coverage of a relatively large area for an extended period of time [4]. However, in many applications the data collected by these distributed platforms is best utilized after it has been aggregated. This requires communication among the robotic or sensing agents. This paper focuses on a particular hierarchical network architecture called a *mobile backbone network*, in which some agents are dedicated to providing communication support for other agents in the form of a *backbone* over which end-to-end communication can take place. Mobile backbone networks can be used to model a variety of multi-agent systems. 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.

E. Craparo is a Research Assistant in the Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, emilyc@mit.edu

J. How is a Professor in the Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, jhow@mit.edu

E. Modiano is an Associate Professor in Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, modiano@mit.edu

Previous work has focused on optimal placement of mobile backbone nodes in networks of fixed regular nodes [11]. Existing techniques, while exact, suffer from intractable computation times, even for problems of modest size. This paper describes more tractable solutions to previously posed problems, including an improved exact algorithm and the first known approximation algorithm with computation time that is polynomial in the number of regular nodes and mobile backbone nodes.

A. Previous work in mobile backbone networks

Mobile backbone networks were described by Rubin et al. [10] and Xu et al. [13]. Srinivas et al. [12] defined two types of nodes, which may be thought of as representing robotic agents: 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. [12] 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 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 described by Srinivas and Modiano [11]. In this model, the throughput (data rate) that can be achieved between a regular node and a mobile backbone node is modeled as a decreasing 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. For example, as described in Ref. [11], the throughput τ between regular node i and mobile backbone node j when using a Slotted Aloha communication protocol can be approximated by

$$\tau(i,j) \approx \frac{1}{e \cdot |A(j)| \cdot d(i,j)^{\alpha}}, \qquad (1)$$

where e is the base of the natural logarithm, $|A(j)|$ is the number of regular nodes assigned to mobile backbone node j, $d(i,j)$ is the distance between regular node i and mobile backbone node j, and α is the path loss exponent. Each regular node is assigned to a single mobile backbone node. An implicit assumption is made that regular nodes assigned to one mobile backbone node encounter no interference from regular nodes assigned to other mobile backbone nodes (for example, because each "cluster" composed of a mobile backbone node and its assigned regular nodes operates at a different frequency than other clusters). It is also assumed that the mobile backbone nodes can communicate effectively with each other over the entire problem domain, so that there is no additional constraint that the mobile backbone nodes need to be "connected" to one another.

Building upon this throughput model, we pose the mobile backbone network optimization problem as follows: our goal is to *place* a number of 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 number of regular nodes that achieve a given minimum throughput level is maximized. This is a sub-problem of the *maximum fair placement and assignment* (MFPA) problem considered in Ref. [11], in which the objective is to maximum the minimum throughput achieved by any regular node, such that all regular nodes are assigned. Thus, an improved algorithm for maximizing the number of regular nodes assigned at a given throughput level also yields an improved algorithm for the MFPA problem.

A typical example of a solution to this problem is shown in Fig. 1 for a group of regular nodes denoted by $\circ$. The mobile backbone nodes, denoted by $\star$, have been placed such that they maximize the number of regular nodes that achieve the given minimum throughput level. This example is typical in that the clusters of regular nodes and mobile backbone nodes are relatively small, and the regular nodes are distributed intelligently among the mobile backbone nodes, with fewer regular nodes being allocated to the mobile backbone nodes with larger cluster radii. In this case, all regular nodes have been successfully assigned to mobile backbone nodes.

A key insight discussed in Ref. [11] is that although the mobile backbone nodes can occupy arbitrary locations, they can be restricted to a relatively small number of locations without sacrificing optimality. Specifically, each mobile backbone node can be placed at the 1-center of its set of assigned regular nodes in an optimal solution. The 1-center location of a set of regular nodes is the location that minimizes the maximum distance from the mobile backbone node to any regular node, and it is easily computable [1]. Fortunately, although there are 2^N possible subsets of N regular nodes, there are only $O(N^3)$ distinct 1-center locations [9]. Leveraging the fact that there are a limited number of possible mobile backbone node locations (polynomially many in N), Ref. [11] solves the MFPA problem by performing an exhaustive search over all possible placements of K mobile backbone nodes for each possible value of the minimum throughput, determining the optimal assignment for each placement by solving an integer network flow problem. The computation time of this search-based algorithm is thus polynomial in the number of regular nodes, but exponential in the number of mobile backbone nodes.

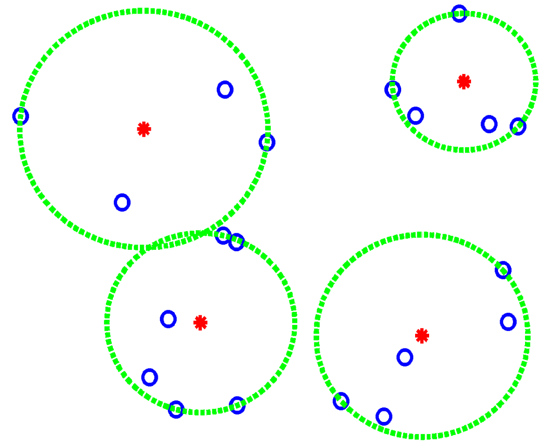


Fig. 1. A typical example of an optimal mobile backbone network. Mobile backbone nodes, indicated by $*$, are placed such that they provide communication support for regular nodes, shown as $\circ$. Each regular node is assigned to one mobile backbone node. Dashed lines indicate the radius of each cluster of nodes.

II. Mixed-Integer Linear Programming Approach To Simultaneous Placement And Assignment

A primary contribution of this paper is the development of a single optimization problem that simultaneously solves the mobile backbone node placement and regular node assignment problems, thus eliminating the need for an exhaustive search over possible mobile backbone node placements. 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 achieve some desired flow characteristics at a minimum cost by intelligently "purchasing" a subset of these arcs [2].

The network design problem that produces an optimal placement and assignment is constructed as follows. A source node, s, is connected to each node i in the set of nodes N$= \{1,\ldots,N\}$ (see Fig. 2). These nodes represent regular node locations. 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 $O(N^3)$. Node $i \in$ N is connected to node $N+j \in$ M iff regular node i is within the radius of 1-center location j, and 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. For example, for the approximate Slotted Aloha throughput function described by Eq. (1),

$$c_i = \left\lfloor \frac{1}{e \cdot \tau \cdot r_i^{\alpha}} \right\rfloor, \tag{2}$$

where τ is the desired minimum throughput and r_i is the radius associated with 1-center location i. This means that if

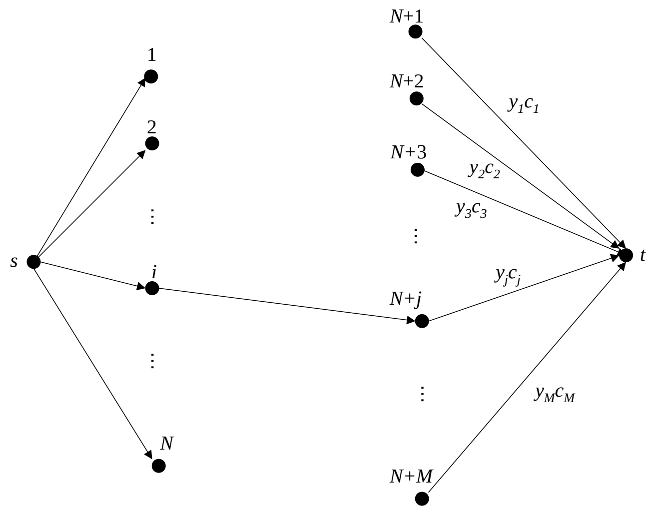


Fig. 2. The network design problem corresponding to the joint placement and assignment problem for mobile backbone networks. Unlabeled arc capacities are equal to one.

at most c_i regular nodes are assigned to the mobile backbone node at location i, each of these regular nodes will achieve throughput at least τ. Denote the set of nodes for this network design problem by $\mathscr{N}$ and the set of arcs by $\mathscr{A}$.

If K mobile backbone nodes are available to provide communication support for N regular nodes at given locations, and a throughput level 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. This network design problem can be solved via the following mixed-integer linear program (MILP):

$$\max_{\mathbf{x},\mathbf{y}} \sum_{i=1}^{N} x_{si} \tag{3a}$$

$$\text{subject to} \sum_{i=1}^{M} y_i \leq K \tag{3b}$$

$$\sum_{j:(i,j)\in\mathscr{A}} x_{ij} = \sum_{l:(l,i)\in\mathscr{A}} x_{li} \qquad i \in \mathscr{N}\setminus\{s,t\} \tag{3c}$$

$$x_{ij} \geq 0 \qquad \forall\ (i,j)\in\mathscr{A} \tag{3d}$$

$$x_{ij} \leq 1 \qquad \forall\ (i,j)\in\mathscr{A} : j\in\mathscr{N}\setminus\{t\} \tag{3e}$$

$$x_{(N+i)t} \leq y_i c_i \qquad \forall\ i\in\{1,\dots,M\} \tag{3f}$$

$$y_i \in \{0,1\} \qquad \forall\ i\in\{1,\dots,M\}. \tag{3g}$$

The objective of this problem is to maximize the flow $\mathbf{x}$ through the graph (Eq. (3a)). The constraints state that at most K arcs (mobile backbone node locations) can be selected (3b), flow through all internal nodes must be conserved (3c), arc capacities must be observed (3d - 3f), and y_i is binary for all i (3g). Note that, for a given specification of the $\mathbf{y}$ vector, all flows $\mathbf{x}$ are integer in all basic feasible solutions of the resulting (linear) maximum flow problem.

A solution to problem (3) 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 $x_{i(N+j)} = 1$.

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 above MILP problem needs only to be solved once for the desired throughput value and with a fixed value of K. To solve the MFPA problem, which is the primary problem of interest in Ref. [11], it is necessary to solve the above MILP problem multiple 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 minimum throughput values; searching among these values using a binary search would require $O(\log(N^4))$ solutions of the MILP problem.

Remark 2: It should be noted that the worst-case complexity of mixed-integer linear programming is exponential in the number of binary variables. However, this approach performs well in practice, and simulation results indicate that it compares very favorably with the approach developed in Ref. [11] for cases of interest (See Table I). 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.

TABLE I

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, USING ILOG CPLEX 9.020.

N	K	MILP Algorithm	Search-based Approach
3	2	3 sec	20 sec
4	2	4 sec	81 sec
5	2	5 sec	202 sec
6	2	6 sec	507 sec
6	3	6 sec	$>$ 30 min
8	3	8.5 sec	$>$ 30 min
10	3	9 sec	$>$ 30 min
15	5	18 sec	$>$ 30 min
20	5	47 sec	$>$ 30 min
25	5	196 sec	$>$ 30 min

III. Approximation Algorithm

The MILP formulation of the previous section provides an optimal solution in tractable time for moderately-sized problems. For large-scale problems, 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 that leads to the approximation algorithm 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 $D = \{1,\dots,d\}$, 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 $i,\ j \in D$, $i \neq j$ and $S \subset D \setminus \{i,j\}$ [5]. In the context of our 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 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 [8]. 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 $O(N^3 K)$ 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.

We now prove that the objective function in the problem under consideration is submodular.

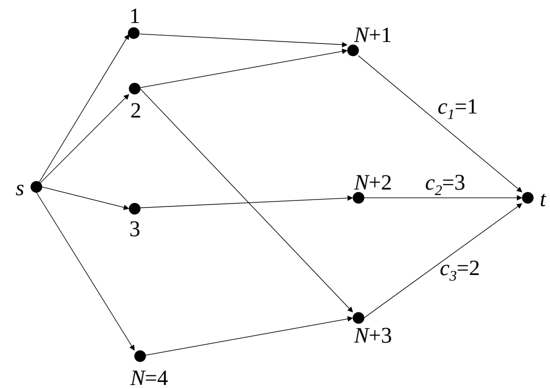
Theorem 1
If G is a graph in the network design problem described in Section II, the maximum flow that can be routed through G is a submodular function of the set of arcs selected.

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

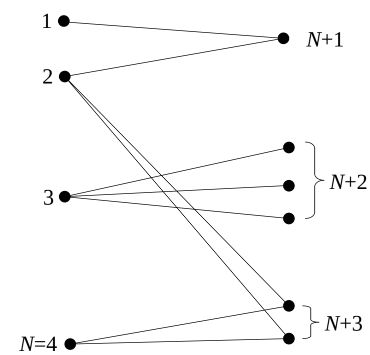
$$f^*(S) + f^*(S \cup \{i,j\}) \le f^*(S \cup \{i\}) + f^*(S \cup \{j\}) \qquad (4)$$

where f^* is the maximum flow through G, as a function of the set of selected arcs. Next, we note that for a fixed selection of arcs S, the problem of finding the maximum flow through G can be expressed as an equivalent matching problem on a bipartite graph with node sets L and R[1]. 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. 3.

[1]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 [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.

Fig. 3. An example of conversion from a maximum flow problem to an equivalent bipartite matching problem, for $N = 4$, $M = 3$.

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 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 Eq. (4) are shown in the top row of Fig. 4: 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.

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. 4 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^* [3]. Denote such a maximum matching for

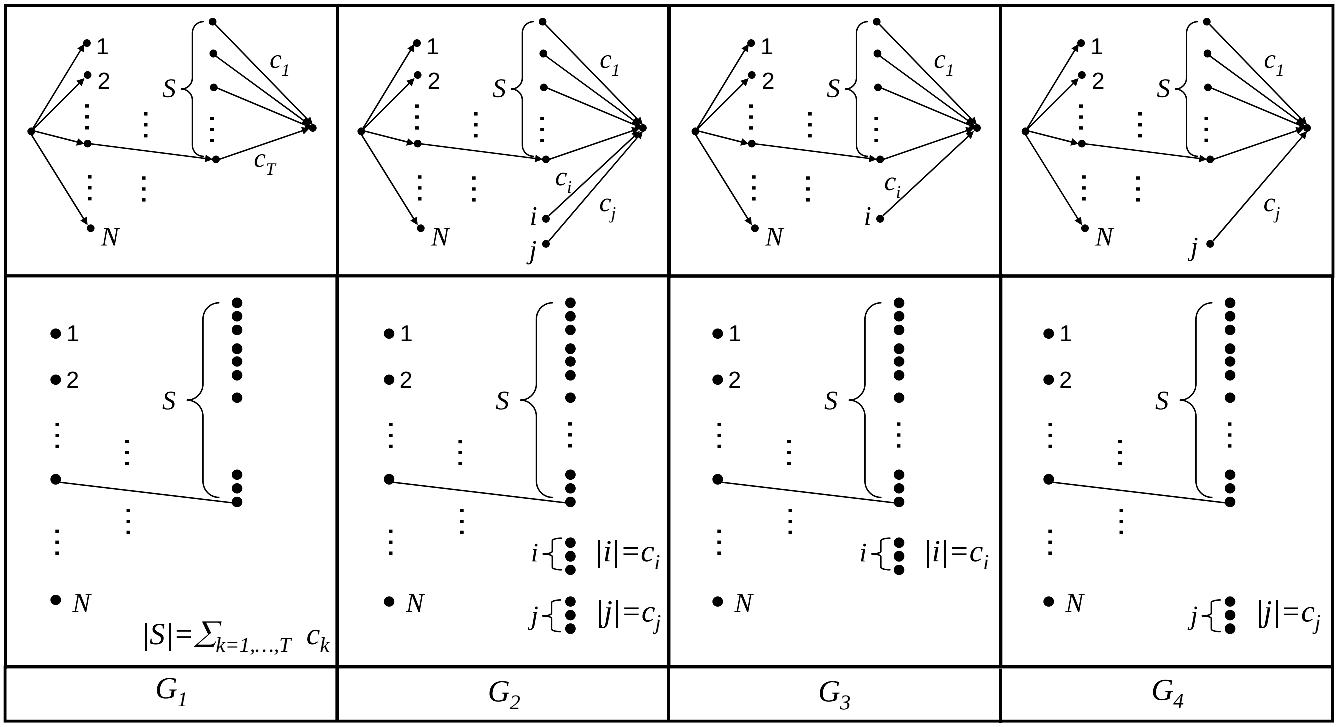


Fig. 4. Schematic representation of the graphs involved in the proof of the submodularity condition. The top graphs relate to the original maximum flow problem, while the bottom graphs are their equivalent reformulations in the bipartite matching problem. For clarity, not all arcs are shown.

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. ■

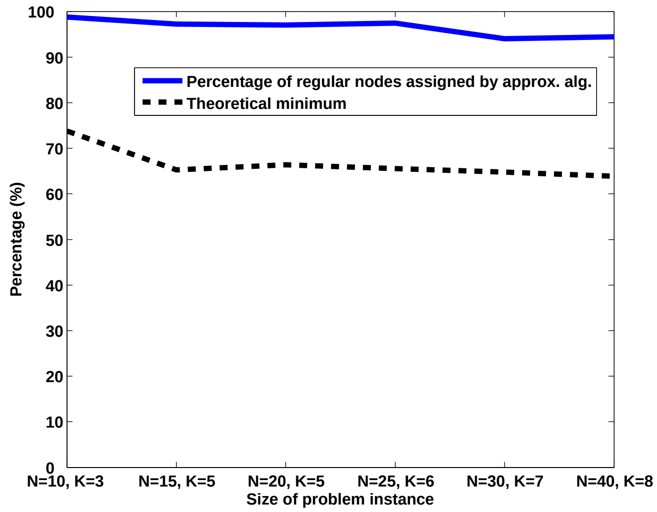
IV. Experimental Evaluation of the Approximation Algorithm

As described in the previous section, 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 II) [8]. However, this observation is based on a general result for nondecreasing submodular functions and not for the specific problem under consideration in this paper. Therefore, it is of interest to experimentally examine the performance of the greedy algorithm for our problem of interest.

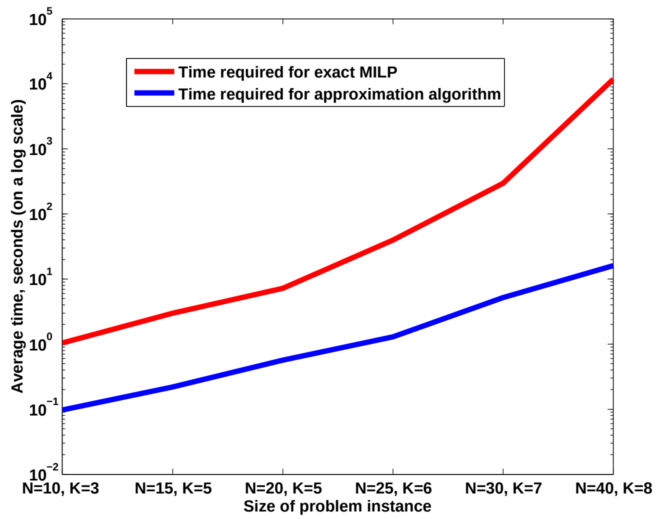
To this end, we have performed computational experiments on a number of problems of various sizes. Regular node locations were generated randomly in a finite 2-dimensional area, and a moderate throughput value was specified (i.e., one high enough that there was no trivial selection of mobile backbone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

Fig. 5 shows the performance of the approximation algorithm relative to the exact (MILP) algorithm. In Fig. 5(a), 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% 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. 5(b) 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 10 seconds for $N = 40$ and $K = 8$, whereas the MILP



(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.

Fig. 5. Comparison of the exact and approximation algorithms developed in this paper. Although the MILP-based exact algorithm developed in this paper 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.

algorithm took nearly three hours to solve a problem of this size. Both the MILP algorithm and the approximation algorithm were implemented using ILOG CPLEX 9.020.

V. Conclusions and Future Work

This work has developed new algorithms for solving the problem of mobile backbone network optimization. Both an exact MILP-based technique and the first known approximation algorithm 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, as well as to a related problem 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.

For cases in which a MILP approach is impractical due to constraints on computation time, the greedy approximation algorithm developed in this paper presents a viable alternative. This algorithm carries the benefit of a theoretical performance guarantee, and simulation results indicate that it performs very well for the problem of assigning a maximum number of regular nodes such that each assigned regular node achieves a minimum throughput level.

Future work will explore the use of mobile backbone network optimization methods in the context of cooperative exploration [7].

VI. Acknowledgments

This work was funded in part by AFOSR grant FA9550-04-1-0458, NSF grant CCR-0325401, and an NSF Graduate Fellowship. The authors thank the anonymous reviewers for their input and advice.

References

[1] P. Agarwal and M. Sharir, "Efficient Algorithms for Geometric Optimization," ACM Comput. Surveys, 30, 1998, pp. 412-458.

[2] A. Ahuja, T. Magnanti and J. Orlin, *Network Flows:Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[3] A. S. Asratian, T. M. J. Denley and R. Häggkvist, *Bipartite Graphs and their Applications*. Cambridge University Press, 1998.

[4] R. Balasubramanian, S. Ramasubramanian and A. Efrat, "Coverage Time Characteristics in Sensor Networks," in *2006 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS 2006)*, October 2006.

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

[6] W. Burgardt, M. Moorstt, D. Fox, R. Simmons and S. Thrun, "Collaborative Multi-Robot Exploration," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 2000.

[7] E. Craparo, J. How and E. Modiano, "Simultaneous Placement and Assignment for Exploration in Mobile Backbone Networks," submitted to the IEEE Conference on Decision and Control, 2008.

[8] G. Nemhauser and L. Wolsey, "Maximizing submodular set functions: Formulations and Analysis of Algorithms," *Studies on Graphs and Discrete Programming*, P. Hansen, ed., 1981, pp. 279-301.

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

[10] I. Rubin, A. Behzadm R. Zhang, H. Luo, and E. Caballero, "TBONE: a Mobile-Backbone Protocol for Ad Hoc Wireless Networks," *Proc. IEEE Aerospace Conference*, 6, 2002.

[11] A. Srinivas and E. Modiano, "Joint node placement and assignment for throughput optimization in mobile backbone networks," To appear in *Proc. IEEE INFOCOM'08*, Apr. 2008.

[12] A. Srinivas, G. Zussmanm and E. Modiano, "Mobile Backbone Networks: Construction and Maintenance," *ACM MOBIHOC 2006*, May 2006.

[13] K. Xu, X. Hong, and M. Gerla, "Landmark Routing in Ad Hoc Networks with Mobile Backbones," *Journal of Parallel and Distributed Computing*, 63, 2, 2003, pp. 110-122.