Constrained Node Placement and Assignment in Mobile Backbone Networks

Emily M. Craparo

Abstract This chapter describes new algorithms for mobile backbone network optimization. In this hierarchical communication framework, *mobile backbone nodes* (MBNs) are deployed to provide communication support for *regular nodes* (RNs). While previous work has assumed that MBNs are unconstrained in position, this work models constraints in MBN location. This chapter develops an exact technique for maximizing the number of RNs that achieve a threshold throughput level, as well as a polynomial-time approximation algorithm for this problem. We show that the approximation algorithm carries a performance guarantee of $\frac{1}{2}$ and demonstrated that this guarantee is tight in some problem instances.

1 Introduction and Background

Data collected by distributed sensor networks often must be collected or aggregated in a central location. The mobile backbone network architecture has been proposed to alleviate scalability problems in ad hoc wireless networks [1, 2], which can hinder the deployment of large-scale distributed sensing platforms. Noting that most communication capacity in large-scale single-layer mobile networks is dedicated to packet-forwarding and routing overhead, Xu et al. propose a multi-layer hierarchical network architecture and demonstrate the improved scalability of a two-layer framework [2]. Srinivas et al. [3] define two types of nodes: regular nodes (RNs), which have restricted mobility and limited communication capability, and mobile backbone nodes (MBNs), which have superior communication capability and which can be deployed to provide communication support for the RNs.

E.M. Craparo (✉)
Department of Operations Research, Naval Postgraduate School, Monterey, CA, USA
e-mail: emcrapar@nps.edu

V.L. Boginski et al. (eds.), *Sensors: Theory, Algorithms, and Applications*,
Springer Optimization and Its Applications 61, DOI 10.1007/978-0-387-88619-0_2,

© Springer Science+Business Media, LLC 2012

In addition to scaling well with network size, the mobile backbone network architecture naturally models a variety of real-world systems, such as airborne communication hubs that are deployed to provide communication support for ground platforms, or mobile agents that are positioned to collect data from stationary sensor nodes.

Srinivas et al. [4] and Craparo et al. [5] address problems involving simultaneous MBN placement and RN assignment. Both [4] and [5] seek to simultaneously *place* K MBNs, which can occupy any location in the plane, and *assign* N RNs to the MBNs, in order to optimize a various throughput characteristics of the network. Srinivas et al. describe an enumeration-based exact algorithm and several heuristics for maximizing the minimum throughput achieved by any RN [4]. Craparo et al. study the problem of maximizing the number of RNs that achieve a threshold throughput level $\tau_{\min}$; they propose an exact algorithm based on mixed-integer linear programming, as well as a polynomial-time approximation algorithm with a constant-factor performance guarantee [5].

A key feature of the formulations in [4] and [5] concerns the potential locations of the MBNs. Although the MBNs can feasibly occupy any locations in the plane, [4] and [5] demonstrate that the MBNs can be restricted to a relatively small set of locations ($O(N^3)$) without compromising the optimality of the overall solution. In particular, each MBN can be placed at the *1-center* of its assigned RNs. (A MBN is located at the *1-center* of a set of RNs if the maximum distance from the MBN to the any of the RNs in the set is minimized.) Additionally, each 1-center location l is associated with a unique radius of communication. This radius is the maximum possible distance between the MBN at location l and any of the RNs in subsets for which l is a 1-center [5]. Thus, the restriction of MBNs to 1-center locations not only dramatically reduces the size of the feasible set of MBN locations, but also removes the communication radius as an independent decision variable in the optimization problem.

In the formulations of [4] and [5], it is always possible to place MBNs in 1-center locations because the MBNs are assumed to be capable of occupying *any* location. In some applications, this assumption is valid. For instance, an airborne communication hub (e.g., a blimp) could easily be placed at the 1-center of its assigned RNs. In other applications, however, the potential locations of the MBNs may be limited. In hastily-formed networks operating in disaster areas, for instance, ground-based communication hubs are generally restricted to public spaces such as schools, hospitals, and police stations [6]. In this case, the mobile backbone network optimization problem is *constrained*, in the sense that the MBNs can occupy only a discrete set of locations, and these potential locations are given as input data. In this application, it is generally impossible to place each MBN at the 1-center of its assigned RNs. Although the restriction of MBNs to a finite set of locations can reduce the size of the solution space with respect to MBN placement, the maximum communication radius of each MBN is a separate decision variable in this case, and the formulations of [4] and [5] are inappropriate. This work describes a mobile backbone network optimization problem with MBN placement constraints and provides exact and approximation algorithms for solving this problem, along with full proofs of results as previously described in [7].

2 Problem Statement

We use the communication model of [4] and [5], in which the throughput τ that can be achieved between a RN n and a MBN k, is a *monotonically nonincreasing* function of two quantities: the *distance* between n and k, and the *number* of RNs that are assigned to k (and thus interfere with n's transmissions). We assume that each RNs are assigned to one MBN encounter, no interference from RNs assigned to other MBNs (for example, because each "cluster" consisting of an MBN and its assigned RNs operates on a dedicated frequency).

Under such a throughput model, we pose the *constrained placement and assignment* (CPA) problem as follows: given a set of N RNs distributed in a plane, *place* K MBNs in the plane while simultaneously *assigning* the RNs to the MBNs, such that the number of RNs that achieve throughput at least $\tau_{\min}$ is maximized. MBNs can occupy locations from the set $\mathrm{L} = \{1, \ldots, L\}$, $L \geq K$, and each RN can be assigned to at most one MBN.

We do not require the MBNs to be "connected" to one another; this model is appropriate for applications in which MBNs serve to provide a satellite uplink for RNs, such as in the hastily-formed networks as mentioned in Sect. 1. It is also appropriate for applications in which the MBNs are powerful enough to communicate effectively with one another over the entire problem domain. We also assume that the positions of RNs are known exactly, through the use of GPS, for example.

Problem CPA is similar to the message ferrying problem, in which RNs have a finite amount of data available to transmit, and MBNs must efficiently collect this data [8–11]. CPA differs in that as it does not assume that the RNs have a limited amount of data to transmit; rather, CPA seeks to provide throughput on a permanent basis. In this sense, CPA is similar to a facility location problem. However, whereas CPA seeks to efficiently utilize a limited resource (the MBNs), most facility location problems focus on servicing all customers at minimum cost. Additionally, the throughput model in this work does not correspond to a notion of "service" in any known facility location problem. CPA is also similar to cellular network optimization; however, most approaches to cellular network optimization involve decomposition of the problem. Some formulations take base station placement as an input and optimize over user assignment and transmission power, with the objective of minimizing total interference [12–15]. Others use a simple heuristic for the assignment of users to base stations and to focus on selection of base station locations [16, 17]. In contrast, CPA seeks to optimize the network *simultaneously* over MBN placement and RN assignment, without assuming that RNs have variable transmission power capabilities.

3 Network Design Formulation

A key insight concerning the structure of the throughput function facilitates solution of CPA. Consider a cluster of nodes consisting of an MBN and its assigned RNs. Note that if the RN that is farthest away from the MBN achieves throughput of at least $\tau_{\min}$, then all other RNs in the cluster also achieve throughput of at least $\tau_{\min}$. Thus, in order to guarantee that all regular nodes in a cluster achieve adequate throughput, we need only to ensure that the most distant RN in the cluster achieves throughput of at least $\tau_{\min}$ [5].

Leveraging this insight, we can obtain an optimal solution to the simultaneous MBN placement and RN assignment problem via a *network design* formulation. In network design problems, a given network can be augmented with additional arcs for a given cost, and the objective is to "purchase" a set of augmenting arcs, subject to a budget constraint, in order to optimize flow in some way [18]. The formulation of the network design problem used in this work is similar to that presented in [5], in that the geometry and throughput characteristics of the problem are captured in the structure of the network design graph. Relative to the formulation in [5], however, we must use additional constraints in the network design problem. These constraints account for the fact that the communication radius of each MBN is an independent decision variable, i.e., it is not uniquely determined by the selection of the MBN location.

Our network design problem is formulated on a graph $G = (\mathcal{N}, \mathcal{A})$ of the form as shown schematically in Fig. 1. The graph G is constructed as follows:

The nodes of G consist of a source s, a sink t, and two node sets, $\text{N} = \{n_1, \ldots, n_N\}$ and $\text{M} = \{m_1^1, \ldots, m_L^N\}$. N represents the RNs, while M represents possible combinations of MBN locations and communication radii; node m_l^n represents the MBN at location l and that communicates with RNs within radius r_l^n of l, where r_l^n is the distance from location l to RN n. The source s is connected to each of the nodes in N via an arc of unit capacity. For each RN i, candidate MBN location l, and communication radius r_l^n, n_i is connected to node m_l^n if and only if $r_l^i \leq r_l^n$. All of the arcs connecting nodes in N to nodes in M have unit capacity. Finally, each node in M is connected to the sink, t. The capacity of the arc connecting node m_l^n to t is the product of a binary variable y_l^n and a constant c_l^n. The binary variable y_l^n represents the decision of whether to place the MBN at location l with maximum communication radius r_l^n. The constant c_l^n is the maximum number of RNs that can be assigned to the MBN at location l such that an RN at a distance r_l^n from l achieves throughput of at least $\tau_{\min}$. This quantity can be computed by means of a given throughput function, τ, and a desired minimum throughput level, $\tau_{\min}$. For an invertible throughput function, one can take the inverse of the function with respect to cluster size, evaluate the inverse at the desired minimum throughput level $\tau_{\min}$, and take the floor of the result to obtain an integer value for c_l^n. If the throughput function cannot easily be inverted with respect to cluster size, one can perform a search for the largest cluster size $c_l^n \leq N$ such that $\tau(c_l^n, r_l^n) \geq \tau_{\min}$. A binary search for c_l^n would involve $O(\log(N))$ evaluations of the function τ for each radius.

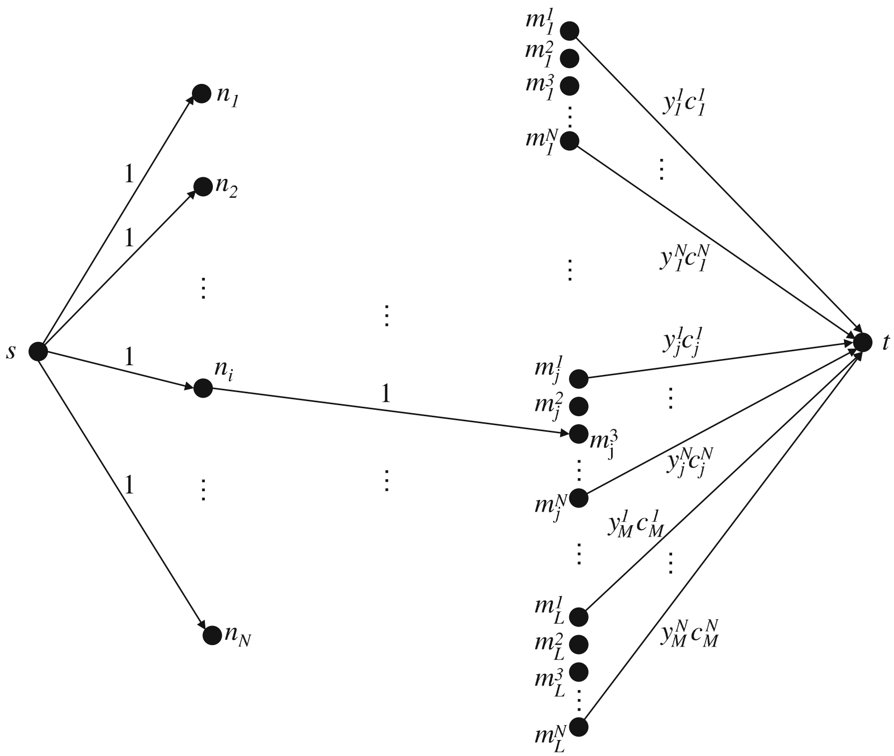

Fig. 1 Schematic representation of the graph on which an instance of the network design problem is posed

The objective of the network design problem is to "activate" a subset of the arcs entering t in such a way as to maximize the volume of flow that can travel from s to t. In addition to the capacity and flow conservation constraints typical of network models, the network design problem also includes cardinality and multiple-choice constraints. The cardinality constraint states that exactly K arcs are to be activated, reflecting the fact that K MBNs are available for placement. The multiple-choice constraints state that at most one arc with subscript l can be activated for each $l = 1, \ldots, L$. These constraints allow at most one MBN to be placed at each location; in other words, the locations $1, \ldots, L$ represent item classes, while the possible radii $r_l^1, \ldots, r_l^N$ represent items within each class, and the multiple-choice constraints state that at most one item can be selected from each class.

We denote the network design problem on G as the Multiple-Choice Network Design (MCND) problem. MCND can be solved via the following mixed-integer linear program (MILP):

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

$$\text{subject to} \quad \sum_{l=1}^{L}\sum_{n=1}^{N} y_l^n = K \tag{1b}$$

$$\sum_{n=1}^{N} y_l^n \leq 1 \qquad \forall\ l = 1,\ldots,L \tag{1c}$$

$$\sum_{i:(i,j)\in\mathcal{A}} x_{ij} = \sum_{k:(j,k)\in\mathcal{A}} x_{jk} \qquad j \in \mathcal{N}\setminus\{s,t\} \tag{1d}$$

$$x_{ij} \geq 0 \qquad \forall\ (i,j)\in\mathcal{A} \tag{1e}$$

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

$$x_{m_l^n t} \leq y_l^n c_l^n \qquad \forall\ l,n \tag{1g}$$

$$x_{n_i m_l^n} \leq y_l^n \qquad \forall\ i,l,n \tag{1h}$$

$$y_l^n \in \{0,1\} \qquad \forall\ l,n. \tag{1i}$$

The objective of MCND is to maximize the flow of $\mathbf{x}$ that traverses G, which corresponds to the total number of RNs that can be assigned at throughput $\tau_{\min}$. Constraint (1b) states that K arcs (MBN locations) are to be selected, and constraint (1c) states that at most one MBN can be placed at each location. Constraints (1d)–(1g) are network flow constraints, stating that flow through all internal nodes must be conserved (1d) and that arc capacities must be observed (1e)–(1g). Constraint (1h) is a valid inequality that improves computational performance by reducing the size of the feasible set in the LP relaxation. Constraint (1i) ensures that y_l^n is binary for all l, n. Note that, for a given specification of the $\mathbf{y}$ vector, all flows of $\mathbf{x}$ are integer in all basic feasible solutions of the resulting linear network flow problem.

An optimal solution to a instance of MCND provides both the placement of MBNs and the assignment of RNs to MBNs. The MBN is placed at location l if $y_l^n = 1$ for some n. RN i is assigned to the MBN at location l if and only if the flow from node n_i to node m_l^j is equal to 1 for some j. The equivalence between MCND and the original problem CPA is more formally stated in Theorem 1.

Theorem 1 *Given an instance of CPA, the solution to the corresponding instance of MCND yields an optimal MBN placement and RN assignment.*

Proof. The proof of Theorem 1 appears in Appendix 1. □

3.1 Hardness of Network Optimization

Although an optimal solution to MCND provides an optimal solution to the corresponding instance of CPA, the MILP approach described above is not computationally tractable from a theoretical perspective. This fact motivates consideration of the fundamental tractability of CPA itself. If CPA is NP-hard, it may be difficult or impossible to find an exact algorithm that is significantly more efficient than the MILP approach. Unfortunately, CPA is indeed NP-hard.

Theorem 2 *Problem CPA is NP-hard.*

Proof. The proof of Theorem 2 appears in Appendix 2. □

4 Approximation Algorithm

The probable intractability of CPA motivates consideration of approximate techniques. This section describes the approximation algorithm for MCND that runs in polynomial time and has a constant-factor performance guarantee.

The approximation algorithm is based on the insight that the maximum number of RNs that can be assigned is a *submodular* function of the set of mobile MBN locations and communication radii that are selected. Given a finite ground set $D = \{1, \ldots, d\}$, a set function $f(S)$, $S \subseteq D$, is submodular if

$$f(S \cup \{i, j\}) - f(S \cup \{i\}) \leq f(S \cup \{j\}) - f(S) \tag{2}$$

for all $i, j \in D$, $i \neq j$ and $S \subset D \setminus \{i, j\}$ [19]. Theorem 3 describes the submodularity of the objective function in the context of problem MCND.

Theorem 3 *Given an instance of MCND on a graph G, the maximum flow that can be routed through G is a submodular function of the set of arcs incident to t that are selected.*

Proof. The proof of Theorem 3 is similar to that of Lemma 1 in [5] and will not be presented here. □

4.1 Submodular Maximization with Multiple-Choice and Cardinality Constraints

Submodular maximization has been studied in many contexts, and with a variety of constraints. Nemhauser et al. [20] showed that for maximization of a nondecreasing, nonnegative submodular function subject to a cardinality constraint, a greedy selection technique produces a solution whose objective value is within $1 - \frac{1}{e}$ of the optimal objective value, where e is the base of the natural logarithm [21]. Approximation algorithms have also been developed for submodular maximization subject to other constraints, for example, Sviridenko [23] described a polynomial-time algorithm for maximizing a nondecreasing, nonnegative submodular function subject to a knapsack constraint.

In MCND, we aim to maximize a nonnegative, nondecreasing submodular function subject to L multiple-choice constraints and one cardinality constraint.

Algorithm 1

$S \leftarrow \emptyset$
$maxflow \leftarrow 0$
$U \leftarrow \{1, \ldots, L\}$
for k=1 to K **do**
  for $l \in U$ **do**
    for n=1 to N **do**
      if $f(S \cup \{y_l^n\}) \geq maxflow$ **then**
        $maxflow \leftarrow f(S \cup \{y_l^n\})$
        $y^* \leftarrow y_l^n$
        $l^* \leftarrow l$
      end if
    end for
  end for
  $S \leftarrow S \cup \{y^*\}$
  $U \leftarrow U \setminus \{l^*\}$
end for
return S

This is a special case of the problem of submodular maximization under multiple linear constraints as described by Kulik et al. [24]. Kulik et al. described the approximation algorithm for this problem, however, their approximation algorithm runs in polynomial time only if the number of linear constraints is a fixed constant [24]. Because MCND has $O(L)$ linear constraints, this algorithm can be quite computationally intensive. Fortunately, a simple greedy approach provides a provably good solution to MCND.

Consider Algorithm 1. Algorithm 1 starts with an empty set of selected arcs, S, and iteratively adds the arc that produces the maximum increase in the objective value, f, while maintaining feasibility with respect to the multiple choice constraints. After K iterations, Algorithm 1 produces a solution that obeys both the multiple-choice and cardinality constraints of MCND. The running time of Algorithm 1 is polynomial in K, L, and N; it requires solution of $O(KLN)$ maximum flow problems on bipartite networks with at most $N + K + 2$ nodes each. Moreover, Algorithm 1 carries a theoretical performance guarantee, as stated in Theorem 4.

Theorem 4 *Algorithm 1 is an approximation algorithm for MCND with approximation guarantee* $\frac{1}{2}$.

Proof. The proof of Theorem 4 appears in Appendix 3. □

That is, if the optimal solution to an instance of MCND has objective value OPT, then Algorithm 1 produces a solution S such that $f(S) \geq \frac{1}{2}\text{OPT}$.

The performance guarantee of $\frac{1}{2}$ shown in Theorem 4 is indeed tight for some problem instances. For example, consider the instance of CPA shown in Fig. 2b, with $K = 2$, $\tau(c,r) = \frac{1}{cr^2}$, and $\tau_{\min} = 1$. The corresponding instance of MCND is shown in Fig. 2a. Note that on the first iteration of the greedy algorithm, nodes m_1^1, m_1^2, and m_2^1 are all optimal; each allows one unit of flow to traverse the graph.

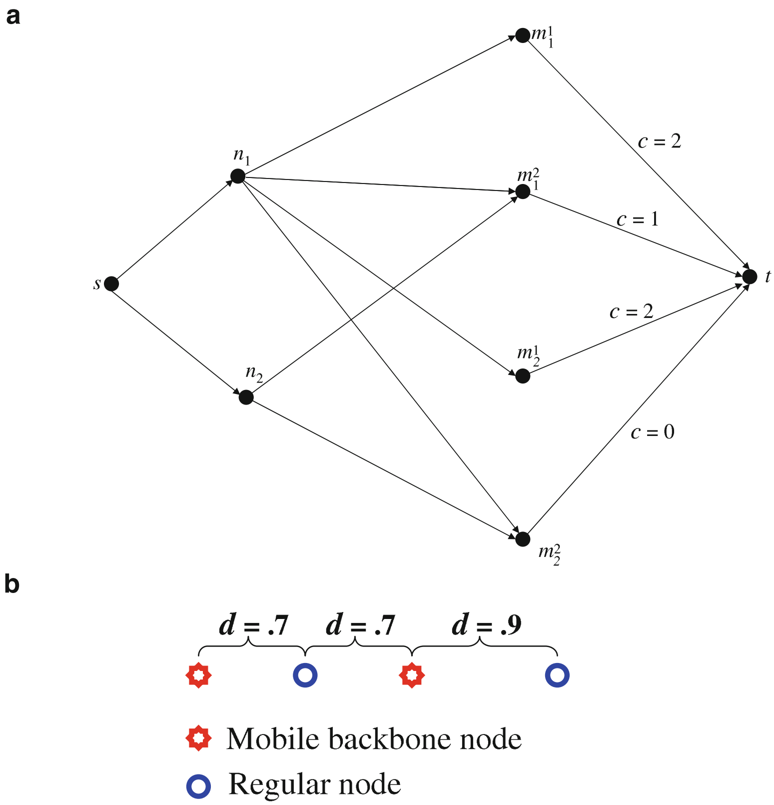

Fig. 2 Example of an instance of CPA for which the $\frac{1}{2}$ approximation guarantee of Algorithm 1 is tight. From left to right, the nodes shown are MBN 2, RN 1, MBN 1, and RN 2. **(a)** Example of an instance of MCND for which Algorithm 1 exactly achieves its performance guarantee. **(b)** A network optimization problem that yields the network design problem shown in Fig. 2a, for $\tau(c,r) = \frac{1}{cr^2}$ and $\tau_{\min} = 1$

Assume that the greedy algorithm selects node m_1^1. Then, on the greedy algorithm's second iteration, nodes m_2^1 and m_2^2 remain available for selection. However, neither of these nodes allows any additional flow to traverse the graph; thus, the total objective value obtained by the greedy algorithm is equal to 1, while an exact algorithm would have selected nodes m_1^2 and m_2^1 to obtain an objective value of 2.

While a theoretical performance guarantee is useful, the empirical performance of Algorithm 1 is also of interest. Figure 3 shows the average performance of Algorithm 1 relative to an exact (MILP) algorithm, for randomly-generated instances of CPA and their corresponding instances of MCND. Both RN locations and candidate MBN locations were generated according to a uniform distribution in a square area. As the figure indicates, Algorithm 1 tends to significantly outperform its performance guarantee, achieving average objective values up to 90% of those obtained by the exact algorithm, with a dramatic reduction in computation time. These results indicate that Algorithm 1 is a promising candidate for large-scale network design problems.

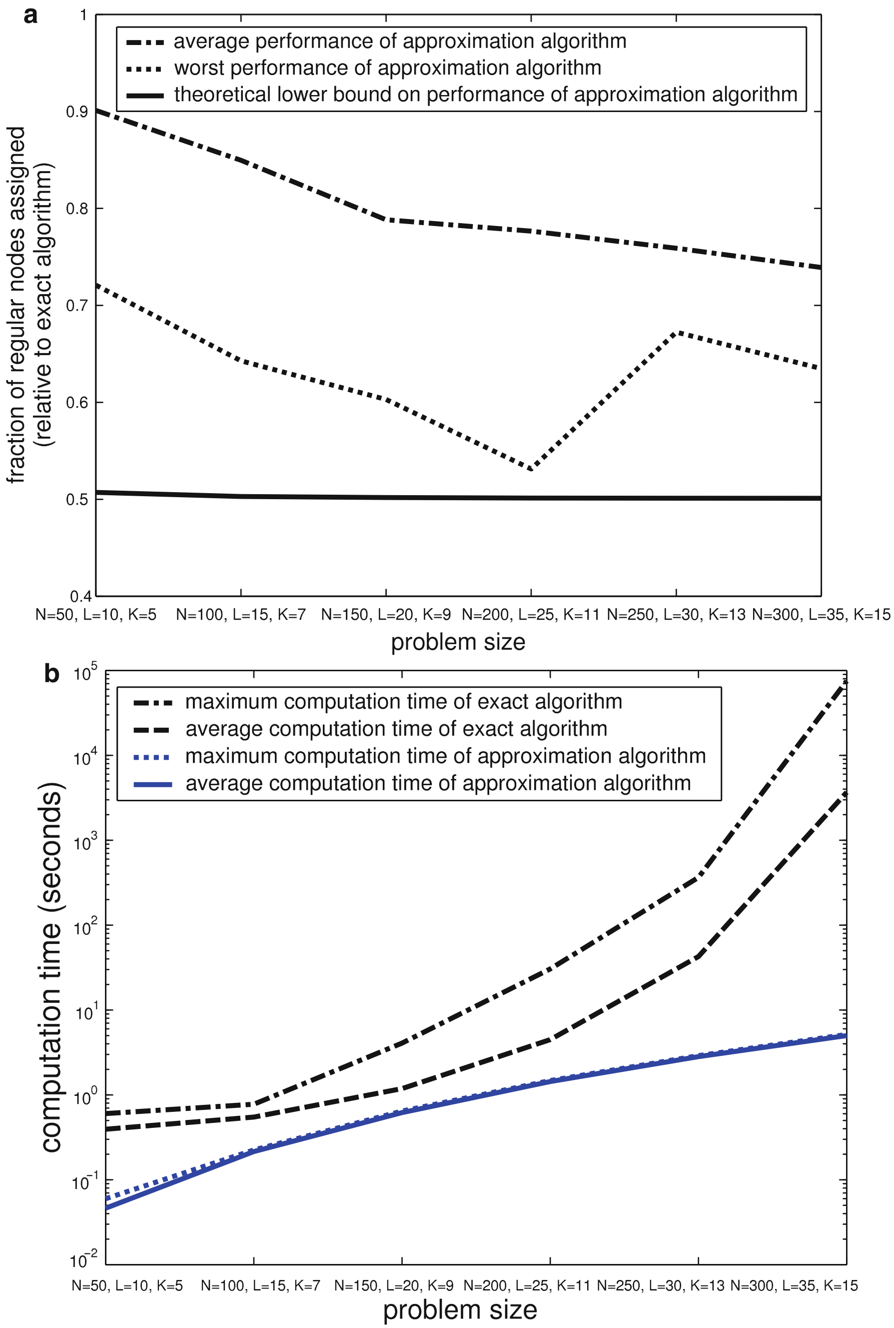

Fig. 3 Comparison of the exact and approximation algorithms developed in this work. **(a)** Performance of the approximation algorithm developed in this work, relative to an exact solution technique, in terms of number of RNs assigned at the given throughput level **(b)** Computation time of the approximation algorithm and the exact (MILP) algorithm for various problem sizes. Due to the large range of values represented, a logarithmic scale is used

5 Conclusion

This chapter has described algorithms for maximizing the number of RNs that achieve a threshold throughput level in a mobile backbone network. While previous work on this topic has assumed that MBNs are unconstrained in position, we model constraints in MBN location. Techniques described in this work include an exact algorithm based on mixed-integer linear programming (MILP) and polynomial-time approximation algorithm. Experimental results indicate that the approximation algorithm achieves good performance with a drastic reduction in computation time, making it suitable for a large-scale applications. We have also shown that the approximation algorithm carries a theoretical performance guarantee, and that this performance guarantee can indeed be tight in some instances, although the empirical performance of the approximation algorithm tends to exceed the performance guarantee.

Appendix 1. Proof of Theorem 1

Fix an instance of CPA, and consider a feasible solution to this instance, that is, a solution in which at most one MBN is placed in each location, each RN is assigned to at most one MBN, and each RN that is assigned to an MBN achieves throughput at least $\tau_{\min}$. Let A_k denote the set of RNs assigned to MBN k. Then, the objective value of this solution is $\sum_k |A_k|$. A solution to the corresponding instance of MCND with objective value $\sum_k |A_k|$ can be constructed as follows.

Consider MBN k occupying location l. Let f denote the most distant RN from k that is in set A_k, and let d_{fk} denote the distance from f to k. Then, set y_l^f equal to 1. For each RN $i \in A_k$, set x_{sn_i} and $x_{n_i m_l^f}$ equal to 1, and set $x_{n_i m}$ equal to zero for all other m. Note that the arc from node n_i to node $m_l^{n_f}$ is guaranteed to exist by construction. Flow conservation (constraint (1d)) is now satisfied at node n_i; repeating this process for each MBN $k = 1, \ldots, K$ results in constraint (1d) being satisfied for all $i \in \cup_k A_k$. For all $i \notin \cup_k A_k$, set x_{sn_i} and $x_{n_i m}$ equal to zero for all m. Flow conservation is now satisfied at all nodes $n_1, \ldots, n_N$, and capacity constraints (1e) and (1f) are now satisfied for all arcs except those entering the sink t. Setting the remaining binary variables to zero results in constraints (1b), (1c), and (1i) being satisfied. Finally, consider the arcs entering the sink t. If the MBN is not placed at location l with radius r_l^n, the arc connecting node m_l^n to t has capacity zero. Therefore, set $x_{m_l^n t}$ to zero, and note that because node m_l^n has no incoming flow, constraint (1d) is satisfied at node m_l^n. On the other hand, if the MBN is placed at location l and has RN n as its most distant assigned RN, then the arc connecting node m_l^n to t has capacity c_l^n. Node m_l^n has $|A_k|$ units of incoming flow, and by definition of c_l^n and our assumption that our original solution to CPA is feasible, we know that $|A_k| \leq c_l^n$. Therefore, we can set $x_{m_l^n t}$ equal to $|A_k|$, thus satisfying constraints (1d) and (1g) for all nodes and arcs. The objective value of this solution is $\sum_k |A_k|$.

We have shown that for every feasible solution to CPA, there is a corresponding feasible solution to MCND with the same objective value. It remains to be shown that for every optimal solution to MCND, there is a corresponding solution to CPA with the same objective value.

Consider an optimal solution to an instance of MCND, and assume that all of the flows in this solution are integer. (Due to total unimodularity of the network flow constraint matrix, all flows $\mathbf{x}$ are integer in all basic feasible solutions of the linear maximum flow problem induced by a specification of the $\mathbf{y}$ vector.) For each $l = 1, \ldots, L$, place in the MBN at location l if and only if $y_l^n = 1$ for some n. K MBNs have now been placed.

For all l and n, if $y_l^n = 1$, assign to the MBN at location l the set of RNs for which $x_{n_i m_l^n} = 1$ in the solution to MCND. Note that, by definition of c_l^n, all of the assigned RNs achieve throughput at least $\tau_{\min}$. Furthermore, because all arcs originating at s have unit capacity, and because all flows in the solution are integer, each RN can be assigned to at most one MBN. Thus, we have obtained a feasible solution to CPA. Furthermore, the value of this solution is equal to that obtained in CPA: each unit of flow represents exactly one RN that is successfully assigned at throughput at least $\tau_{\min}$. Thus, MCND yields an optimal solution to CPA.

Appendix 2. Proof of Theorem 2

The proof of Theorem 2 reduces an instance of the *Euclidean K-center problem on points* to CPA. In the *Euclidean K-center problem*, the input is a set of N points on the plane and a positive real number r, and the objective is to determine whether it is possible to place K discs of radius r in the plane such that every input point is within distance at most r from the center of at least one disc, i.e., every point is covered by at least one disc. The *Euclidean K-center problem on points* has the additional restriction that the center of each disc must coincide with one of the N input points. Both versions of the problem are known to be NP-complete [25].

Proof. Fix an instance of the Euclidean K-center problem on points. Denote the input points by $\mathrm{N} = \{1, \ldots, N\}$ and the radius by r. This instance can be reduced to an instance of CPA as follows: Define N RNs, and let their locations coincide with the input points. Next, define N candidate MBN locations also coinciding with the input points, and let K be the number of MBNs to be placed. Fix $\tau_{\min}$, and define the throughput function τ as follows:

$$\tau(A_k, d_{nk}) = \tau(d_{nk}) = \begin{cases} \tau_{\min} & \text{if } d_{nk} \le r, \\ 0 & \text{if } d_{nk} > r. \end{cases} \tag{3}$$

Note that τ fits the assumptions stated in Sect. 2; it is monotonically nonincreasing with d_{nk} and does not vary with A_k.

Denote an optimal solution to CPA by (A^*, B^*), where B^* denotes the placement of the MBNs (i.e., the subset of the candidate locations $1, \ldots, N$ that are occupied by MBNs) and A^* denotes the optimal assignment of RNs to MBNs. Assume without loss of generality that the nodes are numbered such that $B^* = \{1, \ldots, K\}$. Let A_k denote the set of RNs assigned to MBN k in solution (A^*, B^*).

If the optimal objective value of this instance of CPA is equal to N, then the answer to the original Euclidean K-center problem on points is YES. Given a solution to CPA (A^*, B^*) in which $\sum_k |A_k| = N$, a solution to the Euclidean K-center problem on points in which all points are covered can be constructed by placing discs at locations B^*. By our assumption that all RNs in the set A_k achieve throughput at least $\tau_{\min}$, it follows that all RNs in the set A_k are within radius r of the disc at location k and thus are covered by that disc. Furthermore, since each RN can be assigned to at most one MBN, the fact that $\sum_k |A_k| = N$ implies that *all* RNs achieve throughput at least $\tau_{\min}$. Therefore, all nodes in the original Euclidean K-center problem on points are covered by discs placed at locations B^*.

Likewise, if the answer to the original Euclidean K-center problem on points is YES, then the optimal objective value the corresponding instance of CPA must be equal to N. Let B^* denote a placement of discs such that each input point is covered by at least one disc, and again denote this placement by $B^* = \{1, \ldots, K\}$. Let $C_n \in B^*$ denote the set of discs that cover point n. If point n is covered by the disc at location $k \in C_n$, then the RN at location n can be assigned to the MBN at location k and achieve throughput at least $\tau_{\min}$ in CPA. Since throughput is not a function of cluster size in (3), a feasible solution to CPA consists of a placement of MBNs at the locations in B^* and an assignment A in which each RN n is assigned to exactly one of the MBNs occupying locations in C_n.

Thus, the Euclidean K-center problem on points can be reduced to CPA. The time required to perform this reduction is polynomial in the number of input points; therefore, CPA is NP-hard. □

Appendix 3. Proof of Theorem 4

Proof. Consider the problem of maximizing a nonnegative, nondecreasing submodular set function $f(S)$ subject to multiple-choice and cardinality constraints. Items eligible for inclusion in S belong to a ground set D that is divided into C disjoint subsets called *classes*. A set S is feasible if $|S| \leq K$ and no two items in S belong to the same class. Note that because f is nondecreasing, there always exists an optimal solution such that $|S| = K$.

Let S^g denote the set of items selected by the greedy algorithm, and let S^* denote the set of items selected by an exact algorithm (i.e., the optimal solution). We wish to find a lower bound on the ratio of $f(S^g)$ to $f(S^*)$.

Consider first the special case in which $C = K$, i.e., the number of elements to be selected is equal to the number of item classes. In this case, exactly one item from each class is to be selected.

Assume without loss of generality that the item classes are numbered such that the item from class k was chosen by the greedy algorithm during its kth iteration, for $k = 1, \ldots, K$. Denote the kth item selected by the greedy algorithm by i_k, and denote the item from class k selected by the exact algorithm by i_k^*. Furthermore, denote the set of items selected by the greedy algorithm up to iteration k by S_k^g, i.e., $S_k^g = \{i_1, \ldots, i_k\}$. Finally, denote the marginal increase in the objective value obtained by adding item i_k to the set S_{k-1}^g by δ_k, i.e., $\delta_k = f(S_k^g) - f\left(S_{k-1}^g\right)$. Note that

$$f(S^g) = \sum_{k=1}^{K} \delta_k$$

and

$$\delta_k \leq \delta_{k-1}.$$

A set function g defined over a ground set U is submodular if and only if [21]

$$g(T) \leq g(S) + \sum_{j \in T \setminus S} (g(S \cup \{j\}) - g(S)) - \sum_{j \in S \setminus T} (g(S \cup T) - g(S \cup T \setminus \{j\})) \forall S,\ T \subseteq U.$$

Because f is a nonnegative, nondecreasing submodular function, the final term $\sum_{j \in S \setminus T} (f(S \cup T) - f(S \cup T \setminus \{j\}))$ is nonnegative, and therefore

$$f(T) \leq f(S) + \sum_{j \in T \setminus S} (f(S \cup \{j\}) - f(S)) \ \ \forall S, T \subseteq U. \tag{4}$$

In particular,

$$f(S^*) \leq f(S^g) + \sum_{j \in S^* \setminus S^g} (f(S^g \cup \{j\}) - f(S^g)). \tag{5}$$

Consider item $i_k^* \notin S^g$. Because i_k^* was not chosen by the greedy algorithm, it follows that

$$f(S_{k-1}^g \cup \{i_k^*\}) - f(S_{k-1}^g) \leq \delta_k.$$

By (2),

$$f(S^g \cup \{i_k^*\}) - f(S^g) \leq f(S_{k-1}^g \cup \{i_k^*\}) - f(S_{k-1}^g), \tag{6}$$

$$\leq \delta_k. \tag{7}$$

Substituting this into (5), we obtain:

$$\begin{aligned} f(S^*) &\leq f(S^g) + \sum_{j \in S^* \setminus S^g} f(S^g \cup \{j\}) - f(S^g) \\ &\leq f(S^g) + \sum_{k=1}^{K} \delta_k \\ &= 2f(S^g). \end{aligned}$$

Thus, we have obtained a bound on the ratio of $f(S^g)$ to $f(S^*)$ for the special case in which $C = K$:

$$\frac{f(S^g)}{f(S^*)} \geq \frac{1}{2}. \tag{8}$$

Now consider the case in which $C > K$. In this case, neither the greedy algorithm nor the exact algorithm can select an item from every class, and the two algorithms will not necessarily select items from the same classes. Let S_s^* denote the set of items in the optimal solution that belong to classes from which the greedy algorithm also selected an item, and let S_d^* denote the set of items in the optimal solution that belong to classes from which the greedy algorithm did not select an item. Note that $S^* = S_s^* \cup S_d^*$. Denote the item classes by $c = 1, \ldots, C$.

Consider item $i \in S_s^*$ belonging to class c, and denote the item selected from class c by the greedy algorithm as i_c. Assume that the greedy algorithm selected item i_c in iteration k. Then, by the same argument used in the case of $C = K$,

$$f(S^g \cup \{i\}) - f(S^g) \leq f(S_{k-1}^g \cup \{i\}) - f(S_{k-1}^g), \tag{9}$$

$$\leq \delta_k, \tag{10}$$

where S_{k-1}^g is again the set of items selected by the greedy algorithm at the beginning of iteration k, and δ_k is the marginal increase in the objective value obtained by adding item i_c to the set S_{k-1}^g, i.e., $\delta_k = f(S_{k-1}^g \cup \{i_c\}) - f(S_{k-1}^g)$.

Now consider item $i \in S_d^*$ belonging to class c'. The greedy algorithm did not select an item from class c'; therefore:

$$\begin{aligned} f(S^g \cup \{i\}) - f(S^g) &\leq f(S_{K-1}^g \cup \{i\}) - f(S_{K-1}^g), \\ &\leq \delta_K, \end{aligned}$$

where δ_K is the marginal increase in the objective value obtained when the greedy algorithm adds the final element i_K to the set S^g_{K-1} to obtain $S^g = S^g_{K-1} \cup \{i_K\}$.

Substituting these inequalities into (5), we obtain:

$$\begin{aligned} f(S^*) &\leq f(S^g) + \sum_{j \in S_s^* \setminus S^g} f(S^g \cup \{j\}) - f(S^g) \\ &\quad + \sum_{j \in S_d^* \setminus S^g} f(S^g \cup \{j\}) - f(S^g) \\ &\leq f(S^g) + \sum_{k: i_k \in U_c, U_c \cap S^* \neq \emptyset} \delta_k + |S_d^*| \delta_K \\ &\leq f(S^g) + \sum_{k=1}^{K} \delta_k \\ &= 2f(S^g), \end{aligned}$$

where we have used the fact that $\delta_K \leq \delta_k$ for $k \leq K$.

Thus, the approximation ratio for the case of $C > K$ is the same as in the case of $C = K$, i.e., $\frac{1}{2}$.

We note that an alternative proof of this performance guarantee can be obtained by demonstrating the matroid structure of the feasible set [22]. □

References

1. 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.
2. 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.
3. A. Srinivas, G. Zussmanm and E. Modiano, Construction and Maintenance of Wireless Mobile Backbone Networks, *IEEE/ACM Trans. on Networking*, Vol. 17, No. 1, pp. 239–252, Feb. 2009.
4. A. Srinivas and E. Modiano, Joint node placement and assignment for throughput optimization in mobile backbone networks, *Proc. IEEE INFOCOM 08*, Apr. 2008.
5. E. M. Craparo, J. P. How and E. Modiano, Throughput Optimization in Mobile Backbone Networks, *IEEE Transactions on Mobile Computing*, Vol. 10, No. 4, pp. 560–572, Apr. 2011.
6. 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 (accessed Sept. 2009).
7. E. M. Craparo, Constrained Node Placement and Assignment in Mobile Backbone Networks, *Proc. IEEE Conference on Decision and Control*, Dec. 2010.
8. D. Jea, A. A. Somasundara, and M. B. Srivastava, Multiple controlled mobile elements (data mules) for data collection in sensor networks, In *Proc. IEEE/ACM DCOSS 05*, Jun. 2008.
9. R. Shah, S. Roy, S. Jain, and W. Brunette, Data MULEs: Modeling a three-tier architecture for sparse sensor networks, In *Proc. IEEE SNPA03*, May 2003.

10. M. M. Bin Tariq, M. Ammar, and E. Zegura, Message ferry route design for sparse ad hoc networks with mobile nodes, In *Proc. ACM MobiHoc 06*, May 2006.
11. W. Zhao, M. Ammar, and E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, In *Proc. ACM MobiHoc 04*, May 2004.
12. S. V. Hanly, An algorithm for combined cell-site selection and power control to maximize cellular spread spectrum capacity, *IEEE Journal on Selected Areas in Communications*, 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 Transactions on Vehicular Technology*, Vol. 40, pp. 641–646, 1993.
14. R. Yates and C. Y. Huang, Integrated power control and base station assignment, *IEEE Transactions on Vehicular Technology*, Vol. 44, No. 3, pp.638–644, 1995.
15. R. Mathar and T. Niessen, Integrated power control and base station assignment, *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, F. Malucelli, Radio planning and coverage optimization of 3G cellular networks, *Wireless Networks*, Vol. 14, pp. 435–447, 2008.
18. A. Ahuja, T. Magnanti and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.
19. D. Bertsimas and R. Weismantel, *Optimization over Integers*, Dynamic Ideas, 2005, p. 88.
20. 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.
21. G. Nemhauser, L. Wolsey, and M. Fisher, An analysis of the approximations for maximizing submodular set functions - I, *Mathematical Programming*, vol. 14, 1978, pp. 265–294.
22. M. Fisher, G. Nemhauser, and L. Wolsey, An analysis of the approximations for maximizing submodular set functions - II, *Mathematical Programming Studies*, vol. 8, 1978, pp. 73–83.
23. M. Sviridenko, A note on maximizing a submodular set function subject to a knapsack constraint, *Operations Research Letters*, Vol. 32, No. 1, Jan. 2004, pp. 41–43.
24. A. Kulik, H. Shachnai and T. Tamir, Maximizing submodular set functions subject to multiple linear constraints, *Proc. of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, 2009.
25. S. Masuyama, T. Ibaraki and T. Hasegawa, The computational complexity of the m-center problems on the plane, *Trans. IECE of Japan*, E-64, pp. 57–64, 1981.

Springer
http://www.springer.com/978-0-387-88619-0

Sensors: Theory, Algorithms, and Applications
Boginski, V.L.; Commander, C.W.; Pardalos, P.; Ye, Y.
(Eds.)
2012, X, 242 p.,
ISBN: 978-0-387-88619-0