Optimal Path Planning for Mobile Backbone Networks

Anand Srinivas and Eytan Modiano
Massachusetts Institute of Technology
Cambridge, MA 02139
{anand3, modiano}@mit.edu

Abstract— **Mobile Backbone Networks are heterogeneous wireless networks in which a subset of the nodes are more *capable* than others. The more capable nodes are referred to as Mobile Backbone Nodes (MBNs), whose primary role is to provide a mobile infrastructure in order to facilitate reliable end-to-end communication between nodes in the Network. In this paper, we consider the problem of optimally placing the MBNs *over a finite time horizon*. Specifically, we consider the path planning of a single MBN and aim to maximize the time-average system throughput. We present a discrete problem formulation, and develop an optimal solution for the single step velocity-constrained MBN placement problem. Using this as a sub-routine, we provide a greedy heuristic algorithm for the overall path planning problem. Next, we develop a dynamic programming based approximation algorithm for the problem. Finally, we compare the developed approaches via simulation.**

I. Introduction

Mobile Backbone Networks are heterogeneous wireless networks in which a subset of the nodes are more *capable* than others. The more capable nodes are referred to as Mobile Backbone Nodes (MBNs), and the others as Regular Nodes (RNs). The primary role of the MBNs is to provide a mobile infrastructure in order to facilitate reliable end-to-end communication between the RNs. Such a hierarchical architecture, recently proposed and studied by Rubin et al. and by Gerla et al. (see [9],[15] and references therein), is applicable to both Wireless Sensor Networks (WSNs) and Mobile Ad-Hoc Networks (MANETs).

In this paper, we consider the important problem of optimally placing the MBNs *over a finite time horizon*. Previous formulations of the Mobile Backbone Network construction problem have been based on only the knowledge of the instantaneous locations of the RNs. Specifically, at any given time, the MBNs are placed and mobilized *reactively* based on RNs' locations at that time. Indeed, this was the approach taken in [11], [12]. Yet, in many practical scenarios entire RN trajectories are known a-priori (e.g. as waypoints for particular missions). If this is the case, then placing the MBNs by solving a placement problem independently at each time step is, in general, suboptimal. In particular, it would be desirable to solve for the entire optimal *sequence of placements* for the MBNs at once. In this paper, we address this MBN *path planning* problem from a discrete perspective. For our exposition, we consider planning the path of a *single* MBN, given the trajectories of the RNs over a finite time horizon. Our goal is to optimize system throughput metrics along the lines of those presented in [11], averaged over the time horizon under consideration. The model we assume is that the "throughput" achieved by an RN transmitting to the MBN is a *decreasing* function of the distance between the RN and MBN. Furthermore, the system throughput is proportional to the throughput achieved by the *minimum throughput RN* (e.g. the farthest RN from the MBN). It is explained in detail in [11] that this is a reasonable model for several practical wireless systems (e.g. CDMA, Slotted Aloha-based, etc.). We elaborate further on the mathematical specifics of the model in section III.

It is important to note that if the throughput metrics are simply time-averaged and no consideration is given to the actual movement of the MBN, then there is a straightforward way to calculate the optimal MBN path. Specifically, combining the optimal solutions at each time step yields the overall optimal path. For example, we can obtain such solutions by employing the optimal algorithms developed in [11] independently at each time step. However, such an objective function can result in undesirable solutions for instances in which the required MBN motion in consecutive time steps is very large, even though the actual RN movement is small. In particular, instances exist in which the optimal MBN location moves an *arbitrarily large* distance in response to a small movement by the RNs [10]. Finally, there can be scenarios in which it is undesirable to have large MBN movements even in response to large RN movements, e.g. limited MBN velocity, energy constraints, MBN location predictability, etc.

To address this issue, we introduce a constraint on the MBN velocity. This immediately causes a dependence between the solutions at each time step, considerably increasing the difficulty of the overall problem. In this paper we develop two solution approaches for solving the MBN path planning problem. The first is a greedy approach in which at any timestep and given a "current" MBN location, we relocate the MBN, subject to the velocity constraint, such that the throughput objective at the next timestep is maximized. Note that the presence of the velocity constraint makes optimally solving the single step problem quite non-trivial, and thus an important contribution of this paper is an algorithm for this purpose. In the second approach we develop a dynamic-programming based algorithm that solves for the entire MBN path at one shot. We elaborate upon the advantages and

This research was supported by NSF grant CCR-0325401, by ONR grant N000140610064, and by a grant from Draper Laboratory.

978-1-4244-2247-0/08/$25.00 ©2008 IEEE.

disadvantages of the two methods later in the paper.

Finally, to our knowledge the MBN path planning (MPP) problem with throughput maximization objective has not been considered in the literature. Yet, as mentioned in section II several closely related problems and formulations have been considered. The goal of this paper is to provide a basic discrete formulation, as well a characterization of two natural solution methodologies.

The remainder of this paper is organized as follows. In section II we discuss related work. In section III we provide the problem model and formulation. We develop a greedy solution approach in section IV and a dynamic-programming based approach in section V. Finally, in section VI we present simulation results comparing the two approaches.

II. Related Work

To our knowledge the velocity constrained MBN path planning problem with a throughput maximization objective has not been considered in the literature. Yet, several closely related problems and formulations have been considered. From a discrete perspective, the problem is closely related to several time-horizon network planning and facility location works considered in the past, e.g. [13],[14],[6],[4]. Yet, a key difference between the MPP problem and the network planning works is that for the MPP problem, the set of potential locations for the MBNs is infinite (i.e. anywhere on the plane). By contrast, the network location work assumes that centers/medians (e.g. MBNs in our context) can only move along edges and vertices of the graph. Moreover, we consider general non-linear objective functions as well as a hard constraint on the MBN movement, as will be further described later in the paper.

Along the lines of hard constraints on MBN movement (e.g. velocity) is the work of [2], in which they consider what approximation ratios to the *unconstrained* 1-center/median metrics can be achieved when the ratio of the MBN to RN velocity is upper bounded. By contrast, we enforce a velocity bound on the MBN, but leave the RN velocity unbounded. Our focus is on characterizing the performance with respect to the *MBN velocity constrained* MPP objective function.

Finally, it should be noted that time horizon network planning and facility location problems have also been formulated in the continuous domain, e.g. [8], [10]. Yet, the problems considered and the solutions employed significantly differ from those of this paper.

III. Problem Formulation

We consider a network consisting of N RNs $P = \{p_1, p_2, \ldots, p_N\}$ and a single MBN M, and a finite time horizon $[0, T]$, discretized by Δt-spaced time steps $t = 0, 1, \ldots, K$, $K = T/\Delta t$. We assume all of the nodes in the network are situated on a 2-dimensional plane. We denote by $p_i(t) \triangleq (p_{i_x}(t), p_{i_y}(t))$, $t = 0, 1, \ldots, K$, the x-y position of RN p_i at time step t. Similarly, we define $M(t) \triangleq (M_x(t), M_y(t))$ for the MBN M. Let $d[u, v]$ denote the Euclidean distance between two nodes u and v. We let $d_i(t)$ denote the distance between RN p_i and the MBN M at time step t, i.e., $d_i(t) = d[M(t), p_i(t)]$. Finally, let $d_{max}(t)$ represent the distance from the MBN to the farthest RN at time step t, i.e., $d_{max}(t) = \max_i d_i(t)$.

We assume the trajectories of the RNs are known a-priori over the full time horizon $t = 0, 1, \ldots, K$. Thus the goal is to compute a path $M^* = M(0), M(1), \ldots, M(K)$ for the MBN given this information. We assume the initial position of the MBN M_0 is fixed and known, i.e. $M(0) = M_0$. Finally, we enforce a hard constraint that the maximum speed of the MBN is upper bounded by V, i.e., $d[M(t-1), M(t)] \leq V\Delta t$, $\forall t = 1, \ldots, K$.

In this work we are concerned with maximizing the time average of the system throughput. This throughput objective (without the time-averaging aspect) function was described in [11] as the Maximum-Fair-Placement-and-Assignment (MFPA) throughput metric. Specifically, at a given timestep t, the system throughput is equal to $H[d_{max}(t)]$, where $H()$ is a *decreasing* function that represents the throughput received by the worst case throughput RN. As is described in [11], for several practical system models (e.g. a CDMA-type system), $H[d_{max}(t)]$ can serve as a proxy to describe the system throughput. Thus we have that over K timesteps the time averaged throughput is equal to $\frac{1}{K}\sum_{t=1}^{K} H[d_{max}(t)]$. We term the MBN path planning problem with time average MFPA objective function the ***MPP-MFPA*** problem, and formulate it below.

Problem MPP-MFPA: Given the RN trajectories $p_i(t), \forall i \in P, t = 0, \ldots, K$ and initial MBN position $M(0) = M_0$. Compute the MBN path $M^* = M(0), M(1), \ldots, M(K)$ such that the average MFPA throughput metric is maximized, subject to the maximum MBN speed bounded by V. Mathematically, the problem is expressed as,

$$\max_{M^*} \quad \frac{1}{K}\sum_{t=1}^{K} H[d_{max}(t)] \tag{1}$$

$$s.t. \quad d[M(t-1), M(t)] \leq V\Delta t, \quad \forall t = 1, \ldots, T \tag{2}$$

$$M(0) = M_0 \tag{3}$$

IV. Greedy Approach to the MPP-MFPA problem

In this section we develop a greedy approach towards solving the MPP-MFPA problem. The most natural greedy approach to a multi-step optimization problem aims to optimize the 1-step instantaneous problem at each time step. We take this approach, and present a high level algorithm below.

Algorithm 1 High Level MPP-MFPA Greedy Algorithm

1: **Initialize** $M(0) = M_0$
2: **for** t=1,2,...,K **do**
3: **Compute** the location for $M(t)$ that maximizes $H[d_{max}(t+1)]$, subject to $d[M(t-1), M(t)] \leq V\Delta t$
4: **return** $M^* = M(0), M(1), \ldots, M(K)$

The key step in the high level algorithm is the solution of the 1-step optimization problem in line 3. A more complete formulation of this problem is as follows.

Problem 1-step MPP-MFPA: Given the RN positions at time $t+1$, $p_i(t+1), \forall i \in P$ and previous MBN position, $M(t)$.

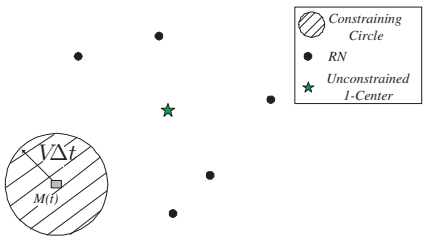

Fig. 1. Illustration of constrained 1-center instance in which the unconstrained 1-center is outside the constraining circle.

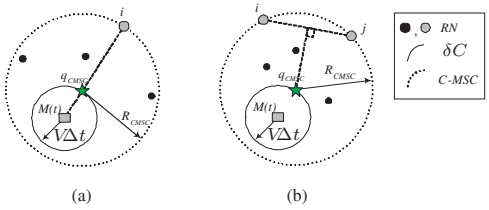

Fig. 2. Illustration of the two unique ways the constrained 1-center can be defined. (a) By a single RN. (b) By a pair of RNs.

Calculate the optimal MBN position at time $t+1$, $M(t{+}1)$, such that the MFPA throughput metric at time $t+1$ is maximized, subject to the maximum MBN velocity bounded by V. Mathematically, the problem is expressed as,

$$\max_{M(t+1)} \quad H[d_{max}(t+1)] \tag{4}$$

$$s.t. \quad d[M(t), M(t+1)] \leq V\Delta t \tag{5}$$

With the above formulation, we note that the 1-step MPP-MFPA problem can be viewed as a *constrained 1-center problem*[1]. Specifically, since $H()$ is a decreasing function in $d_{max}(t+1)$, minimizing $d_{max}(t+1)$ will maximize the objective function in (4). If not for the velocity constraint in (5), the problem would reduce to finding the unconstrained 1-center, for which several efficient polynomial time algorithms exist (e.g. [1]). Yet, with the constraint in mind we can view the problem as one in which we need to to find the 1-center of the RNs at time $t+1$ such that it lies within a circle of radius $V\Delta t$ around $M(t)$. This is depicted in Fig. 1.

The convex polygon constrained 1-center problem was addressed in [3]. Yet, their algorithm cannot be applied here since a circular constraint cannot be expressed as a polygonal constraint. In the next section we develop a simple optimal algorithm to solve the circular constrained 1-center problem.

A. Circular Constrained 1-Center (CC-1C) Algorithm

We begin with the following observation, which provides the first step in our algorithm to solve the circular constrained 1-center problem. Let C denote the constraining circle of radius $V\Delta t$ with center $M(t)$

Observation 1: If the solution to the unconstrained 1-center problem lies within the circle C, then this is the solution to the constrained 1-center problem.

Thus the main difficulty lies in solving the constrained problem *when the unconstrained solution lies outside* C (e.g. shown in Fig. 1). The following lemma provides the first key to solving this problem, where we have defined δC as the boundary of the circle C.

Lemma 1: Assume the solution to the unconstrained 1-center problem lies outside the circle C. Then, the solution to the constrained 1-center problem must lie on δC.

Proof: By the previous discussion, the solution to the constrained 1-center problem involves minimizing $d_{max}(t{+}1)$ subject to the circular constraint. The proof involves first showing that $d_{max}(t{+}1)$ is *convex* in $M(t+1) \triangleq [M_x, M_y]$. This will allow us to conclude that from a given MBN placement at $M(t{+}1)$, changing the solution along the gradient direction ∇d_{max} will decrease $d_{max}(t{+}1)$. Note that while $d_{max}(t)$ is not differentiable at certain points, directional derivatives exist everywhere [5]. Next, we assume that the circular constrained optimal $M(t{+}1)$, denoted $M^*(t{+}1)$, is an *interior* point of C. However, if this were the case, then there must exist another location $M'(t{+}1)$ along the direction $\nabla d_{max}(t{+}1)$ such that $M'(t{+}1)$ is within C (i.e. either also interior to C or on δC). Thus $M'(t{+}1)$ must yield a lower value of $d_{max}(t{+}1)$, contradicting the optimality of $M^*(t{+}1)$.

To see why $d_{max}(t{+}1)$ is convex in $[M_x, M_y]$ consider its full expansion, dropping the $(t{+}1)$ dependence for legibility.

$$\begin{aligned} d_{max} &= \max_{i \in P} d_i \\ &= \max_{i \in P} \left\{ \sqrt{[M_x - p_{i_x}]^2 + [M_y - p_{i_y}]^2} \right\} \end{aligned} \tag{6}$$

We know from optimization theory that the maximum of a set of convex functions is also convex. Since the Euclidean distance function $d_i()$ is convex [5], the result follows. ■

Lemma 1 allows us to restrict our search for $M(t+1)$ to the locus of points defined by δC. We define the *Constrained Minimum Spanning Circle* ($CMSC$) for the RNs at time $t+1$, as the circle with center at the optimal location of $M(t+1)$ and radius equal to the corresponding value of $d_{max}(t{+}1)$. We denote the center and radius of the $CMSC$ as q_{CMSC} and R_{CMSC} respectively. Consider the following lemma regarding the $CMSC$, illustrated in Fig. 2.

Lemma 2: Assume the unconstrained 1-center is outside the constraining circle C. Then there are are two unique ways the $CMSC$ can be defined.

1) By a single RN $i \in P$. If this is the case, then q_{CMSC} is located at the *first* intersection between δC and the directed line segment $\overrightarrow{iM(t)}$. R_{CMSC} is equal to $d(q_{CMSC}, i)$.
2) By a pair of RNs $i, j \in P$. If this is the case, then q_{CMSC} is located at an intersection point between δC and the perpendicular bisector of i and j. The intersection point is chosen to minimize $R_{CMSC} = d(q_{CMSC}, i)$.

Proof: Recall that by Lemma 1, the optimal q_{CMSC} must lie on δC. We now go through several cases regarding the farthest RN(s) from q_{CMSC}. First assume exactly one RN

[1]The (unconstrained) 1-center problem places a single MBN such that the farthest distance from any RN to the MBN is minimized.

is farthest from q_{CMSC}. In this case, in order to minimize R_{CMSC} subject to the constraint that $q_{CMSC} \in \delta C$ it is a basic geometric fact that q_{CMSC} and R_{CMSC} are defined as in the first part of the Lemma. The same holds true with respect to the second part of the Lemma if we assume that exactly two RNs are simultaneously farthest from q_{CMSC}. Finally, assume exactly $k \geq 3$ RNs are simultaneously farthest. In this case, we have that *all pairs* of the k farthest RNs must be equidistant from the center. Yet, a pair of equidistant RNs coupled with the constraint that the center must be on δC uniquely determines a center location (e.g. as per the second part of the Lemma). Therefore, $k \geq 3$ simultaneously farthest RNs represents an over-determined situation, wherein the corresponding q_{CMSC}, R_{CMSC} tuple would have been considered under the second part of the Lemma. ■

Algorithm 2 CC-1C Algorithm

1: **Compute** the unconstrained 1-center location, q_{UC}, using an algorithm from [1]
2: **if** q_{UC} is within C **then**
3: **return** $M(t+1) = q_{UC}$
4: **Set** $R_{min} = \infty$
5: **for** all *single* RNs $i \in P$ **do**
6: **Let** q be the first intersection point between the line segment $\overrightarrow{iM(t)}$ and δC.
7: **Let** R_q be the distance between q and i, $R_q = d(q,i)$
8: **if** $d(q,j) \leq R_q, \forall j \in P$ and $R_q < R_{min}$ **then**
9: **Set** $M(t+1) = q$ and $R_{min} = R_q$
10: **for** all *pairs* of RNs $i, j \in P$ **do**
11: **if** the perpendicular bisector of i, j and δC intersect **then**
12: **Let** q be that intersection point which yields the lowest value of $d(q,i)$.
13: **Let** R_q be the distance between q and i, $R_q = d(q,i)$
14: **if** $d(q,k) \leq R_q, \forall k \in P$ and $R_q < R_{min}$ **then**
15: **Set** $M(t+1) = q$ and $R_{min} = R_q$
16: **return** $M(t+1)$

The Circular Constrained 1-Center (CC-1C) algorithm is shown above. The algorithm, which finds the constrained 1-center, works by directly applying the constructive implications of the previous discussion. It starts by checking whether the condition outlined in observation 1 holds. Assuming it does not, we next check all possible q_{CMSC}, R_{CMSC} tuples as outlined in Lemma 2 to see if they define a valid CMSC (i.e. if they cover all the RNs). The valid CMSC with minimum radius is taken as the overall solution.

The computational complexity of the CC-1C algorithm is $O(N^2)$, where N is the number of RNs. This is because the for loop in line 10 considers all pairs of RNs, and thus results in the most complex operation. The solution of the unconstrained problem (line 1) can be found with $O(N \log N)$ computational complexity [1].

V. Dynamic-Programming-based Approach

The greedy algorithm derived in the previous section has good performance for most problem instances. However, as is the case with any general path-planning type problem, there exist certain "bad" problem instances in which a greedy-type solution can be significantly sub-optimal, and for which one needs to solve for the entire path at a single shot. In this section we provide a dynamic-programming based algorithm that does just this. We start, however, by motivating the necessity of this type of algorithm by providing a problem instance in which the greedy approach has poor performance.

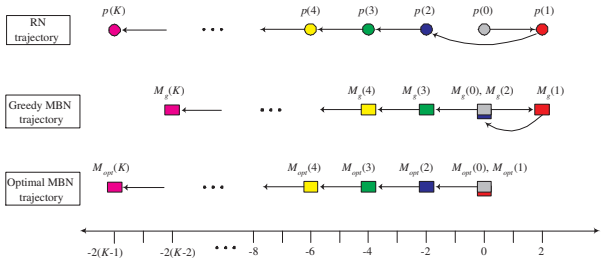

Fig. 3. Single RN, Single MBN, 1-D example of greedy vs. optimal approaches. Assume $M_g(0) = M_{opt}(0) = p(0) = 0$ and $V = 2$, $\Delta t = 1$.

The example, involving a single RN travelling on a line is illustrated in Fig. 3. The trajectory of the RN is shown on top, and we assume that the MBN's speed is bounded by $V = 2$, and that $\Delta t = 1$, i.e. $K = T$. In the example both the RN and MBN start at the same location on the line, i.e. $M(0) = p(0) = 0$. Note that in the MPP-MFPA formulation, the speed of the RN is not bounded, and in the example it travels at speed equal to 4 between time steps 1 and 2. In many scenarios, RNs might not travel faster then the MBNs. Yet, as mentioned earlier, the 1-center of the RNs can travel arbitrarily faster than any particular RN movement. Thus one can also think of the RN in the example as a proxy for the 1-center of a number of RNs.

An MBN path obtained by applying a greedy (i.e. myopic) approach that tries to maximize the instantaneous MFPA objective at every time step is shown in the middle of the figure. Notice that for all time steps $t \geq 2$, the greedy MBN trails the RN by a distance of 2. This results in an MPP-MFPA objective of $\frac{1}{K}[H(0)+(K-1)H(2)]$. By contrast, the optimal MBN path involves accepting some sub-optimality in the first time step by staying at position 0 at time step $t = 1$. However, doing this allows the optimal MBN to follow the RN exactly for all time steps $t \geq 2$, yielding an MPP-MFPA objective of $\frac{1}{K}[H(2) + (K-1)H(0)]$. Depending on the exact form of $H()$, this can be significantly larger than that achieved by the greedy approach.

A. DPA Algorithm

We now present the Dynamic-Programming based Approximation algorithm (DPA). We start by gridding the plane with vertical and horizontal spacing $\epsilon \leq V\Delta t$. Next, we construct K copies of the resultant grid points, denoted by $Y(1), Y(2), \ldots, Y(K)$, where a grid point $v \in Y(t)$ represents a potential location for the MBN at time t. For notational convenience, we define the set $Y(0)$ to denote just a single point, M_0, i.e. the given MBN starting position.

We next define an edgeweighted graph $G = (V', E)$, illustrated in Fig. 4, as follows. Let the vertex set V' consist of all the $Y(t)$'s, $t = 0, 1, \ldots, K$. We add an edge (u, v)

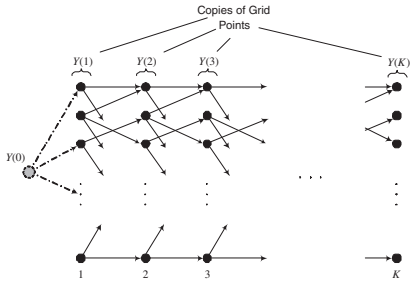

Fig. 4. Illustration of Trellis Structure. Edges between vertices at consecutive time steps are drawn only if the grid points they represent are at most $V\Delta t$ distance apart.

to E between $u \in Y(t)$, $v \in Y(t+1)$, $t = 0, \ldots, K-1$, if $d(u,v) \leq V\Delta t$, where $d(u,v)$ is the distance between grid points u and v. Constructing the edge set in this way restricts the MBN to only travel between grid points in successive time steps that are at most a distance $V\Delta t$ apart. Finally, we define the weight $w(u,v)$ of an edge $(u,v) \in E$, $u \in Y(t)$, $v \in Y(t+1)$, $t = 0, \ldots, K-1$ to equal to the *instantaneous throughput value assuming the MBN is located at v at time $t+1$*. Specifically, this is expressed as $w(u,v) = H[\max_i\{d(v, p_i(t+1))\}]$. By construction, the graph has the following property; this forms the main justification for the algorithm.

Lemma 3: Assume the MBN is restricted to travel between grid points during time steps $t = 1, \ldots, K$. The optimal MPP-MFPA path *subject to this restriction* is equivalent to the longest (maximum weight) path in G from the vertex $Y(0)$ to some vertex $v \in Y(K)$.

We next observe that the graph G represents a *Trellis Graph*, or more generally, a *Directed Acyclic Graph*. In such graphs the *longest path* can be found efficiently in a similar manner to finding a shortest path, e.g. by slightly modifying a well known dynamic-programming based algorithm known as the *Viterbi algorithm* [7]. For brevity we do not present the algorithm here, although a detailed description can be found in [10]. The computational complexity of the algorithm is equal to $O\big(\big(|Y(1)|\big) \cdot (\lceil \frac{2V\Delta t}{\epsilon} \rceil)^2 \cdot K\big)$, where $|Y(1)|$ is the total number of grid points. Note that for a plane of dimensions $L \times L$, $|Y(1)| = (\lfloor \frac{L}{\epsilon} \rfloor + 1)^2$.

Finally, we note that as mentioned in Lemma 3, the DPA algorithm finds the optimal MPP-MFPA path subject to the constraint that the MBNs must only travel between grid points. Yet, it would be desirable to calculate how close this solution approximates the original unconstrained optimal MPP-MFPA solution. It turns out that for an unbounded plane, this can be a function of both the grid spacing ϵ as well as the end time step K. In presenting the results of the analysis, for brevity we omit the proofs, which can be found in [10]. We define $d^{opt}_{max}(t)$ to be the distance from the MBN to the farthest RN at time step t in the optimal solution (i.e. not constrained to lie on grid points).

Theorem 1: For an unbounded plane, the MPP-MFPA objective value of the solution path found by the DPA algorithm is *at least* $\frac{1}{K}\sum_{t=1}^{K} H(d^{opt}_{max}(t) + 2\sqrt{2}t\epsilon)$.

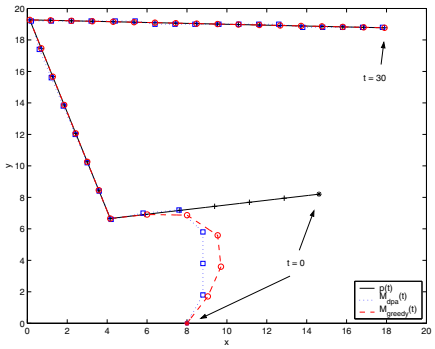

Fig. 5. MPP-MFPA Single RN, 2-D example. The RN travels according to a random waypoint model. Both DPA and Greedy Approaches use $\Delta t = 1$. The plot show the MBNs spatial movement with respect to the RN, and '*' is used to depict the starting locations.

Yet, since the above Theorem simply shows a lower bound on the MPP-MFPA objective achievable by the DPA algorithm, we do not know if this is a true reflection of the worst case performance of the algorithm. As reflected in the following Theorem, it turns out the lower bound is tight in the sense that the difference between $d^{dpa}_{max}(t)$ and $d^{opt}_{max}(t)$ can potentially increase without bound as a function of the number of time steps (i.e. assuming fixed Δt).

Theorem 2: For an unbounded plane, there exists a worst case problem instance where $\lim_{t\to\infty}\{d^{dpa}_{max}(t) - d^{opt}_{max}(t)\} = \infty$.

VI. Simulation Results

In this section we present simulation results comparing the greedy and DPA algorithmic approaches developed in this paper. We start with the situation in Fig. 5. This shows a single RN instance travelling in a 20×20 2-dimensional plane according to a Random Waypoint Model. In such a mobility model, RNs continually repeat the process of choosing a random location in the plane and travel there at a randomly chosen constant speed in the range $[V_{min}, V_{max}]$. We chose $V_{min} = 0.5, V_{max} = 2$, and assumed the MBN speed was bounded by $V = 2$. We consider a time horizon $t \in [0, 30]$ with $\Delta t = 1$ for both algorithms, and $\epsilon = 0.2$ for the DPA. Finally, we assume the MBN starts at $\boldsymbol{M}_0 = (8, 0)$, and denote starting points with a star. From the figure, we can see that early in the time horizon the greedy deviates from the DPA, but because the RN is not moving faster than the MBN it is able to catch up by time step $t = 5$. As the performance ratio plot in figure 6 would indicate, it is up to here that the DPA algorithm seems to be performing better than the greedy algorithm. Specifically, the DPA is worse than the greedy during $0 \leq t \leq 2$, but better for $3 \leq t \leq 5$. For time steps $t \geq 5$ however, the greedy MBN is able to stay exactly on top of the RN, whereas the DPA MBN is restricted to travel between grid points. Again this is reflected in the instantaneous time performance ratio plot, since for time steps $t \geq 5$ the throughput achieved by the greedy algorithm is either as good as the DPA or slightly

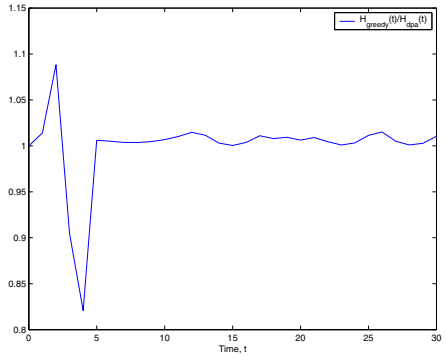

Fig. 6. Evolution of the greedy to DPA performance ratio with respect to time. Plot corresponds to the 2D random waypoint example in Fig. 5.

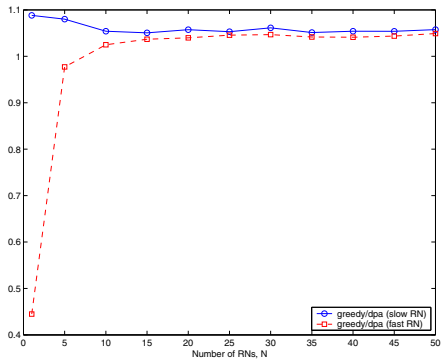

Fig. 7. Average case plot for varying numbers of RNs, over a time horizon $t \in [0, 100]$ and $\Delta t = 1$ for both algorithms.

better. Indeed, in general it would seem that in situations when the MBN can travel at a speed as fast or faster than the RNs the greedy algorithm can perform quite well.

This observation is confirmed by the plot depicted in Fig. 7. It shows the average ratio (over 10 runs) between the MPP-MFPA objective achieved by the greedy to that achieved by the DPA algorithm for varying numbers of RNs over a time horizon $t \in [0, 100]$, with $\Delta t = 1$ and MBN speed bounded by $V = 2$. The mobility model used for the RNs is a random waypoint model over a 50×50 dimension plane, for a "fast RN" and "slow RN" scenario. For the "fast RN" scenario we assume $[V_{min}, V_{max}] = [5, 10]$, and we assume $[V_{min}, V_{max}] = [0.5, 2]$ for the "slow RN" scenario. As can be seen in the plot, for faster speeds and small numbers of RNs, the DPA algorithms outperforms the Greedy algorithm. However, once the number of RNs passes 10, the greedy performs as well or better than the DPA algorithm. This can likely be attributed to the fact that for larger number of RNs, the 1-center is more stable and slower moving, and thus the greedy can get to the exact location even though the MBN is slower than individual RNs. In contrast, the DPA algorithm is limited to an approximate location (i.e. on a grid point). Indeed, for slower RNs the greedy outperforms the DPA algorithm for this reason.

Thus in general the following observations can be made regarding the two solution approaches. First, while the greedy approach can be vulnerable to certain "bad" problem instances (e.g. Fig. 3), it performs well on average. Moreover, consider a problem instance in which the unconstrained 1-center locations at consecutive time steps are always within a distance $V\Delta t$ of each other (e.g. a "slow RN" case). Also assume that M_0 is within $V\Delta t$ the unconstrained 1-center location in the first time step. If this is the case the greedy approach will find the exact optimal solution. By contrast, the DPA algorithm would still only find an approximate solution, since the MBN placements would be restricted to grid points.

VII. Conclusion

In this paper we considered the path planning of a single MBN with the goal of maximizing the time-average system throughput. To this end, we formulated a discrete MBN path planning problem with velocity constraint as the MPP-MFPA problem. We developed two solution approaches: (i) a greedy approach based on an optimal algorithm for the single timestep problem, and (ii) a dynamic programming based approximation algorithm, for which we presented worst case results. Future work includes extending the single MBN formulation to multiple MBNs, as well as a continuous formulation and solution of the MBN path planning problem. Additional work includes considering formulations in which the hard velocity constraint is relaxed (e.g. as a penalty).

References

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

[2] S. Bereg, B. Bhattacharya, D. Kirkpatrick, M. Segal, "Competetive Algorithms for Maintaining a Mobile Center," *Mobile Networks and Applications*, 11, pp. 177-186, 2006.

[3] P. Bose and G. Toussaint, "Computing the constrained euclidean geodesic and link centre of a simple polygon with applications," *Proc. of Pacific Graphics International*, 1996.

[4] D. Erlenkotter, "A comparative study of approaches to dynamic location problems," *Eur. J. Oper. Res.*, 6, pp. 133143, 1981.

[5] S. K. Jacobsen, "An algorithm for the minimax weber problem," *Eur. J. Oper. Res.*, 6, pp. 144-148, 1981.

[6] S. L. Hakimi, M. Labbe and E. Schmeichel, "Locations on Time-Varying Networks," *Networks*, 34, 4, pp. 250-257, 1999.

[7] S. Haykin, "Communication Systems," 4th Ed., Wiley, 2001.

[8] A. Orda and R. Rom, "Location of central nodes in time varying computer networks," *Oper Res Lett*, 10, pp. 143152, 1991.

[9] I. Rubin, A. Behzad, R. Zhang, H. Luo, and E. Caballero, "TBONE: a mobile-backbone protocol for ad hoc wireless networks," in *Proc. IEEE Aerospace Conf.*, Mar. 2002.

[10] A. Srinivas, "Mobile Backbone Architecture for Wireless Ad Hoc Networks: Algorithms and Performance Analysis," *PhD Thesis*, Massachusetts Institute of Technology, June 2007.

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

[12] A. Srinivas, G. Zussman, and E. Modiano, "Mobile Backbone Networks: Construction and Maintenance," *ACM MOBIHOC'06*, May 2006.

[13] G.O. Wesolowsky, "Dynamic facility location," *Mgmt Sci*, 19, pp. 12411248, 1973.

[14] G.O. Wesolowsky and W.G. Truscott, The multiperiod location- allocation of facilities, *Mgmt Sci*, 22, pp. 5765, 1975.

[15] K. Xu, X. Hong, and M. Gerla, "Landmark routing in ad hoc networks with mobile backbones," *J. Parallel Distrib. Comput.*, 63, 2, pp. 110-122, 2003.