Path Planning for Network Performance

Lex Fridman*, Steven Weber*, Charles Graff†, Moshe Kam*

*Drexel University, Philadelphia, Pennsylvania 19104

†CERDEC U.S. Army RDECOM, Fort Monmouth, New Jersey 07703

Abstract— **Path planning and network design are often treated by architects of mobile communication networks as separate problems. In fact, most mobile ad hoc network (MANET) designs do not consider the path that the network nodes would take as part of the objective set, but incorporate them in an abstract form as general constraints on mobility (limit on the initial configuration and the node velocity). It appears that significant performance improvement may be possible if multiobjective optimization is attempted on performance indices such as connectivity of the network and arrival time of all nodes at a specified terminal set. In this study we provide an efficient decentralized method for computing the Pareto optimal set of plans, and show the benefit of path planning under six different metrics of MANET performance.**

I. Introduction

It is conventional when considering mobile networks to treat user mobility as exogenous, and to seek network designs that perform adequately regardless of user positions. This paradigm of movement is often natural and appropriate: most of the time getting to our destination in the shortest possible time trumps the desire to stay connected to the network. Yet with our ever increasing dependence on and expectation of ubiquitous connectivity, it is ever more conceivable that connectivity concerns may guide our path planning decisions. Given the choice between a shortest path that suffers frequent network outages and a longer path that offers continuous connectivity, the designer may choose the latter if the sacrifice in travel time is not too great.

In this paper we explore the interplay between communication and path planning in mobile ad hoc wireless networks. In our setting, at each point in time, each node receives feedback on the "communication value" of each of the different paths it might take in seeking its final destination. In particular, we employ a discrete space model, where each node incorporates knowledge of how each of its feasible next "hops" would affect the communication pattern on the network. Each node is therefore balancing the potentially competing objectives of minimizing the time to destination (by taking the minimum-time path) and maximizing "information flow" (by following the path most advantageous to communication). Nodes are assumed cooperative: even relay nodes, who are not the source or sink for any data, will attempt to balance these two objectives. Our results demonstrate that incorporation of path planning in the network design process can in some cases improve network performance significantly.

II. Related Work

This paper bridges the fields of network design and path planning by allowing movement to be a controllable resource in the optimal allocation process. The pertinent problem of path planning under geometric constraints has received significant attention in literature, most notably under the topic of formation control (e.g. [1], [2]). In general, formation control involves two steps. First, the nodes move into a geometric formation with either hard or soft constraints on the type of formation; then these nodes attempt to maintain their geometric relationship to each other while moving and avoiding obstacles. The techniques of formation control generally do not scale in the number of nodes, do not consider travel time as a significant objective, and do not use network performance metrics for models of the relationships between nodes. Moreover, when the formation requirement is formulated as a communication metric (e.g. [3], [4]), it is considered only as a hard constraint. The key difference of our work is that we can achieve significant improvement in network performance when connectivity is *sparse*, which is only possible when communication is used in optimization as an *objective*. We combine this objective with a conflicting goal of travel time, which allows us to form a trade-off curve between the two metrics. To get the solution produced by formation control, we simply look at the solution on the trade-off curve that allows unlimited travel time. In [5], the authors propose a method for communication-based movement planning. The key difference of this paper with ours is that it assumes a kinematic model that is constrained to a predetermined path and it only considers one basic metric of connectivity, while this paper considers six more complex and realistic models of network performance.

III. Movement and Communication Model

In the subsequent discussion we adopt commonly used models from [6], [7] and use notation that is consistent with [5].

A. Movement Graph

In order to discretize traversability information provided by the presence of obstacles, boundaries, and kinematic restrictions in an arena, a movement graph $G_M(V_M, E_M)$ is formed. V_M is a set of vertices, each of which indicates a state that can be occupied by a single network node. The edges E_M indicate a set of feasible movements in the arena. Specifically, $\Lambda(i) \equiv \{j : (i,j) \in E_M\}$ is the set of neighbors of i. Each step has an associated travel time of one time unit.

B. Mobility Model

The mobile network consists of a set n nodes. Each node $i = 1, 2, ..., n$ in the network has a movement plan $\mathbf{P}_i$ that is

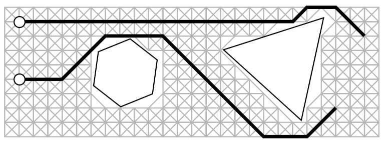
Fig. 1. A network of two nodes, each with a movement plan. The polygons are obstacles, grey lines are movement graph edges, circles are nodes, and bold black lines are movement plans.

an ordered set of vertices in V_M. $P_i(t)$ is the tth element in set $\mathbf{P}_i$. In other words, $P_i(t)$ is the position of node i at time $t = 0, 1, ..., |\mathbf{P}_i|$. Figure 1 shows an example of two nodes, each with a movement plan.

Time is discrete, and movement is assumed to be synchronous. Therefore, the motion planning problem involves a set of $|\Lambda(i)| + 1$ choices for each node at each state i of the network. The additional choice is the decision to remain in the same state.

The time it takes for a node i to travel from its origin to its destination is denoted as T_i. We refer to the sequence of choices in this paper as a *plan* or *policy*. $T_{\max} = \max\{T_i : i = 1, 2, ..., n\}$ designates the length of the longest plan, which is also the time to completion of the scenario. The term *scenario* is used to denote the set of n plans as a whole. After time $t = T_i$, node i remains in the same position until $T_{\max}$. It continues to transmit and receive messages during this period.

C. PHY and MAC Models

We use the PHY and MAC model abstraction defined in [6]. Specifically, scheduling is done in the temporal domain, where the transmissions are synchronized. Each schedule is a choice of transmitters with associated powers, and the set of schedules form a cycle which is looped continuously. Every node i has an associated fixed transmission power P_i, and scheduling is performed such that in one cycle each node gets at least one opportunity to transmit. We use the random packing procedure in [6] to find a set of schedules, as shown on the left hand side of Figure 2. This procedure picks a set of schedules that (1) guarantee each node at least one chance to transmit and (2) guarantee a sufficient distance between concurrently transmitting nodes for signal interference to be acceptably low. On the right hand side of the same figure is the graph $G_F = (V_F, E_F)$ formed by multiplexing these schedules together. That is, if in one of the schedules the node i can successfully communicate with node j, edge (i, j) is added to G_F with a capacity corresponding to the sum of capacities on each such feasible instance of communication.

The control parameters of the PHY graph are (1) the fixed power P_i on each node i; (2) the communication radius R_C which defines an estimated range of feasible communication; and (3) the minimum acceptable signal-to-interference-plus-noise (SINR) ratio γ which defines the quality-of-service requirement for the channels. Given these values, the random packing procedure adds transmitters to a time slot incrementally such that the SINR requirement is not violated for any of the links in E_P. SINR is computed by:

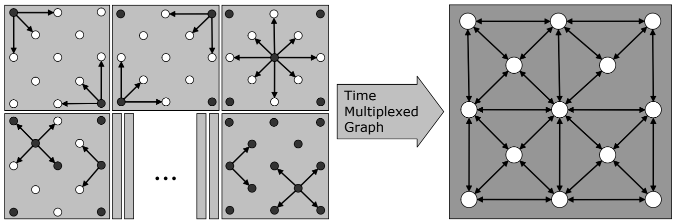

Fig. 2. The process of forming a capacitated graph from a set of schedules.

$$\text{SINR}_{ij} = \frac{P_i A_{ij}}{\sum_{k \neq i} P_k A_{kj} + \eta}, \tag{1}$$

where SINR_{ij} is the SINR on the channel between transmitter i and receiver j, η is the channel noise power which is assumed to be receiver-independent. A_{ij} is attenuation between transmitter i and receiver j defined as $A_{ij} = \left(\frac{d_{\text{ref}}}{d_{ij}}\right)^{\alpha}$, where d_{ij} is the distance between nodes i and j, d_{ref} is a channel reference distance, and α is the attenuation exponent.

The result of the packing procedure is a collection of graphs $\mathbf{G}_P \equiv \{G_P^1, G_P^1, \ldots, G_P^m\}$ where $m \leq n$. All edges (i, j) in graphs of $\mathbf{G}_P$ satisfy the requirement that $\text{SINR}_{ij} > \gamma$. The capacity on each of those edges is determined in one of two ways: (1) capacity of 1 on all feasible edges, or (2) SINR-dependent Shannon capacity defined as follows:

$$C_{ij}(G_P^z) = B_{ij}(G_P^z) \log_2 \left(1 + \text{SINR}_{ij}(G_P^z)\right), \tag{2}$$

where B_{ij} is the bandwidth on edge (i, j) which is assumed to be 1, effectively setting capacity to the spectral efficiency of the channel, C_{ij} is the capacity on edge (i, j), and $z = 1, 2, \ldots, |\mathbf{G}_P|$ is the index of the physical graph. Note that B, C, and $SINR$ are shown as functions of a specific graph. This is used to distinguish the capacities and SINR on individual physical graphs from the single multiplexed flow graph. Unless otherwise specified, capacity C_{ij} refers to the flow graph capacity $C_{ij}(G_F)$.

The multiplexing procedure is additive for capacity, that is, the capacity of an edge (i, j) in the flow graph G_F is computed from $\mathbf{G}_P$ by:

$$C_{ij}(G_F) = \sum_{z=1}^{|\mathbf{G}_P|} C_{ij}(G_P^z). \tag{3}$$

The result of the multiplexing procedure is a capacitated flow graph $G_F = (V_F, E_F)$ that has an edge $(i, j) \in E_F$ if and only if $C_{ij} > 0$.

D. Multicommodity Flow

Let there be K commodities, each commodity specified by a source node $\sigma_k \in V_F$, and a destination node $\delta_k \in V_F$. A commodity specifies a stream of unique data. A *throughput vector* $\mathbf{f} = \{f_1, \ldots, f_K\}$ is *feasible* if it respects both network capacity constraints and conservation of flow constraints. The objective is is to maximize the sum throughput summed over the K commodities. Each commodities throughput may be split across multiple paths; we define a *flow* as the set $\{x_e^k, e \in E_F, k \in [K]\}$, so that x_e^k is the flow for commodity k on edge e. The max multicommodity flow problem is:

$$\begin{aligned} \max_{\mathbf{x}} \quad & F(\mathbf{f}) = \sum_{k=1}^{K} f_k \\ \text{s.t.} \quad & \sum_{i \in V_M} x_{i,\delta_k}^k = f_k,\ k \in [K] \\ & \sum_{k \in [K]} x_{ij}^k \le c_{ij},\ i \in V_F, j \in V_F \\ & \sum_{j \in V_M} x_{ji}^k = \sum_{j \in V_M} x_{ij}^k,\ k \in [K], i \in V_F \end{aligned} \tag{4}$$

The first line states the objective is to maximize the sum commodity throughput. The first constraint defines the throughput for each commodity as the sum of the flow over all edges which terminate in commodity k's terminal node δ_k. The second constraint is the capacity constraint; the third is the conservation of flow constraint.

E. Optimization Formulation

The path planning problem we define is a maximization of a global network performance measure under the constraints of a fixed travel time.

The network performance is an average over the duration of the scenario. In this paper we define six objectives which are described in detail in Section IV. Therefore, the objective value is the measure of network performance for the path divided by the global time to completion $T_{\max}$. We use $N(\mathbf{P}(t))$ to denote the function which computes the network performance given the position of the nodes $\mathbf{P}(t)$. The movement planning problem can be formulated as follows:

$$\begin{aligned} \max \quad & \frac{1}{T_{\max}+1} \sum_{u=0}^{T_{\max}} N(\mathbf{P}(u)) \\ \text{s.t.} \quad & P_i(t) = g_i \quad |\mathbf{P}_i| \le t \le T_{\max}, \quad i = 1...n \\ & P_i(0) = o_i \quad i = 1...n \\ & (P_i(t), P_i(t+1)) \in E_M \quad i = 1...n \end{aligned} \tag{5}$$

where E_M, as defined previously, is the set of all possible single-step movements allowed in the lattice graph G_M.

The key decision variable of this problem is the set of paths $\mathbf{P}$. The key control parameter is $T_{\max}$. For a given instance of a movement planning problem, the increase of $T_{\max}$ is likely to result in an increased improvement in performance in the average case. Therefore, the optimization problem defined in (5) is solved for $T_{\max} = T_{\min}, T_{\min}+1, T_{\min}+2, \ldots$, where $T_{\min}$ is the duration of the scenario if all the nodes march down their respective paths without stopping and with no regard to any communication considerations. The solutions to each of these problems considered against $T_{\max}$ form a Pareto-optimal front [8]. $T_{\max}$ is a secondary objective in that we may wish to minimize the time to completion. Therefore, the Pareto front allows one to choose between faster travel time or better network performance.

IV. Network Performance Objectives

We define three categories of metrics of network performance and within each category specify two objectives. Unless otherwise stated, these metrics measure performance on the multiplexed flow graph G_F. In some cases, these metrics will be referred to by their respective section: A1, A2, B1, B2, C1, C2.

A. Connectivity

1) Connected Components: This is the number of clusters of nodes where a message from any member in that cluster can reach any other member inside the same cluster. This value is computed using Kosaraju's algorithm [9]. The range of values for this metric is $\{1, 2, ..., n\}$.

2) Isolated Nodes: This is the number of nodes that do not have feasible communication with any other nodes. This happens when a node does not have any neighbors in a disk area of radius R_C.

B. Capacity

1) Link Density: This is the number of links in all the graphs of $\mathbf{G}_P$ combined. Another interpretation of this metric is the sum-rate capacity of the network under the constant unit capacity model.

2) Shannon Capacity: This is the sum-rate capacity of G_F under the Shannon capacity model as defined in (3).

C. Throughput

1) SINR-Independent Flow: This is the value of the maximum multicommodity flow defined in (4) with the flow graph G_F formed under the constant unit capacity model.

2) Shannon Capacity Flow: This the SINR-dependent throughput but with the flow graph G_F formed under the Shannon capacity model.

V. Search Methodology

In a system of nodes, a single node makes an intelligent decision about its future movement by optimizing its path based on a well-defined communication or movement objective. This planning procedure is a distributed procedure in a sense that it does not consider cooperative movement. However, it does assume the availability of a centralized heuristic function which is able to measure the global utility of a network state.

The search for the minimum-cost path is performed by a variation of the A* search procedure [10], [11] as shown in Algorithm 1, **`HeuristicSearch`**$(G_M, s, t, T_{\max}, g, h)$. Here, $G_M(V_M, E_M)$ is the finite movement graph described previously, s is the position of the node immediately before planning begins, and t is the goal position. Also, $T_{\max}$ is an optional parameter which provides a hard constraint on how long the path is allowed to be. Finally, $g(v)$ and $h(v)$ define an estimate of the current value and predicted value, respectively, of a vertex v (in the context of a path) in the movement graph. For convenience of pseudocode expression, g is overloaded as:

$g : V \to \mathbb{R}$, $g : (V, V) \to \mathbb{R}$, and $g : \mathcal{P}(V) \to \mathbb{R}$ where $\mathcal{P}(A)$ is a power set of V. $f(v) = g(v) + h(v)$ is the cost function which combines the two heuristic functions, $g(v)$ and $h(v)$, to make a decision about which vertices to investigate as possible candidate members of the minimum-cost path.

HeuristicSearch is a planning procedure for a single node. It keeps three lists: (1) Open, (2) Closed, (3) Unvisited. Together, without intersection, these lists contain all nodes in V_M. The algorithm starts by adding the starting position s to the open list. It then adds all the neighbors of s to the open list, while adding s to the close list. Next, it adds the children of the lowest-cost state BEST in the open list to the open list, and adds BEST to the closed list. This procedure is repeated until the destination t is added to the closed list. Every time a node is put in the open list, it is removed from the unvisited list.

Algorithm 1 HeuristicSearch

```
Input: $G_M(V_M, E_M); s, t \in V_M; T_{\max}; g, h : V_M \to \mathbb{R}$
  OPEN ⇐ {s} // OPEN is the open list
  CLOSED ⇐ ∅ // CLOSED is the closed list
  while OPEN ≠ ∅ do
    BEST ⇐ $\arg\min\{f(v) : v \in V_M\}$
    OPEN ⇐ OPEN \ BEST
    CLOSED ⇐ CLOSED ∪ BEST
    if IsGoal ( BEST ) = true then
      return Path(BEST)
    end if
    if |Path(BEST)| > $T_{\max}$ then
      continue
    end if
    CHILDREN ⇐ Children(BEST) ∩ Path(BEST)
    for all $c \in$ CHILDREN do
      if $c \notin$ (OPEN ∪ CLOSED) then
        OPEN ⇐ OPEN ∪ $c$
      else if $g$(Path($c$)) > $g$(Path(BEST)) + $g$(BEST, $c$) then
        $c$.parent ⇐ BEST
        if $c \in$ CLOSED then
          Update($c$)
        end if
      else
        CHILDREN ⇐ CHILDREN \ $c$
      end if
    end for
    BEST.children ⇐ CHILDREN
  end while
  return failure
```

The function **IsGoal** determines if the state passed to it is a goal state. The function **Children(**v**)** $\equiv \{w : (v, w) \in E_M\}$ generates all the positions in the movement graph which are reachable from the current position. It is independent of the execution state of the algorithm. Specifically, it only represents the discrete kinematic information formed from the global knowledge of the world. The function **Update(**v**)** recursively updates the value of each member in a node's family tree. This is a computationally important, but often skipped step, which is done when $v \in$ CLOSED and is chosen to be "reopened". The function **Path(**v**)**, determines the steps that lead to a state v from s by recursively tracing backwards along the chain of *parent* pointers until a null pointer is reached. Specifically, **Path**$(v) \equiv v \cup$ **Path**$(v.\text{parent})$, and **Path**$(\emptyset) = \emptyset$.

Algorithm 1 maintains a list of *open* and *closed* movement nodes V_M which ensures that a state is not explored more than once unless it appears to be better than originally estimated. The *open* list efficiently represents (in our case as a binary heap) the search frontier that is used in making best-first expansions of the search graph.

VI. Simulation and Results

We present performance improvement against the time allow for the nodes to move to the destination over the shortest possible duration of the scenario, or $T_{\max} - T_{\min}$. Intuitively, the more time the nodes are given, the greater the improvement of average network performance.

A. Simulation Setup

All simulations were performed on an arena of 100×100 m^2, with 20 nodes. The origin and goal for each node are placed independently and uniformly at random over the arena. Each data point on the presented plots is an average of 5,000 simulations.

The range of communication requirement R_C is chosen using a parameter $m \geq 0$ that is scalable in the number of nodes and the size of the world [12]: $R_C = \frac{m}{\pi}\sqrt{\frac{A \log n}{n}}$, where A is the area of the arena and n is the number of nodes. When $m \gg 1$ the network is well connected and motion planning is unlikely to improve the network performance, unless most of the arena is covered with obstacles. When $m \approx 1$, the network is likely to be partially disconnected and therefore motion planning is expected to show improvement. When $m \ll 1$, the granularity of the movement graph is greater than R_C and therefore motion planning is likely to be unable to improve connectivity. We used a value of $m = 0.5$ which in general results in poor network performance if movement is not optimized.

The sources and sinks for different commodities of flow are chosen randomly at the start of the scenario and kept constant throughout the scenario. In simulation, there are two commodities, and thus, out of 20 nodes, 16 are designated as relay nodes.

B. Improvement

Improvement is defined as $N^*/N - 1$, where N^* is the average network performance with path planning, and N is the average network performance of the naive movement policy of marching down the shortest path without consideration of communication objectives. Therefore, this improvement metric measures the relative increase in performance provided by path planning, with 0 indicating no improvement, 1 indicating 100% improvement, etc.

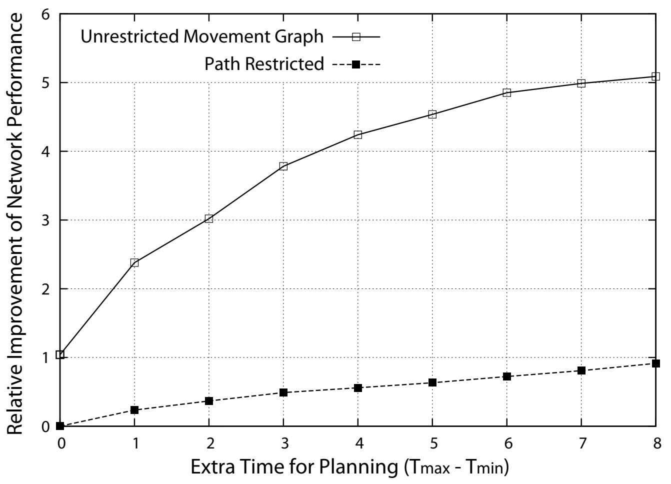

Fig. 3. Comparison of network performance improvement from two different models of path planning. The first method is restricted to a prespecified plan. The second method is restricted only by the movement graph. The measure of network performance for this plot is the number of connected components.

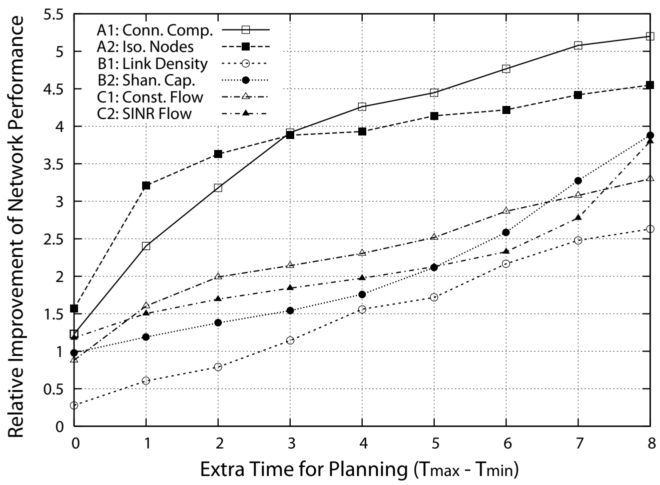

Fig. 4. The improvement in six different metrics of network performance attained by path planning with respect to increasing planning time. The six metrics used here are described in Section IV.

C. Path Restricted vs. Movement Graph Restricted

When nodes are forced to travel along a fixed path, and their only choices are to step forward, stay in place, or step back, we expect to see less improvement in network performance than when the nodes can move without restriction on a general movement graph. The former model is considered in [5], where improvement is shown in terms of the number of connected components. Figure 3 compares the improvement attained from the path restricted and the movement graph restricted models of movement. On the x-axis of this plot is the extra time allowed for planning. On the y-axis is the relative improvement of the plan after optimization as compared to a shortest-path plan. To be clear, the plot shows that the movement graph method provides 100% improvement for when zero extra time over the duration of a shortest path is allotted to the nodes, and as high as 500% improvement for $T_{\max} - T_{\min} = 8$.

D. Improvement Under Different Measures of Performance

We define six measures of network performance (see Section IV), each of which are used as objectives in the optimization problem defined by (5). Figure 4 shows a improvement of network performance under all six metrics. An interesting insight shown in the plot is that when $T_{\max} - T_{\min} = 0$, under all metrics, path planning provides a large improvement. This is the point of no cost in terms of time. In other words, if scenario duration is a critical cost, path planning can provide improvement in network performance, without increasing the cost associated with longer travel times.

VII. Conclusion

Movement is a key variable for highly-dynamic communication networks, and its optimal allocation proves to be effective at improving the network performance. We develop an efficient decentralized path planning method and show its ability to improve communication as measured by six metrics based on realistic models of ad-hoc wireless networks. The key assumption of our method is that the kinematics of the system can be modeled by a finite movement graph. This is the input to the optimization engine. The output is the Pareto optimal set of plans for the trade-off between scenario duration and network performance.

References

[1] P. K. C. Wang, "Navigation strategies for multiple autonomous mobile robots moving in formation," in *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Tsukuba, Japan, September 1989, pp. 486–493.

[2] G. A. S. Pereira, A. K. Das, R. V. Kumar, and M. F. M. Campos, "Decentralized motion planning for multiple robots subject to sensing and communication constraints," in *Proceedings of the 2003 International Workshop on Multi-Robot Systems*, 2003, pp. 267–278.

[3] R. W. Beard and T. W. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003, pp. 25–30.

[4] D. P. Spanos and R. M. Murray, "Motion planning with wireless network constraints," in *Proceedings of the 2005 American Control Conference*, Portland, OR, June 2005, pp. 87–92.

[5] L. Fridman, S. Weber, and M. Kam, "Communication-based motion planning," in *Proceedings of the 41st Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, March 2007.

[6] Y. Wu, P. Jain, and K. Kung, "Network planning in wireless ad hoc networks: a cross-Layer approach," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 1, pp. 136–150, 2005.

[7] V. Davies, "Evaluating mobility models within an ad hoc network," Master's thesis, advisor: Tracy Camp, Dept. of Mathematical and Computer Sciences.Colorado School of Mines, 2000.

[8] Y. Censor, "Pareto optimality in multiobjective problems," *Applied Mathematics and Optimization*, vol. 4, no. 1, pp. 41–59, 1977.

[9] R. Tarjan, "Depth-First Search and Linear Graph Algorithms," *SIAM Journal on Computing*, vol. 1, p. 146, 1972.

[10] N. J. Nilsson, *Artificial Intelligence: Aew Synthesis*. San Francisco: Morgan Kaufmann Publishers, 1998.

[11] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 2003.

[12] P. Gupta and P. Kumar, "Critical power for asymptotic connectivity in wireless networks," *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of WH Fleming*, pp. 547–566, 1998.