2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Distributed Path Planning for Connectivity Under Uncertainty by Ant Colony Optimization

Alex Fridman*, Steven Weber*, Vijay Kumar†, Moshe Kam*
* Drexel University, Philadelphia, Pennsylvania 19104
† University of Pennsylvania, Philadelphia, Pennsylvania 19104

***Abstract*— Movement and allocation of network resources for a system of communicating agents are usually optimized independently. Path planning under kinematic restrictions and obstacle avoidance provides a set of paths for the agents, and given the paths, it is then the job of network design algorithms to allocate communication resources to ensure a satisfactory rate of information exchange. In this paper, we consider the multiobjective problem of path planning for the sometimes conflicting goals of fast travel time and good network performance. In previous work we considered this problem under the assumption of full knowledge of network topologies and unlimited computational resources. In this paper, nothing is known a priori about topology, information is exchanged between nodes within a connected component of the network, and sources of environment-dependent communication failure can only be approximately estimated through learning. All the planning must be done online in a distributed fashion. We apply ant colony optimization to this problem of planning under uncertain information, and show that significant benefit in network performance can be achieved even under the difficult conditions of the scenario. Furthermore, we show the ability of nodes to quickly learn the communication patterns of the arena, and use this information for improved path planning.**

I. INTRODUCTION

The scenario of multi-agent systems moving on a predetermined mission with a high value attached to maintaining acceptable network performance is an important and well studied problem in multi-robot systems, and more generally in systems of semi-autonomous vehicles which have the ability to transmit and receive signal. The general approach to such a scenario is to decouple the movement planning from the problem of optimal network resource allocation. However, there are cases when the only resource which can guarantee good communication is movement. For example, when the scenario arena contains jamming nodes which transmit continuously at full power, thus preventing communication in the surrounding area by causing overwhelming interference, controlling the allocation of power, modulation, scheduling, or routing protocols will not help improve network performance. The most effective way of improving communication in this case is to avoid the areas affected by the jammers. Another example is when nodes have highly constrained power resources, as is common for mobile systems that run on batteries. Moving in closer together while still reaching the waypoints of the mission, in this case, is a lower cost solution than increasing transmission power to cover a greater range of communication.

When network performance is the primary objective, movement is the primary controllable resource, and the main threats to communication are jammers and landmarks in the physical space which cause poor signal propagation, the optimization problem is similar to the tradition path planning problem with one key difference. The movement plan must avoid obstacles not in the physical medium but in the communication medium. This is not a big difference under the assumption of complete knowledge, that is, if it is assumed that every node has exact information about the location of jammers and bad areas for communication a priori. However, if it is assumed that nothing is known initially by the network nodes, and the location of network performance threats must be mapped through exploration while still reaching the destination of the mission with no significant cost to travel time, the problem becomes much more difficult. One of the sources of difficulty is that the areas of poor communication cannot be easily mapped from sensed information until many nodes encounter that area under different configurations. Therefore, the learning process is slower, because of the high degree of uncertainty in deducing knowledge of the environment from knowledge of the local network state.

II. RELATED WORK

The defining *objective* of the optimization problem in this paper is connectivity. The defining aspect of the *model* we propose is uncertainty and incompleteness of state knowledge. The defining *method* which is used to construct a solution to the problem is ant colony optimization. These three defining elements have been thoroughly studied in literature individually but, to the best of our knowledge, have not been bridged together to plan movement in support of network performance under uncertain state knowledge.

A. Path Planning for Connectivity

The problem of movement planning under geometric constraints has been thoroughly studied, most notably under the topic of formation control. Wang [1] studies the use of nearest neighbor tracking to maintain formation. Their method has the important qualities of being simple, computationally efficient, and distributed. However, it does not directly consider collision and obstacle avoidance. Desai, et al., [2] use methods of feedback linearization to stabilize the distance of a follower node to a leader node. They further investigate this problem in [3], proposing a graph theoretic framework for the coordination of transitions from one formation to another. Feddema, et al. [4] apply decentralized control theory to the analysis of cooperating robot motion, proving stability for a fixed topology configuration. Yamaguchi proposes in [5] and [6] an elegant approaches to formation control by using a formation vector with smooth feedback control. Formation control is a basis for the more recent efforts, more focused on the motion planning under communication constraints. Beard and McLain [7] propose a dynamic programming method for the cooperation motion planning under communication constraints that is polynomial in the number of nodes but exponential in the depth of the lookahead window. The key characteristic of this method relative to our proposed methods is that the limited range connectivity is a hard constraint, while in our method it is an objective. The hard constraint is a prerequisite which makes their method inapplicable to the more general model considered in this paper. Pereira, et al., [8] present a decentralized method of motion planning under communication constraints. One of the defining features of their method is that it first seeks to achieve connectivity, and then move to the goal positions in such a way as to not break connectivity, which is essentially the problem of formation control. Spanos and Murray [9] define a connectivity robustness metric, which uses conservative connectivity constraints to attain good network performance despite the algorithm's distributed structure with minimal added limitations on physical reachability. The key difference of this work from our paper is that Spanos and Murray view connectivity as a constraint while we view it as an objective, which is more general in that it seeks to also optimize communication for cases when connectivity is sparse.

B. Path Planning Under Incomplete Knowledge

There are two models of incomplete knowledge in path planning which are most prevalent in literature. The first is the presence of uncertainty in movement, sensing, or environment dynamics. As an example, Bicho and Monteiro [10] use nonlinear attractor dynamics to model nonholonomic formations of robots. The key contribution of their work is the ability of robots to maintain robust formations in the face of unknown environmental perturbations. The common approach to this model of incomplete knowledge is that of stochastic or robust optimization.

The second model of incomplete knowledge is an initial lack of information about the environment. One of the planning goals for this model is that of mapping: exploring the environment and map the location obstacles. [11] proposes a framework for representing an unstructured environment as estimated by sensor data fusion during discovery. Many papers show specialized improvement of approaches to the mapping problem, such as estimate error correction [12], multi-resolution 3D mapping [13], landmark-based map matching [14], etc. The key distinction of our work is that the obstacles which we are mapping are in the communication medium not in the physical medium and are significantly more difficult to deduce from sensed information.

C. Path Planning by Ant Colony Optimization

Ant Colony Optimization (ACO) is designed for complex optimization problems on large graphs and so it is a natural fit for path planning on a movement graph. [15] presents a method of avoiding mobile obstacles with genetic algorithms used to tune the ACO parameters. Many other applications of ACO to the classical path planning problem have been proposed [16], [17]. In general many of the ACO application propose a hybrid algorithm which combine ACO with another method, such as potential fields in [18]. However, to the best of our knowledge, ACO has not been applied to network-centric planning or planning under incomplete knowledge.

III. Movement and Communication Model

The model of movement and communication is based on previous work published by the authors in [19] and [20], with common notation for common elements. We define two classes of graphs. The first class is represented as G_M which is defined on the set of reachable positions in the 2D physical space, and the second class is represented as G_F which is defined on the set of mobile network nodes.

A. Movement Graph

The kinematic constraints of the physical space provided by the presence of obstacles, boundaries, and capabilities of mobile agents in an arena are defined by a movement graph $G_M(V_M, E_M)$. V_M is a set of vertices, each of which indicates a state that can be occupied by a single network node. Movement is synchronized with a constant-magnitude time step. Thus, the edges E_M indicate a set of feasible movements in the arena. Specifically, the set $\mathbf{\Lambda}(u) \equiv \{v : (u,v) \in E_M\}$ is all the positions in the movement graph which are reachable from position u. We will use u, v, and w to denote indices into V_M. This is in contrast to indices into V_F, which are denoted as i, j, and k.

B. Mobility Model

The mobile network consists of n nodes. Each node i in the network has a set of movement plans $\mathbf{\Pi}(i) \equiv \{\mathbf{\Pi}(i,1), \mathbf{\Pi}(i,2), \ldots, \mathbf{\Pi}(i,Q)\}$. Each plan q has an associated probability $\rho(i,q)$ of being chosen as the plan that is followed by the node at the current time. The plan $\mathbf{\Pi}(i,q)$ is an ordered set of vertices in V_M. $\Pi(i,q,t)$ is the tth element in set $\mathbf{\Pi}(i,q)$. In other words, $\Pi(i,q,t)$ is the position of node i at time t for $t \in \{0, 1, ..., \|\mathbf{\Pi}(i,q)\|\}$ according to the qth plan. A node maintains a set of plans through the scenario. For expressive convenience, at some time t_c, $\Pi(i,q,t)$ is still defined for $0 \le t \le t_c$. In this case it returns the information about where the node was actually located at time t. We use $\mathbf{\Pi}^c(i)$ (c for chosen) to denote the plan that is chosen to be followed by node i. This is not necessarily the optimal path.

Time is discrete, and movement is assumed to be synchronous. Therefore, the motion planning problem involves a set of $\|\mathbf{\Lambda}(u)\| + 1$ choices for each node at each state u. 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 only time a node may have an empty plan is when it is located in its goal state.

We consider two movement policies. The first we refer to as the "naive" policy, in which each node chooses the shortest path to its destination and marches along that path without any consideration of network performance. The second policy is the "intelligent" policy where each node dynamically plans its path after each step based on its knowledge of the network state and the physical space.

C. Connectivity

At a snapshot in time, the network of n nodes is described as a connectivity graph $G_F = (V_F, E_F)$. Two nodes $i \in V_F$ and $j \in V_F$ are connected if $(i,j) \in E_F$. This edge is added to E_F if the distance d_{ij} between the two nodes is less than a predetermined constant $d_{\max}$. The graph G_F can be considered undirected because of the symmetric property of the distance requirement.

The metric for network performance that is used in planning is the number of connected components. A connected component is a set of nodes in V_F such that there exists a path between any two vertices in that component. We use $N(G_F)$ to denote the function which computes the number of connected components in a network state as represented by a connectivity graph G_F.

D. NoComm Zones

In addition to the distance requirement for connectivity between two nodes, there are sources external to the network of G_F that affect the ability of two nodes to communicate. For example, a jammer node transmitting continuously at full power will prevent nodes around it from communicating. Another example is an area of the physical space, such as a building, may have certain signal propagation characteristics which prevents nodes inside the area from communicating with each other and with nodes that are outside the area.

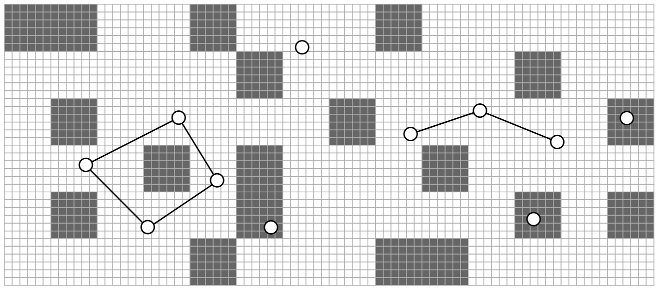

Fig. 1. Example of a movement lattice (light gray squares), nocomm zones (large dark gray squares), and 11 V_F nodes (circles). The lines between nodes indicated edges in E_F. There are 6 connected components in this figure. Three of the nodes are isolated because they are located inside a nocomm zone.

We model this environment phenomena by selecting connected components in G_M to be "nocomm" zones. Specifically, we define a function $J : (u \in V_M) \rightarrow \{0, 1\}$ which is 1 if a location u is part of a nocomm zone, and 0 otherwise. Therefore, two nodes are connected if $d_{ij} < d_{\max}$ and if neither node is inside a nocomm zone. Figure 1 shows an example of the discretized physical space with nocomm zones and the resultant connectivity of the network. The density and shape characteristics of nocomm zones is a key parameter in testing the effectiveness of the algorithms presented in this paper. When the density of nocomm zones is low, it is expected that network performance throughout the scenario will be good, and when the density of nocomm zones is high, even with optimal path planning, network performance will be poor. A crucial goal of the paper, therefore, is to achieve good network performance at a medium-level density of nocomm zones.

E. Knowledge Uncertainty

Initially the location of nocomm zones is unknown, and the network topology outside of a node's connected component is unknown to that node. It is assumed that nodes within a connected component can exchange information without restrictions. This is a reasonable assumption given the previously defined requirements for connectivity, and the fact that network topology changes at a rate that is orders of magnitude slower than the rate of communication. Therefore, the nodes inside a connected connected component have current information about each others paths. However, they only have outdated information about the paths of nodes that are outside the connected components. The fusion of this information in making movement decisions is a key element of the planning problem and is discussed in Section IV-C. We denote node i's estimate of node j's location at time t as $\hat{\Pi}_i(j, q, t)$ assuming that node chooses path q.

As the nodes move in the arena they can estimate the locations of nocomm zones around them and share that information with each other. Each individual node $i \in V_F$ estimates the function $J(u)$ for location $u \in V_M$ by maintaining a probability $\lambda_i(u) = \mathbb{P}_i(J(u) = 0)$ that the location u is not part of a nocomm zone and the certainty $\zeta_i(u) = [0, 1]$ with which node i believes the $\lambda_i(u)$ estimate to be correct. Initially, all nodes have no knowledge, therefore for all $i \in V_F$ and all $j \in V_M$, the certainty $\zeta_i(u) = 0$. When the world static (i.e. nocomm zones do no move), then $\lambda_i(u) \in \{0, 1\}$. In the more general case, however, $\lambda_i(u) = [0, 1]$.

IV. Optimization Framework

A. Communication-Based Planning

The optimization objective is to find a set of movement plans which minimizes the number of connected components averaged over the scenario duration. Each node i starts at a location $o_i \in V_M$ and must reach a destination $g_i \in V_M$. The optimal solution to this problem without constraints is a set of infinite plans where the nodes find the configuration for the best network state and stay in that configuration. As the plan length tends to infinity, the average number of connected components tends to the number of components in that configuration. Therefore, the conflicting objective of scenario duration is required. A set of Pareto optimal solutions is found by adding a hard constraint on the scenario duration and then solving the optimization problem for a set of durations.

The network performance is an average over the duration of the scenario. Therefore, the objective value is the measure of network performance for the plan divided by the global time to completion $T_{\max}$. We use $N(\mathbf{\Pi}^c(t))$ to denote the function which computes the network performance given the position of the nodes $\mathbf{\Pi}^c(t)$. Note that this implies a direct mapping from $\mathbf{\Pi}^c(t)$ to G_F. This mapping includes the computation of E_F as described in Section III-C. The movement planning problem can be formulated as follows:

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

Where E_M, as defined previously, is the set of all possible single-step movements allowed in the lattice graph G_M, and $T_{\max}$ is the constraint on the duration of the scenario.

The key decision variable of this problem is the set of paths $\mathbf{\Pi}^c$. 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 (1) is solved for $T_{\max} = T_{\min}, T_{\min}+1, T_{\min}+2, \ldots$. The solutions to each of these problems form a Pareto-optimal front [21]. $T_{\max}$ 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.

B. Knowledge Estimation

1) Node Position: A node in network G_F has up-to-date information about the location and paths of all the nodes that are in its connected component. Using this information it estimates the position of those nodes some time in the future based on the chosen paths of those nodes. For example, if node i and node j are in the same connected component, node i estimates the location of node j by $\hat{\Pi}_i^c(j, t)$. Note that node j has Q paths and that in this case only its main path is considered.

The more complex situation is when node i and node j are not in the same connected component. In this case, the information that node i has about node j's path is outdated. The node position estimate that node i forms is computed by:

$$X_i(j,t) = \frac{\sum_{q=1}^{Q} \hat{\Pi}_i(j,t,q) \cdot [\hat{\rho}_i(j,q)]^{1+\phi \cdot (t_c - S_P(i,j))^{-1}}}{\sum_{q=1}^{Q} [\hat{\rho}_i(j,q)]^{1+\phi \cdot (t_c - S_P(i,j))^{-1}}} \tag{2}$$

Where $X_i(j,t)$ is the node i's estimate of node j's position at time t. We introduce a new function $S_P(i,j)$ which is the "timestamp" of the information i has about j's paths. For those nodes in i's connected component, $S_P(i,j) = t_c$ where t_c is the current time. Equation 2 also defines a constant $\phi \geq 0$. A high ϕ means that node i assumes that node j will follow its best path for a long time. A low ϕ is good for high densities of nocomm zones and significant presence of uncertainty when node j is likely to change path preference with time.

2) Nocomm Zone Location: As described in Section III-E, $\lambda_i(u)$ and $\zeta_i(u)$ are the estimates that node i maintains about each location $u \in V_M$. There are two sources of information which force node i to adjust this estimate, both of which are described in Section IV-C.2.

C. Knowledge Fusion

1) Path Data: There are two sources of new path data information. The first is when nodes within a connected component change their plan after having moved. The second is when a new node becomes a member of a connected component that was not part of that component before. In both cases, the fusion of data is straight forward. All the nodes in a connected component set their path estimates to the most recent information available about that path.

2) Nocomm Zone Data: There are two sources of new nocomm zone information. The first is when a node senses new information about its surroundings after having moved. The second source is when a new node becomes a member of a connected component. In the former case, the node udates its estimate of its current location and the surrounding area. Specifically, the surrounding area is a disk of radius R_N. We denote the new information about the probability of a location $u \in V_M$ being a nocomm zone as $\lambda'_i(u)$ and the certainty of the new information as $\zeta'_i(u)$. $\lambda'_i(u) = 1$ if node i is connected to someone, and $\lambda'_i(u) = 0$ if it is isolated. If the node is connected at the current time, then the certainty of new information $\zeta'_i(u) = 1$. If it is not connected, $\zeta'_i(u)$ is computed by:

$$\begin{aligned}
\zeta'_i(u) &= U \cdot V \cdot W \\
U &= \max\left[0, 1 - \left(\frac{\|u - u_c(i)\|}{R_N}\right)^2\right] \\
V &= 1 - \left(\frac{\Delta_i(t_c - H_i)}{d_{\max}}\right)^2 \\
W &= \frac{1}{H_i}
\end{aligned} \tag{3}$$

Where $u \in V_M$ is a location on the movement graph, $u_c(i) \in V_M$ is the current position of the node, $\|u - u_c\|$ is the distance between u and u_c, H_i is the duration of time that node i has been disconnected, and $\Delta_i(t_c - H_i)$ is the distance to the closest neighbor at time $t_c - H_i$ which is the last time node i was connected.

Equation 3 has three distinct sources U, V, and W which add to a decrease of certainty. The first source U is the distance of the position in V_M from the current position of node i. This is a standard property of an inverse distance squared mask. The second source V decreases certainty if it appears that node i lost connectivity simply because it moved outside the range $d_{\max}$ of all it neighbors. The third source W decreases certainty as the age of the disconnection event increases.

Given the $\lambda'_i(u)$ and $\zeta'_i(u)$ estimates, the direct reactive update to $\lambda_i(u)$ and $\zeta_i(u)$ is done as follows:

$$\begin{aligned}
\lambda_i(u) =& \frac{\lambda_i(u)\zeta_i(u) + \lambda'_i(u)\zeta'_i(u)}{\zeta_i(u) + \zeta'_i(u)} \\
\zeta_i(u) =& \frac{1}{2}\left(\zeta'_i(u) - \frac{\zeta'_i(u) + \zeta_i(u)}{2}\right)^2 + \\
& \frac{1}{2}\left(\zeta_i(u) - \frac{\zeta'_i(u) + \zeta_i(u)}{2}\right)^2
\end{aligned} \tag{4}$$

The reinforcement learning update is done when an isolated node becomes connected again. The estimate update in this case requires that node i considers the part of the path that it has traveled while being isolated, in other words for times $\{t \in \mathbb{Z} : t_c - H_i \leq t < t_c\}$. For each position $\Pi^c(i,t)$ in this time period, the newly gained knowledge about the paths of nodes inside node i's connected component is used, to update the estimate. If $\Delta_i(t) < d_{\max}$, that means that at time t there was a node in node i's range and that it still was isolated. This information is integrated into the estimate of nocomm zone locations by using $\lambda'_i(\Pi^c(i,t)) = 1$ and $\zeta'_i(\Pi^c(i,t)) = \frac{1}{2}$ in an updated of $\lambda_i(\Pi^c(i,t))$ and $\zeta_i(\Pi^c(i,t))$ by using Equation 4.

The second source of nocomm zone information is when the estimates of nocomm zone locations change for at least one of the nodes in the connected component. This could be either due to the new information gained by movement of nodes inside a connected component, or due to a node entering a connected component that was not part of that component before. A single node i initiates this synchronization of the connected component. We denote all the nodes in node i's connected component as $\Lambda^\infty(i)$, which can be thought of as all the nodes reachable from i in an infinite number of packet hops. It is assumed that all nodes in this group have potentially differing estimates of nocomm zone locations, and the goal of the synchronization process is for all the nodes inside a connected component to have identical estimates. The update procedure for $\lambda_i(u \in V_M)$ is a mean weighted by the certainty, and for $\zeta_i(u \in V_M)$ is the variance weighted by the certainty:

$$\begin{aligned}
\lambda_{\Lambda^\infty(i)}(u) &= \frac{\sum_{j \in \Lambda^\infty(i)} \lambda_j(u)\zeta_j(u)}{\sum_{j \in \Lambda^\infty(i)} \zeta_j(u)} \\
\zeta_{\Lambda^\infty(i)}(u) &= \frac{\sum_{j \in \Lambda^\infty(i)} \zeta_j(u)\left[\lambda_j(u) - \lambda_{\Lambda^\infty(i)}(u)\right]^2}{0.25\sum_{j \in \Lambda^\infty(i)} \zeta_j(u)}
\end{aligned} \tag{5}$$

Where $\lambda_{\Lambda^\infty(i)}(u)$ is the new $\lambda_j(u)$ estimate for all $j \in \Lambda^\infty(i)$, and $\zeta_{\Lambda^\infty(i)}(u)$ is the new $\zeta_j(u)$ estimate for all $j \in \Lambda^\infty(i)$. Note the constant 0.25 in the denominator. It is a normalization constant which expands the range of $\zeta_{\Lambda^\infty(i)}(u)$ to $[0,1]$. This update is performed for all $u \in V_M$ so that the final result is a consensus on the estimate of nocomm zone locations inside node i's connected component.

D. Heuristic Functions

The cost of each state $u \in V_M$ when considered by node i at time t is determined by a weighted sum of three objectives:

$$\mathbf{H}_i(u,t) = w_1 H_i^1(u) + w_2 H_i^2(u,t) + w_3 H_i^3(u,t) \tag{6}$$

Where w_1, w_2, and w_3 are weights, and H^1, H^2, and H^3 are heuristic functions which are described in more detail in this section. The cost of state u at time t for node i is denoted as $\mathbf{H}_i(u,t)$. Equation 6 applies when the length of the shortest path to i's goal from position u is less than $T_{\max} - t$. Otherwise, node i marches along the shortest path. This ensures that all nodes are at their destination at or before time $T_{\max}$.

1) Distance to Goal $H_i^1(u)$: This is the main objective of the path planning problem: to reach a prespecified destination. The measure is the distance from u to the goal position of node i. H_2 is normalized by the distance from the origin of node i to the destination of node i. The higher this value is the higher the cost of the state. What this objective expresses is the need to always keep moving toward the destination unless network performance objectives strongly suggest detours.

2) Distance to Nearest Neighbor $H_i^2(u,t)$: This measure is based both on outdated and up-to-date information as described in Section IV-B.1. The measure is the distance to the closest node based on estimated position whether the node is inside or outside the connected component. H_1 is normalized by $d_{\max}$. The higher this value is the higher the cost of the state. What this objective expresses is the need for nodes to stay close together in order to satisfy the distance requirement for connectivity.

3) Location of Nocomm Zones $H_i^3(u,t)$: This measure uses the estimates of nocomm zones computed by each node. For each state $u \in V_M$, there are two values which describe the estimate $\lambda_i(u)$ and $\zeta_i(u)$. These two values are combined by $H_i^3(u,t) = \zeta_i(u) \cdot (0.5 - \lambda_i(u)) + 0.5$. The range of $H_i^3(u,t)$ is $[0,1]$. Value of 0 suggests that this location is, with high certainty, not part of a nocomm zone; value of 0.5 suggest a lack of certainty about the nocomm zone membership of u; value of 1 suggest that this location is in fact part of a nocomm zone. What this objective expressed is the need to avoid areas that have some high probability of being a nocomm zone.

4) Utility Function: For ant colony optimization it is more convenient to define a utility function $\mathbf{U}_i(u,t)$ based on the cost function $\mathbf{H}_i(u,t)$. Since the ranges of the three individual heuristic functions are known, the maximum value $\mathbf{H}_{\max}$ that $\mathbf{H}_i(u,t)$ can take on can be determined by:

$$\mathbf{H}_{\max} = w_1 H_{\max}^1 + w_2 H_{\max}^2 + w_3 H_{\max}^3 \tag{7}$$

$H_{\max}^1$ is the maximum distance between any two vertices in V_M. $H_{\max}^2$ is $H_{\max}^1$ divided by $d_{\max}$. $H_{\max}^3$ is equal to 1. Using this information the utility function can be defined as:

$$\mathbf{U}_i(u,t) = 1 - \frac{\mathbf{H}_i(u,t)}{\mathbf{H}_{\max}} \tag{8}$$

The nice property of this utility function is that its range is [0,1].

V. Optimization Methodology

At each time step, each node $i \in V_F$ has to compute a set of paths based on its estimate of the network state and the environment. The secondary optimization problem is to compute the weights that are used to form a single heuristic function from a multiobjective one. The former problem is solved using an efficient and flexible method of ant colony optimization. The latter problem is solved using simulated annealing based on insight on the relationship between the density of nocomm zones and weights used for scalarization.

A. Ant Colony Path Planning

Ant Colony Optimization (ACO) [22] is a general nature-inspired method of searching a weighted graph. It has several unique characteristics which make it effective to the problem of time-critical path planning under uncertainty. First, it is an algorithm which provides a solution at each iteration. This is important for time-critical planning as the amount of time that can be dedicated to replanning is severely limited in the medium of highly mobile networks. Second property is that it provides a set of solutions, instead of just one solution. This is important for the case of planning under uncertainty. When node i has lost contact with node j, it is important for node i to know of several paths that node j was considering. This improves node i's ability to predict the range of positions that node j is likely to occupy in the future. The third property is that every complete execution of the algorithm constructs estimates of state values which can be used in the future to adjust plans. Therefore, every execution of the algorithm does not start from scratch. If node i is continuously replanning, by using this table as a starting point, it is able to adjust promptly to the changes in the dynamic world brought about by new alterations of the environment or the acquisition of new information.

In order to guide the path search for an ant we use a pheromone table $\tau : E_M \to \mathbb{R}$. It is important to note that in most cases the self-reference edges $(u,u) \in E_M, \forall u \in V_M$. Initially, before the start of the scenario, $\tau(u,v)$ is 1 for all $(u,v) \in E_F$. Each node $i \in V_F$ computes a set of Q plans before performing each movement step. Each node in a connected component plans noncooperatively, but informs other nodes in that component of their plan. This procedure is repeated until the plans created by each node remain unchanged or a fixed time T_{plan} runs out. In this way, the nodes inside the same connected component cooperate indirectly.

To find a single plan from node i's current position to its destination, node i sends out a virtual ant on the movement graph G_M. At each position u on that graph, the ant chooses its next position $v \in \Lambda(u)$ with probability:

$$p_{uv} = \frac{[\tau_i(u,v)]^{\alpha} [\mathbf{U}_i(v, t_a+1)]^{-\beta}}{\sum_{w \in \Lambda(u)} [\tau_i(u,w)]^{\alpha} [\mathbf{U}_i(w, t_a+1)]^{-\beta}} \tag{9}$$

Where $t_a \geq t_c$ is the current virtual time in the ants simulated walk on the movement graph, and α and β determine the relative importance of pheromone intensity and estimated state utility. Note that β is negative because the range of the utility function is $[0,1]$ while the pheromone values almost always are greater than 1. Therefore, increasing α or β increase the relative important of pheromones and utility, respectively.

Once the virtual ant reaches i's destination, it backtracks along the path that it traveled and updates the pheromone table based on the average number of connected component along that path. Specifically, given that each nodes sends out Z ants at each time step, the pheromone update value $\tau_i^z(u,v)$ for the zth ant sent out by node i, for all (u,v) in that ants path, is:

$$\tau_i^z(u,v) = 1 - \frac{1}{n} \sum_{t=t_c}^{T_{\max}} N(\Pi_{\text{ant}}(i,z,t)) \tag{10}$$

Where $\mathbf{\Pi}_{\text{ant}}(i,z)$ is a plan computed by one of the Z ants which are sent out by node i at each time step.

Once the new pheromone information has been computed in a single time step the pheromone table for node i is updated by:

$$\tau_i(u,v) = (1-\delta) \cdot \tau_i(u,v) + \sum_{z=1}^{Z} \tau_i^z(u,v) \tag{11}$$

Where δ is the rate of pheromone evaporation.

For a time step, the result of the ant path search process for node i is Z paths. Node i selects Q best paths out of that set as its currently best available paths. Since the set of Z virtual paths may have repeats, it is possible for node i to have less than Q paths. The updated pheromone table τ_i is used in the next iteration for better guidance of the ants. This is a crucial repository of knowledge that the node stores and updates throughout the scenario.

B. Simulated Annealing in Weight Space

Each node uses the cost function in Equation 6 to help determine which paths are more likely to lead to a low number of connected components. The total cost of a state is computed by taking a weighted sum of three heuristic measurements. Picking these weights arbitrarily can lead to poor network performance, as shown in Section VI. Therefore, optimization in the $\mathbb{R}^3$ weight space is required.

Simulated annealing [23] is used to converge to a set of weights that is optimal for a given density of nocomm zones. From extensive experimentation, this process showed a correlation between the nocomm density and the weights. For mid densities $[0.2, 0.7]$, w_3 becomes very important relative to the other two weights. This means that when nocomm zones are present in manageable amounts, their avoidance is the best way to improve the connectivity of the network. For low densities $[0, 0.2]$, w_2 is high relative to the other weights. This mean that when most of the arena is good for communication, not wondering out of range is the most effective strategy to maintain good network performance. For high nocomm densities $[0.7, 1.0]$, all the weights are about the same with w_1 slightly higher than the others, which means that the naive march ahead policy is most effective for these densities.

The simulated annealing process is a one-time offline initialization process for the weights. The result of is a function which maps the density of nocomm zones to a set of value for the three weights. The network nodes, however, do not know the density of nocomm zones, but as is shown in Section VI can estimate it efficiently and with sufficient accuracy to make appropriate assignments to the weights. Each node has its own estimate of nocomm zone locations which is constructed online as shown in Section IV-C.2. This estimate is what is used to compute an estimate for the density of nocomm zones. Only the $\lambda_i(u)$ values which has a certainty of $\zeta_i(u) \geq \zeta_{\min}$ are used in the estimate. $\zeta_{\min}$ is a threshold of certainty which suggests that the estimate can be trusted.

VI. Simulation and Results

The problem of path planning for connectivity under uncertainty requires that network nodes are able to both efficiently acquire knowledge about the environment and take advantage of that information to form movement plans which show network performance improvement over the naive movement policy.

A. Simulation Setup

Simulations were performed on an arena of 100×150 m^2. It was discretized to form a lattice movement graph G_M of $15,000$ thousand nodes, with each non-border node having 8 neighbors and a degree of 9 which includes the self-reference edge. Each simulation involved 10 nodes. The origins of the nodes were evenly spaced along the bottom of the arena and the destinations of the nodes were evenly spaced along the top of the arena. So the origin of node 0, for example, is $(7.5, 0)$, and the origin of node 1 is $(22.5, 0)$. So for each node i, the distance at start and end of a scenario to its closest neighbor is 15. $d_{\max}$ was chosen to be 20. Each data point on the presented plots is an average of $10,000$ simulations.

The shortest scenario duration $T_{\min} = 100$, since the lengths of the shortest paths given the topology used in these simulations are all equal to 100. $T_{\max}$ was set to 105, which means that there is a slack time of 5 steps that is allowed for planning. The greater $T_{\max}$ is the more network performance improvement is attainable [19], [20]. Only a small slack was provided to show that improvement can be attained with almost no cost to the travel time.

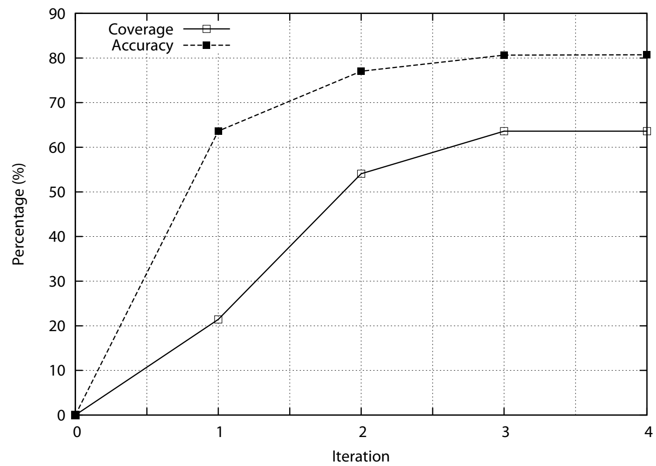

Fig. 2. The coverage increase of knowledge certainty about nocomm zone locations and the accuracy of that knowledge with repeating traversals of an environment.

ϕ from Equation 2 was set to 1. For ant colony optimization, the rate of evaporation δ was set to 0.1, and both α and β from Equation 9 were set to 1. Unless otherwise stated, the weights for the heuristic function in Equation 7 are all set to 1.

B. Learning Rate and Accuracy

Exploration as an objective is not directly part of the optimization process in that nodes only acquire enough information to help local maneuvering around nocomm zones and within the communication range of other nodes. Over time, however, the knowledge acquisition and fusion process establishes at each node a detailed estimate of the environment. If the nodes are allowed to travel along that same environment again, the breadth of their knowledge expands. Figure 2 shows the percentage of positions in V_M that on average the nodes have more than 0.8 certainty about in terms of it being or not being part of a nocomm zone. Also, the figure shows, for the estimates that have more than 0.8 certainty, the probability of it being a correct estimate. After only 3 runs, it appears that the estimates converge to about 80% accuracy. It is important to note the difference between obstacles in the physical medium and nocomm zones which are obstacles in the communication medium. It is more difficult to detect nocomm zones, because when a node's communication goes out, a node cannot easily distinguish which of the three scenarios caused this. From the perspective of a node that just got disconnected, the scenarios again are: (1) I'm in a nocomm zone, (2) My neighbors are in a nocomm zone, (3) My neighbors went out of my range, or some combination of (2) and (3). Only when other nodes confirm one of the three hypotheses, can a node be more sure about the communication status of that position.

C. Connectivity Improvement

The main goal of this paper is develop a methodology for constructing a movement policy that can provide good network performance under uncertain knowledge and gradually-learned estimate of the environment and network state. Figure 3 shows the average network connectivity attained with and without planning with a communication objective. The base of comparison is a naive strategy which marches ahead along the shortest path to the

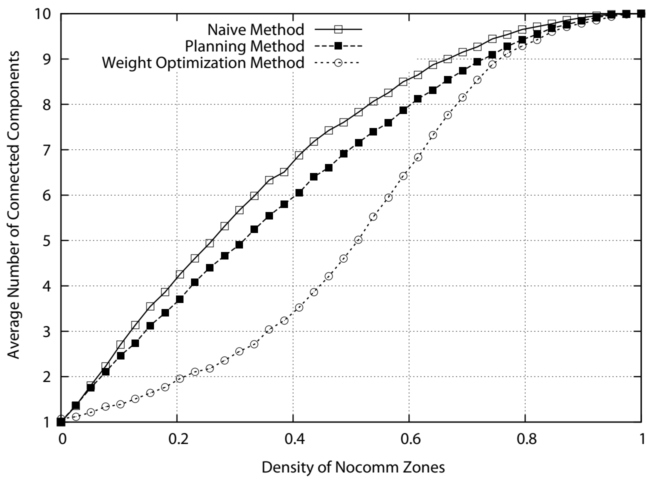

Fig. 3. Comparison of the improvement of network connectivity attained by three different methods with increasing density of nocomm zones.

destination. The two other strategies shown in the figure are the network-centric planning with heuristic weights of $(1, 1, 1)$ and with optimized weights as described in Section V-B. This plot was generated by running the nodes from their origins to their goals and then back to their origins, recording the average connectivity for the trip back. The first part of this scenario is meant to demonstrate the ability of the network nodes to estimate the communication signal properties of the environment, and the second part of this scenario is meant to demonstrate the ability of the nodes to take advantage of the newly gained information in improving connectivity.

The key insight provided by Figure 3 is that path planning can achieve significant improvement in connectivity relative to the naive march-ahead approach for low to medium densities of nocomm zones. Another insight is that the offline optimization of weights, along with online estimation of nocomm zone density, can greatly improve the benefit gained from planning with communication objectives.

VII. Conclusion

We show the benefit of path planning in communication networks under incomplete and uncertain knowledge by proposing a distributed method which both acquires knowledge through exploration, fuses estimation data from other nodes, and uses this information to form movement plans of good average network performance. The latter is computed by ant colony optimization, with the scalarization weights of the heuristic function optimized by simulated annealing.

Acknowledgments

This work is funded by the U.S. Army Communications–Electronics Research, Development and Engineering Center (CERDEC), under contract #DAAB-07-01-9-L504.

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] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 2864–2869.

[3] J. P. Desai, P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 905–908, December 2001.

[4] J. T. Feddema, C. Lewis, and D. A. Schoenwald, "Decentralized control of cooperative robotic vehicles: Theory and application," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 852–864, October 2002.

[5] H. Yamaguchi, "Adaptive formation control for distributed autonomous mobile robot groups," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997, pp. 2300–2305.

[6] H. Yamaguchi and J. W. Burdick, "Asymptotic stabilization of multiple nonholonomic mobile robots forming group formations," in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 3573–3580.

[7] 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.

[8] 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.

[9] 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.

[10] E. Bicho and S. Monteiro, "Formation control for multiple mobile robots: a non-linear attractor dynamics approach," in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, October 2003, pp. 2016–2022.

[11] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[12] S. Engelson and D. McDermott, "Error correction in mobile robot map learning," *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pp. 2555–2560, 1992.

[13] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications tomulti-robot and 3d mapping," *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, 2000.

[14] G. Dedeoglu and G. S. Sukhatme, "Landmark-based matching algorithm for cooperative mapping by autonomous robots," in *Distributed Autonomous Robotic Systems 4, Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems, DARS 2000*, L. E. Parker, G. A. Bekey, and J. Barhen, Eds. Springer, 2000, pp. 251–260.

[15] J. A. Sauter, R. S. Matthews, H. V. D. Parunak, and S. Brueckner, "Evolving adaptive pheromone path planning mechanisms," in *AAMAS*. ACM, 2002, pp. 434–440. [Online]. Available: http://doi.acm.org/10.1145/544741.544841

[16] M. Plucinski, "Application of the ant colony algorithm for the path planning," pp. 345–352, 2005.

[17] X. Fan, X. Luo, S. Yi, S. Yang, and H. Zhang, "Optimal path planning for mobile robots based on intensified ant colony optimization algorithm," *Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on*, vol. 1, 2003.

[18] H. Mei, Y. Tian, and L. Zu, "A hybrid ant colony optimization algorithm for path planning of robot in dynamic environment," 2005.

[19] A. 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.

[20] A. Fridman, S. Weber, C. Graff, and M. Kam, "Path planning for network performance," in *Proceedings of the 50th Annual IEEE Globecom Conference*, Washington, DC, November 2007.

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

[22] M. Dorigo and T. Stutzle, "The ant colony optimization metaheuristic: Algorithms, applications and advances," *Handbook of Metaheuristics*, vol. 57, pp. 251–285, 2002.

[23] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science, Number 4598, 13 May 1983*, vol. 220, 4598, pp. 671–680, 1983. [Online]. Available: citeseer.ist.psu.edu/kirkpatrick83optimization.html