

# COMMUNICATION MODELS FOR A COOPERATIVE NETWORK OF AUTONOMOUS AGENTS

ASHWIN ARULSELVAN, CLAYTON W. COMMANDER, MICHAEL J. HIRSCH,  
AND PANOS M. PARDALOS

ABSTRACT. We consider the problem of maximizing the total connectivity for a set of wireless agents in a mobile ad hoc network. That is, given a set of wireless units each having a start point and a destination point, our goal is to determine a set of routes for the units which maximizes the overall connection time between them. Known as the COOPERATIVE COMMUNICATION PROBLEM IN MOBILE AD HOC NETWORKS (CCPM), this problem has several military applications including coordination of rescue groups, path planning for unmanned air vehicles, and geographical exploration and target recognition. The CCPM is  $\mathcal{NP}$ -hard; therefore heuristic development has been the major focus of research. In this work, we survey the CCPM examining first some early combinatorial formulations and solution techniques. Then we introduce new continuous formulations and compare the results of several case studies. By removing the underlying graph structure, we are able to create a more realistic model of the problem as supported by the numerical evidence.

## 1. INTRODUCTION

Research in the area of cooperative networks has surged in recent years [5, 6, 17, 22]. This particular branch of telecommunications is leading the way for future technologies and the development of new network organizations [25]. In particular, so-called *mobile ad hoc networks* (MANETs) are at the forefront of the work in autonomous cooperative networks [10]. MANETs are comprised of a set of loosely coupled agents which communicate via a shared radio channel with other agents within a specified range. The unique feature of MANETs that separates them from traditional cellular networks is the fact that the topology of MANETs is dynamic. That is, with each movement of the agents, a new topology is established.

The lack of a pre-established infrastructure is an attractive feature of MANETs. They are particularly useful in situations where communication is required, but no fixed telecommunication system exists. MANETs are also helpful when a set of mobile users need to be in constant contact with each other. Specific examples include combat search and rescue teams, and medical teams. In the wake of disasters such as the terrorist attacks of September 11, 2001, and Hurricane Katrina, the nation saw first hand that communication among the emergency responders was critical to the success of the rescue operations.

Most cooperative networks require coordination among the group of users in order to accomplish the objective. The coordination of the system usually depends on communication being guaranteed amongst the agents. In typical ad hoc networks, bandwidth and communication time are very limited resources. Therefore, we see that the lack of a central command center for MANETs, while appealing from a distributed perspective, does lead to several problems in terms of routing, communication, and path-planning [4, 8]. Perhaps the most important among these, and the focus of this chapter, is the study of communication models in the network. In particular, we study the problem of coordinating a set of wireless agents involved in a task that requires them to travel from a source location to a destination. The objective is to determine the paths, or trajectories for the agents which maximizes the connectivity between them subject to constraints on the initial and final

configurations, and several limitations on the movements of the agents [23]. This problem is known as the COOPERATIVE COMMUNICATION PROBLEM IN MOBILE AD HOC NETWORKS (CCPM), and is known to be  $\mathcal{NP}$ -hard [16]. In the next section, we review the currently existing work on the CCPM, which is primarily focused on heuristics for the problem posed as a discrete optimization problem.

The organization of this chapter is as follows. In Section 2, we provide the problem formulation for the CCPM. Then in Section 3 we present a review of the previous work in the area of communication in cooperative networks. In Section 4, we derive two mixed integer formulations for the CCPM using the combinatorial problem as a guide. We provide some preliminary numerical results in Section 5 and discuss conclusions and directions of future research in Section 6.

## 2. PROBLEM FORMULATION

Consider an undirected graph  $G = (V, E)$ , where  $V = \{v_1, v_2, \dots, v_n\}$  represents the set of available positions for the wireless agents. Each node in  $V$  is assumed to be connected only to nodes that can be reached in one time step. Let  $U$  represent the set of agents,  $S = \{s_1, s_2, \dots, s_{|U|}\} \subseteq V$  the set of initial positions, and  $D = \{d_1, d_2, \dots, d_{|U|}\} \subseteq V$  the set of destination positions for the agents. Furthermore, let  $N(v) \subseteq 2^V$ , for  $v \in V$ , represent the set of neighbors, or nodes, which are adjacent to node  $v$ . Given a time horizon  $T$ , the objective of the problem is to determine a set of routes for the agents, such that each agent  $u_i \in U$  starts at source node  $s_i$  and finishes at its respective destination node  $d_i$  after at most  $T$  units of time [10].

For each agent  $u \in U$ , the function  $p_t : U \rightarrow V$  returns the position of the agent at time  $t \in \{1, 2, \dots, T\}$ . Then at each time instant  $t$ , an agent  $u \in U$  can either remain in its current location, i.e.  $p_{t-1}(u)$ , or move to a node in  $N(p_{t-1}(u))$ .

We can represent a route for an agent  $u \in U$  as a path  $\mathcal{P} = \{v_1, v_2, \dots, v_k\} \subseteq V$  where  $v_1 = s_u$ ,  $v_k = d_u$ , and, for  $i \in \{2, \dots, k\}$ ,  $v_i \in N(v_{i-1}) \cup \{v_i\}$ . Finally, if  $\{\mathcal{P}_i\}_{i=1}^{|U|}$  is the set of trajectories for the agents, we are given a corresponding vector  $\mathcal{L}$  such that  $\mathcal{L}_i$  is a threshold on the size of path  $\mathcal{P}_i$ . This value is typically determined by fuel or battery life constraints on the wireless agents.

We assume that the agents have omnidirectional antennas and that two agents in the network are connected if the distance between them is less than some radius  $r \in \mathbb{R}$ . The particular value of  $r$  is determined by the capabilities of the wireless equipment, such as the antenna strength and power amplifier. More specifically, let  $\delta : V \times V \rightarrow \mathbb{R}$  represent the Euclidean distance between a pair of nodes in the graph. Then, we can define a function  $c : V \times V \rightarrow \{0, 1\}$  such that

$$c(p_t(u_i), p_t(u_j)) = \begin{cases} 1, & \text{if } \delta(p_t(u_i), p_t(u_j)) \leq r \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

With this, we can define the CCPM as the following optimization problem as given by Commander et al. [10]:

$$\max \sum_{t=1}^T \sum_{u, v \in U} c(p_t(u), p_t(v)) \quad (2)$$

$$\text{s.t.} \quad \sum_{j=2}^{n_i} \delta(v_{j-1}, v_j) \leq \mathcal{L}_i, \quad \forall \mathcal{P}_i = \{v_1, v_2, \dots, v_{n_i}\} \quad (3)$$

$$p_1(u) = s_u \quad \forall u \in U \quad (4)$$

$$p_T(u) = d_u \quad \forall u \in U, \quad (5)$$

where constraint (3) ensures that the length of each path  $\mathcal{P}_i$  is less than or equal to its maximum allowed length  $\mathcal{L}_i$ .

It has been determined [23] that the problem described above is  $\mathcal{NP}$ -hard. This can be shown by a reduction from the well known 3SAT problem. Moreover, it is  $\mathcal{NP}$ -hard even to find an optimal solution for one stage of the problem at a given time  $t$ . To see this, consider an algorithm that maximizes the number of connections at time  $t$ , by defining the positions for members of the network. Run this algorithm for different sets  $U^i$ , with  $|U^i| = i$  and  $i$  varying from 1 to  $T$ . Then the algorithm stops when the number of connections is less than  $\binom{i}{2}$ , and the value returned is  $i - 1$ ; clearly, the algorithm described above computes the value of the maximum clique on the underlying unit graph [9]. However, computing optimal solutions for the MAXIMUM CLIQUE PROBLEM on a unit graph is known to be  $\mathcal{NP}$ -hard [7].

Due to the computational complexity of the problem, real-world instances cannot be solved exactly. Therefore, we turn our attention to the design and implementation of efficient heuristics to solve large-scale instances within reasonable computing times. In the following section, we review the recent work in this area and describe the implementation of the first advanced metaheuristic for the CCPM based on the Greedy Randomized Adaptive Search Procedure [26].

### 3. PREVIOUS WORK

Since the introduction of the CCPM by Oliveira and Pardalos [23], heuristic design has been a major focus [9, 10]. In [10], the authors introduced a construction heuristic for the CCPM based on shortest paths [1]. The goal was to provide a way to quickly calculate sets of feasible trajectories for the agents. Pseudo-code for this algorithm is provided in Figure 1. The procedure takes as input an instance of the CCPM consisting of the graph  $G$ , the set of agents  $U$ , source nodes  $S$ , destination nodes  $D$ , and a maximum travel time  $T$ . The total number of connections ( $c$ ) represents the value of the objective function from equation (2) and is initialized to zero. The set of trajectories for the agents ( $\text{solution}$ ) is initialized to the empty set. In line 3, we compute the shortest source-destination path for each agent using the Floyd-Warshal algorithm [15, 27]. For each agent  $i \in U$ , the corresponding shortest path is assigned as the trajectory  $\mathcal{P}_i$  for the agent. The trajectory is feasible if agent  $i$  is able to reach its destination  $d_i$  in at most  $T$  time units. Any agent which reaches its target location in less than  $T$  time steps will remain there until all other agents reach their respective destinations. If any path is infeasible, the algorithm terminates. Otherwise, the number of connections is updated and the process repeats until all agents have been considered.

```

procedure ShortestPath( $G, U, S, D, T$ )
1   $c \leftarrow 0$ 
2   $\text{solution} \leftarrow \emptyset$ 
3  Compute all shortest paths  $\text{SP}(s_i, d_i)$  for each pair  $(s_i, d_i) \in S \times D$ 
4  for  $i = 1$  to  $|U|$  do
5     $\mathcal{P}_i \leftarrow \text{SP}(s_i, d_i)$ 
6    if length of  $\mathcal{P}_i > T$  then
7      return  $\emptyset$ 
8    else
9       $\text{solution} \leftarrow \text{solution} \cup \mathcal{P}_i$ 
10      $c \leftarrow c + \text{new connections generated by } \mathcal{P}_i$ 
11    end
12 end
13 return ( $c, \text{solution}$ )
end procedure ShortestPath

```

FIGURE 1. Pseudo-code for the shortest-path construction heuristic.

The aforementioned algorithm provides feasible solutions for instances of the CCPM in  $\mathcal{O}(|V|^3)$  time. However, the trajectories calculated are not guaranteed to be locally optimal, let alone globally optimal. Therefore, a local neighborhood search enhancement was applied. In general, a local search method receives a feasible solution as input and returns a solution that is guaranteed to be optimal with respect to the given neighborhood structure. Let  $\mathcal{S}$  be the set of feasible solutions for an instance  $\Pi$  of the CCPM. Then for some  $s \in \mathcal{S}$  the neighborhood of  $s$ , denoted  $\mathcal{N}(s)$ , can be defined as the set of all solutions  $s' \in \mathcal{S}$  that differ from  $s$  in exactly one route. Notice that the number of feasible paths between any source-destination pair is exponential, and could lead to unreasonable computation times. Therefore instead of exhaustively searching the entire neighborhood the authors probe only  $|U|$  neighbors at each iteration (one for each source-destination pair). Also, because of the exponential size of the neighborhood, the maximum number of iterations performed was limited to a constant `MaxIter`.

Pseudo-code for the local improvement heuristic can be seen in Figure 2. Let  $f$  represent the objective function for the CCPM as given in equation (2) above. New routes are computed using a randomized version of the standard depth-first search (DFS) [1]. As mentioned in [10], at each step of the randomized DFS, the node selected to explore is uniformly chosen among the available children of the current node. Randomization helps to find a route that may improve the solution, while avoiding being trapped at a local optimum after only a few iterations. The local search is a standard hill-climbing method. Beginning with the feasible solution from the shortest path constructor, the local search begins computing new trajectories for the agents by using the randomized DFS to explore the neighborhood as described above. The method iterates over all the agents and repeats a total of `MaxIter` iterations after which the current best solution is deemed locally optimal and returned.

```

procedure HillClimb(solution)
1   $c \leftarrow f(\text{solution})$ 
2  while solution not locally optimal and iter < MaxIter do
3    for  $i = 1$  to  $|U|$  do
4      solution  $\leftarrow$  solution  $\setminus$   $\{\mathcal{P}_i\}$ 
5       $\mathcal{P}'_i \leftarrow \text{DFS}(s_i, d_i)$ 
6       $c' \leftarrow f(\text{solution} \cup \mathcal{P}'_i)$ 
7      if length of  $\mathcal{P}'_i < T$  and  $c' > c$  then
8         $c \leftarrow c'$ 
9        iter  $\leftarrow$  0
10     else
11       Restore path  $\mathcal{P}_i$ 
12     end
13   end for
14   iter  $\leftarrow$  iter+1
15 end while
16 return (solution)
end procedure HillClimb

```

FIGURE 2. Pseudo-code for the Hill Climbing intensification procedure.

The two aforementioned heuristics can be combined into a one-pass heuristic for the CCPM as shown in Figure 3. Using this method, locally optimal solutions can be efficiently computed in  $\mathcal{O}(\max\{n^3, kTu^2m\})$  time, where  $T$  is the time horizon,  $u = |U|$ ,  $n = |V|$ ,  $m = |E|$ , and  $k = \text{MaxIter}$  is the maximum number of iterations allowed in the local search phase.

We now describe the implementation of a more advanced randomized multi-start heuristic for the CCPM based on the Greedy Randomized Adaptive Search Procedure (GRASP)

```

procedure OnePass( $G, U, S, D, T$ )
1  solution  $\leftarrow$  ShortestPath( $G, U, S, D, T$ )
2  solution  $\leftarrow$  HillClimb(solution)
3  return (solution)
end procedure OnePass

```

FIGURE 3. Pseudo-code for the one-pass heuristic.

[26] framework. GRASP is a two-phase metaheuristic for combinatorial optimization that aims to find very good solutions through the controlled use of random sampling, greedy selection, and local search. GRASP has been used extensively in the last decade on numerous optimization problems and produces excellent results in practice [14]. Let  $F$  be the set of feasible solutions for the a problem  $\Pi$ , where each solution  $S \in F$  is composed of  $k$  discrete components  $a_1, \dots, a_k$ . GRASP constructs a sequence  $\{S_i\}_i$  of solutions for  $\Pi$ , such that each  $S_i$  is feasible for  $\Pi$ . At the end, the algorithm returns the best solution found.

```

procedure GRASP(MaxIter)
1   $X^* \leftarrow \emptyset$ 
2  for  $i = 1$  to MaxIter do
3     $X \leftarrow$  ConstructionSolution( $G, g, X$ )
4     $X \leftarrow$  LocalSearch( $X, \text{MaxIterLS}$ )
5    if  $f(X) \geq f(X^*)$  then
6       $X^* \leftarrow X$ 
7    end
8  end
9  return  $X^*$ 
end procedure GRASP

```

FIGURE 4. GRASP for maximization.

Pseudo-code for the GRASP is provided in Figure 4. Notice that each GRASP solution is built in two stages, called *greedy randomized construction* and *intensification* phases. The construction phase receives as parameter an instance of the problem, a ranking function  $g : A(S) \rightarrow R$  (where  $A(S)$  is the domain of feasible components  $a_1, \dots, a_k$  for a partial solution  $S$ ), and a constant  $0 < \alpha < 1$ . It starts with an empty partial solution  $S$ . Assuming that  $|A(S)| = l$ , the algorithm creates a list of the best ranked  $\alpha l$  components in  $A(S)$ , and returns a uniformly chosen element  $x$  from this list. The current partial solution is augmented to include  $x$ , and the procedure is repeated until the solution is feasible, i.e. until  $S \in F$ .

The intensification phase consists of the implementation of a hill-climbing procedure. Given a solution  $S \in F$ , let  $N(S)$  be the set of solutions that can be found from  $S$  by changing one of the components  $a \in S$ . Then,  $N(S)$  is called a neighborhood of  $S$ . The improvement algorithm consists of finding, at each step, the element  $S^*$  such that

$$S^* = \operatorname{argmax}_{S' \in N(S)} f(S'),$$

where  $f : F \rightarrow R$  is the objective function of the problem. At the end of each step, we assign  $S \leftarrow S^*$  if  $f(S) > f(S^*)$ . The algorithm will converge to a local optimum, in which case the procedure above will generate a solution  $S^*$  such that  $f(S^*) \geq f(S)$  for each  $S \in N(S^*)$ .

To apply GRASP to the CCPM, we need to specify the set  $A$ , the greedy function  $g$ , the parameter  $\alpha$ , and the neighborhood  $N(S)$ , for  $S \in F$ . The components of each solution

$S$  are feasible moves of a member of the ad hoc network from a node  $v$  to a node  $w \in N(v) \cup \{v\}$ . The complete solution is constructed according to the following procedure. Start with a random  $u \in U$  and find the shortest path  $P$  from  $s_u$  to  $d_u$ . If the total distance of  $P$  is greater than  $D_u$ , then the instance is clearly infeasible, and the algorithm ends. Otherwise, the algorithm considers each feasible move. A feasible move connects the final node of a sub-path  $P_v$ , for  $v \in U \setminus \{u\}$ , to another node  $w$ , such that the shortest path from  $w$  to  $d_v$  has distance at most  $D_v - \sum_{e \in P_v} \text{dist}(e)$ . The set of all feasible moves in a solution is defined as  $A(S)$ .

The greedy function  $g$  returns for each move in  $A(S)$  the number of additional connections created by that move. As described above, the construction procedure will rank the elements of  $A(S)$  according to  $g$ , and return one of the best  $\alpha \cdot |A(S)|$  elements. This is repeated until a complete solution for the problem is obtained.

The improvement phase is defined by the perturbation function, which consists of selecting a wireless agent  $u \in U$  and rerouting it, i.e. finding a complete path using the procedure described above for each time step 1 to  $T$ . The set of all perturbations of a solution  $S$  is its neighborhood  $N(S)$ . At each step, all elements  $u \in U$  are tested, and the procedure stops when no such element  $u$  that improves the current solution can be found [9].

#### 4. CONTINUOUS FORMULATIONS

In this section, we present continuous formulations of the CCPM [2]. These formulations will provide a more realistic scenario than the discrete formulation provided above, in that movement is not restricted to a discrete set of positions. We will assume that the agents are operating in a battlespace  $\mathcal{Q} \subseteq \mathbb{R}^d$ , where  $\mathcal{Q}$  is a compact, convex set with unit volume and the Euclidean norm  $\|\cdot\|_2$  in  $\mathbb{R}^d$ . For our purposes, we are going to consider the planar case, i.e.  $d = 2$ , with the understanding that extensions to higher dimensions are easily achieved. Suppose there are  $M$  wireless agents in the ad hoc network. The  $M$  agents are assumed to be omnidirectional and are modeled as point masses. We allow the agents to move freely within  $\mathcal{Q}$  at some bounded velocity.

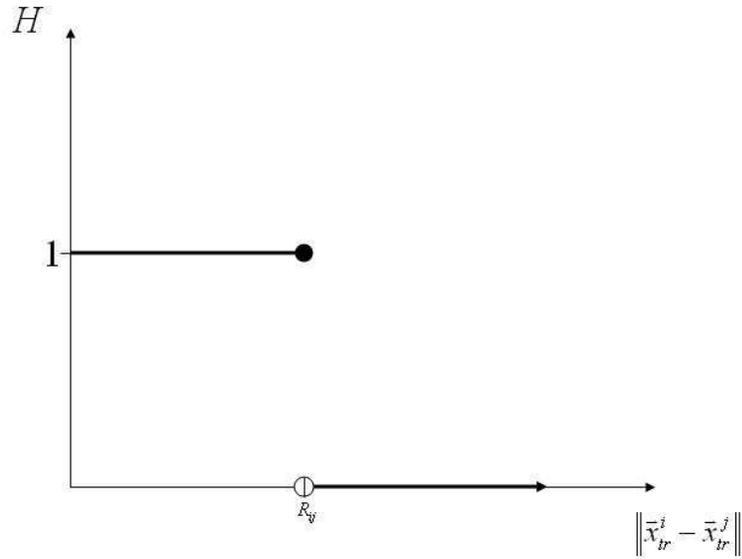
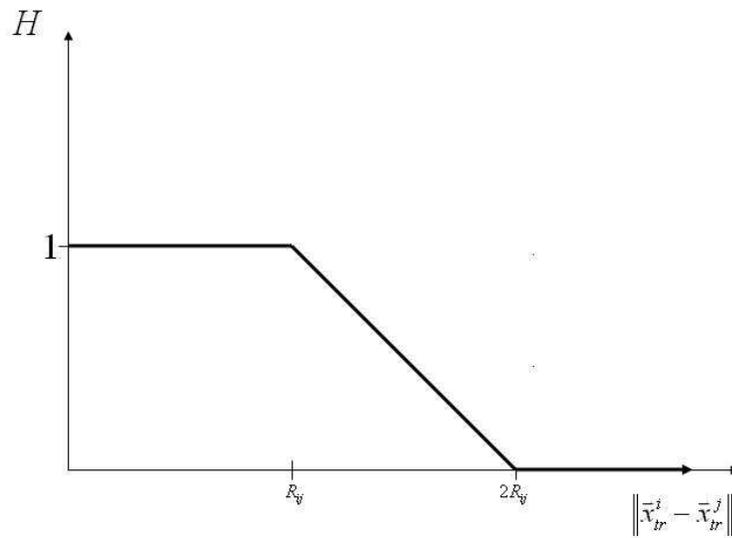
**4.1. Formulation 1: A Continuous Analog of CCPM-D.** In order to derive a continuous formulation, we need to define an objective function that is consistent with that of the discrete formulation. Let  $R_{ij}$  be the communication constant for agents  $i$  and  $j$ . That is,  $R_{ij}$  is the radius of communication for the two agents. One possible objective is to maximize the *Heaviside function*, defined as

$$H_1[R_{ij} - \|\vec{x}(t)^i - \vec{x}(t)^j\|_2] = \begin{cases} 1, & \text{if } \|\vec{x}(t)^i - \vec{x}(t)^j\|_2 \leq R_{ij} \\ 0, & \text{if } \|\vec{x}(t)^i - \vec{x}(t)^j\|_2 > R_{ij}. \end{cases} \quad (6)$$

A graphical representation of  $H_1$  is displayed in Figure 5. While this function will work as an objective, it is very extreme in the sense that there is a large jump from perfect communication at distances less than or equal to  $R_{ij}$  to no communication as soon as the distance becomes larger than  $R_{ij}$ . A more desirable function is one that approximates  $H_1$  but degrades in a continuous fashion from perfect to no communication.

We consider two alternatives to  $H_1$ . The first is a piecewise continuous, linear function defined by

$$H_2[R_{ij} - \|\vec{x}_t^i - \vec{x}_t^j\|_2] = \begin{cases} 1, & \text{if } \|\vec{x}_t^i - \vec{x}_t^j\|_2 \leq R_{ij} \\ 2 - \frac{\|\vec{x}_t^i - \vec{x}_t^j\|_2}{R_{ij}}, & \text{if } R_{ij} < \|\vec{x}_t^i - \vec{x}_t^j\|_2 \leq 2R_{ij} \\ 0, & \text{if } \|\vec{x}_t^i - \vec{x}_t^j\|_2 \geq 2R_{ij}. \end{cases} \quad (7)$$

FIGURE 5. The Heaviside function,  $H_1$ .FIGURE 6.  $H_2$ , continuous approximation to  $H_1$ .

This function, whose graph is provided in Figure 6, has a value equal to one if agents  $i$  and  $j$  are within the communication radius  $R_{ij}$  of one another. The function then decreases constantly until the agents have distance  $2R_{ij}$ , at which time they are unable to communicate.

The third and final objective function we will consider is a continuously differentiable decreasing function of the distance between agents  $i$  and  $j$ . This function, displayed in Figure 7, is defined by

$$H_3[\|\vec{x}_t^i - \vec{x}_t^j\|_2, R_{ij}] = e^{-\left(\frac{\|\vec{x}_t^i - \vec{x}_t^j\|_2}{R_{ij}}\right)^2}. \quad (8)$$

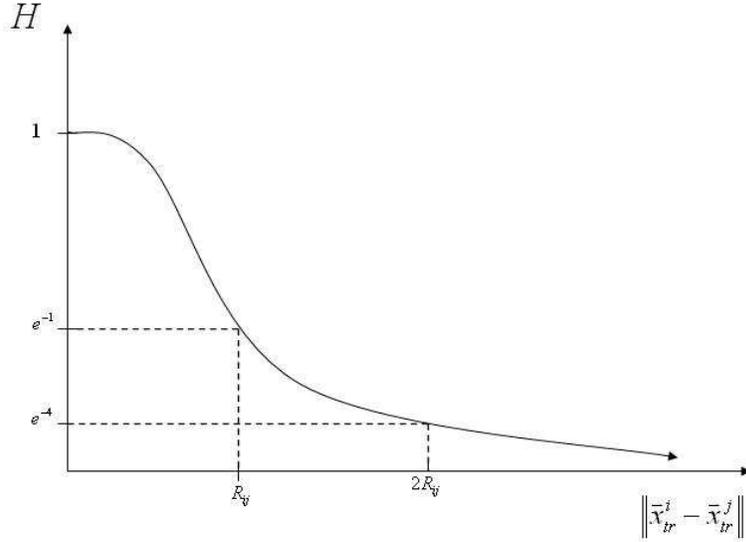


FIGURE 7.  $H_3$ , continuously differentiable approximation of  $H_1$ .

This is perhaps the best approximation of  $H_1$  in that it can be interpreted as the probability of agents  $i$  and  $j$  directly communicating as a function of the distance between them.

Now that we have found a suitable objective function we can define the remaining parameters and constraints of the problem. Let  $\bar{x}^i(t)$  be the position of agent  $i$  at time  $t$ . Similarly, let  $\bar{v}^i(t)$  be the velocity of agent  $i$  at time  $t$ . The relationship between velocity and position is the standard one, given by  $\bar{v}^i(t) = \frac{d\bar{x}^i(t)}{dt}$ . In order to formulate the continuous time analog of the CCPM, we must constrain the maximum speed of each agent. This is the continuous time analog of the constraints on the maximum distance traveled in the discrete formulation, between any two time steps. If  $s_i \in \mathbb{R}^2$  is the starting position of agent  $i$ , and  $d_i \in \mathbb{R}^2$  is the destination point of agent  $i$ , then we can formulate the CONTINUOUS COOPERATIVE COMMUNICATION PROBLEM ON MOBILE AD HOC NETWORKS (CCPM-C) as follows.

$$\max \int_0^T \sum_{i < j} H_3[\|\bar{x}^i(t) - \bar{x}^j(t)\|_2, R_{ij}] \quad (9)$$

s.t.

$$\bar{x}^i(0) = s_i, \forall i = 1, \dots, M \quad (10)$$

$$\bar{x}^i(T) = d_i, \forall i = 1, \dots, M \quad (11)$$

$$\|\bar{v}^i(t)\| \leq V_i, \forall i = 1, \dots, M, t \in [0, T] \quad (12)$$

$$\bar{x}^i(t) \in \mathbb{R}^2, \forall i = 1, \dots, M, t \in [0, T] \quad (13)$$

The above formulation provides a set of trajectories along which the agents can be routed in order to ensure that the communication between them is maximized. Clearly, if the agents remain in a tightly coupled formation then communication will be maximized; however, unlike the discrete version of the problem in which the vertices of a graph had to be traversed, in a continuous setting this problem is relatively easy and worse yet, not very interesting. Alternatively, consider a set of UAVs involved in a search-and-rescue or reconnaissance mission. For obvious reasons, missions of this sort generally require the UAVs to traverse a large portion of the battlespace before arriving at their destinations. In this case, the above formulation is not helpful. With this in mind, we move on to develop

a second continuous formulation which not only maximizes the communication between the agents, but also maximizes the coverage of predefined regions of the battlespace.

**4.2. Formulation 2: A Continuous Multiobjective Formulation.** In the following paragraphs, we derive a second continuous formulation which guarantees that certain locations will be visited by the UAVs as they traverse the battlespace from their sources to their respective destinations. Previous work on target visitation problems appear in [3]. Once again, we are considering a set of  $M$  UAVs. We keep the assumption that the  $i$ th UAV starts at a position  $s_i = (s_{ix}, s_{iy})$ , at time 0, and ends at position  $d_i = (d_{ix}, d_{iy})$ , at time  $T$ . The  $i$ th UAV, at time  $t \in [0, T]$ , has position  $\vec{x}^i(t) = (x_i(t), y_i(t))$ . Assume that the following holds:

$$\vec{x}^i(t) \in [x_{low}, x_{high}] \times [y_{low}, y_{high}] \quad \forall i = 1, \dots, M, t \in [0, T].$$

Furthermore, assume that there exists  $J$  positions in the domain, each of which must be visited by at least one UAV in the time interval  $[0, T]$ . These positions are given by  $Q_j = (\bar{x}_j, \bar{y}_j)$ , for each  $j = 1, \dots, J$ . Lastly, the  $i$ th UAV has a minimum and maximum speed given by  $\epsilon_i^{\min}$  and  $\epsilon_i^{\max}$  for each  $i = 1, \dots, M$ , respectively.

In order to implement a solution technique in a digital computer, we make use of the  $\mathcal{L}_1$ -norm as a measure of the distance between two points and discretize the time domain into  $\rho$  equal time steps,  $\Delta t = T/(\rho - 1)$ . Let  $t_k = k\Delta t$ , for each  $k = 0, \dots, \rho - 1$ . Thus the position of the  $i$ th UAV at time step  $k$  is given by  $\vec{x}^i(t_k) = (x_i(t_k), y_i(t_k))$ , for each  $i = 1, \dots, M$ , and for each  $k = 0, \dots, \rho - 1$ .

Then the problem, which is denoted as **CCPM-C**, can be written as:

$$\min \quad \sum_{i_1 < i_2} \sum_{k=0}^{\rho-1} [|x_{i_1}(t_k) - x_{i_2}(t_k)| + |y_{i_1}(t_k) - y_{i_2}(t_k)|] \quad (14)$$

s.t.

$$x_i(0) = s_{ix}, \quad y_i(0) = s_{iy}, \quad \forall i = 1, \dots, M \quad (15)$$

$$x_i(T) = d_{ix}, \quad y_i(T) = d_{iy}, \quad \forall i = 1, \dots, M \quad (16)$$

$$\epsilon_i^{\min} \leq \frac{1}{\Delta t} [|x_i(t_k) - x_i(t_{k-1})| + |y_i(t_k) - y_i(t_{k-1})|] \quad \forall i, \quad \forall k = 1, \dots, \rho - 1 \quad (17)$$

$$\epsilon_i^{\max} \geq \frac{1}{\Delta t} [|x_i(t_k) - x_i(t_{k-1})| + |y_i(t_k) - y_i(t_{k-1})|] \quad \forall i, \quad \forall k = 1, \dots, \rho - 1 \quad (18)$$

$$\beta_{ijk} [|x_i(t_k) - \bar{x}_j| + |y_i(t_k) - \bar{y}_j|] = 0 \quad \forall i, \quad \forall j \in J, \quad \forall k \quad (19)$$

$$\sum_{i=1}^M \sum_{k=0}^{\rho-1} \beta_{ijk} \geq 1 \quad \forall j \quad (20)$$

$$x_{low} \leq x_i(t_k) \leq x_{high} \quad y_{low} \leq y_i(t_k) \leq y_{high} \quad \forall i, \quad \forall k \quad (21)$$

$$\beta_{ijk} \in \{0, 1\} \quad \forall i, \quad \forall j, \quad \forall k. \quad (22)$$

**Theorem 1.** *The above formulation for CCPM-C is correct.*

*Proof.* Clearly, the objective function minimizes the pairwise distances between the agents. Thus as the distance between the agents decreases, the communication increases. Constraints (15) and (16) respectively specify the starting points and the destination points for the agents. Together, constraints (17) and (18) bound the speed of the agents. Next (19) and (20) ensure that at least one agent co-locates with the set of points in the domain which must be visited. More specifically, (19) implies that for all points  $j \in J$ , there must be a time when an agent occupies position  $j$ . In the constraint, this is accomplished by ensuring the distance between the visiting agent and point  $j$  is 0. Constraint (20) implies that at

least one agent must visits each point  $j \in J$ . Finally, constraints (21) and (22) define the domain of the decision variables.  $\square$

We can linearize the mixed integer programming formulation in (14)-(22) as follows. To begin with, replace the objective function (14) with

$$\sum_{i_1 < i_2} \sum_{k=0}^{\rho-1} \hat{x}_{i_1 i_2 k} + \hat{y}_{i_1 i_2 k} \quad (23)$$

with the additional constraints

$$\hat{x}_{i_1 i_2 k} \geq x_{i_1}(t_k) - x_{i_2}(t_k) \quad \forall i_1, i_2 = 1, \dots, M, i_1 < i_2, \quad \forall k = 0, \dots, \rho - 1 \quad (24)$$

$$\hat{x}_{i_1 i_2 k} \geq -[x_{i_1}(t_k) - x_{i_2}(t_k)] \quad \forall i_1, i_2 = 1, \dots, M, i_1 < i_2, \quad \forall k = 0, \dots, \rho - 1 \quad (25)$$

$$\hat{y}_{i_1 i_2 k} \geq y_{i_1}(t_k) - y_{i_2}(t_k) \quad \forall i_1, i_2 = 1, \dots, M, i_1 < i_2, \quad \forall k = 0, \dots, \rho - 1 \quad (26)$$

$$\hat{y}_{i_1 i_2 k} \geq -[y_{i_1}(t_k) - y_{i_2}(t_k)] \quad \forall i_1, i_2 = 1, \dots, M, i_1 < i_2, \quad \forall k = 0, \dots, \rho - 1 \quad (27)$$

Next, we replace (17) and (18) with

$$\epsilon_i^{\min} \leq \alpha_{ik} + \bar{\alpha}_{ik} \leq \epsilon_i^{\max} \quad (28)$$

adding the constraints

$$\alpha_{ik} \geq x_i(t_k) - x_i(t_{k-1}) \quad \forall i = 1, \dots, M, \quad \forall k = 0, \dots, \rho - 1 \quad (29)$$

$$\alpha_{ik} \geq -[x_i(t_k) - x_i(t_{k-1})] \quad \forall i = 1, \dots, M, \quad \forall k = 0, \dots, \rho - 1 \quad (30)$$

$$\bar{\alpha}_{ik} \geq y_i(t_k) - y_i(t_{k-1}) \quad \forall i = 1, \dots, M, \quad \forall k = 0, \dots, \rho - 1 \quad (31)$$

$$\bar{\alpha}_{ik} \geq -[y_i(t_k) - y_i(t_{k-1})] \quad \forall i = 1, \dots, M, \quad \forall k = 0, \dots, \rho - 1 \quad (32)$$

Finally, we replace (19) with

$$\phi_{ijk} + \bar{\phi}_{ijk} = 0 \quad (33)$$

and add the constraints

$$\theta_{ijk} \geq x_i(t_k) - \bar{x}_j, \quad \forall i, \forall j, \forall k \quad (34)$$

$$\theta_{ijk} \geq -[x_i(t_k) - \bar{x}_j], \quad \forall i, \forall j, \forall k \quad (35)$$

$$\bar{\theta}_{ijk} \geq y_i(t_k) - \bar{y}_j, \quad \forall i, \forall j, \forall k \quad (36)$$

$$\bar{\theta}_{ijk} \geq -[y_i(t_k) - \bar{y}_j], \quad \forall i, \forall j, \forall k \quad (37)$$

$$\phi_{ijk} \leq \beta_{ijk}(x_{high} - x_{low}), \quad \forall i, \forall j, \forall k \quad (38)$$

$$\phi_{ijk} \geq 0, \quad \forall i, \forall j, \forall k \quad (39)$$

$$\phi_{ijk} \leq \theta_{ijk}, \quad \forall i, \forall j, \forall k \quad (40)$$

$$\phi_{ijk} \geq \theta_{ijk} - [1 - \beta_{ijk}][x_{high} - x_{low}], \quad \forall i, \forall j, \forall k \quad (41)$$

$$\bar{\phi}_{ijk} \leq \beta_{ijk}(y_{high} - y_{low}), \quad \forall i, \forall j, \forall k \quad (42)$$

$$\bar{\phi}_{ijk} \geq 0, \quad \forall i, \forall j, \forall k \quad (43)$$

$$\bar{\phi}_{ijk} \leq \bar{\theta}_{ijk}, \quad \forall i, \forall j, \forall k \quad (44)$$

$$\bar{\phi}_{ijk} \geq \bar{\theta}_{ijk} - [1 - \beta_{ijk}][y_{high} - y_{low}], \quad \forall i, \forall j, \forall k \quad (45)$$

Thus the problem becomes one of minimizing (23) subject to the constraints (15), (16), (20)-(22), (24)-(45). The resulting formulation is a mixed integer linear program (MILP) and can be solved using a number of commercial software packages. In the following section, we present some preliminary results from one such package, as well as providing a discussion of the experiments.

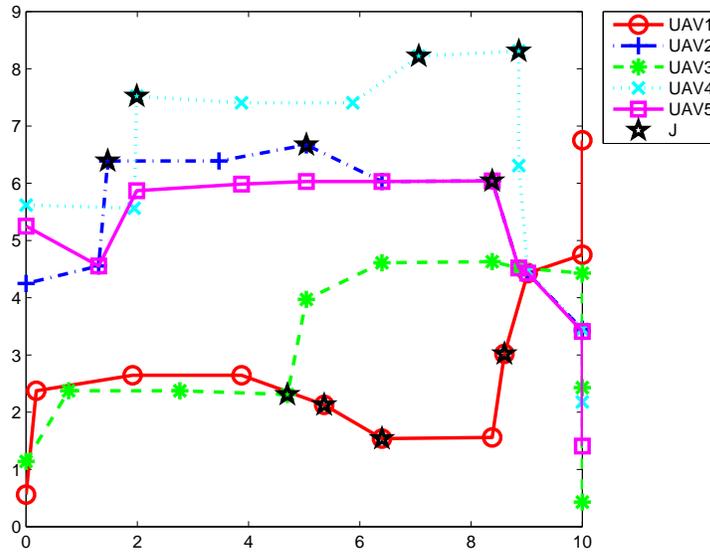


FIGURE 8. Example with 5 agents.

## 5. CASE STUDIES

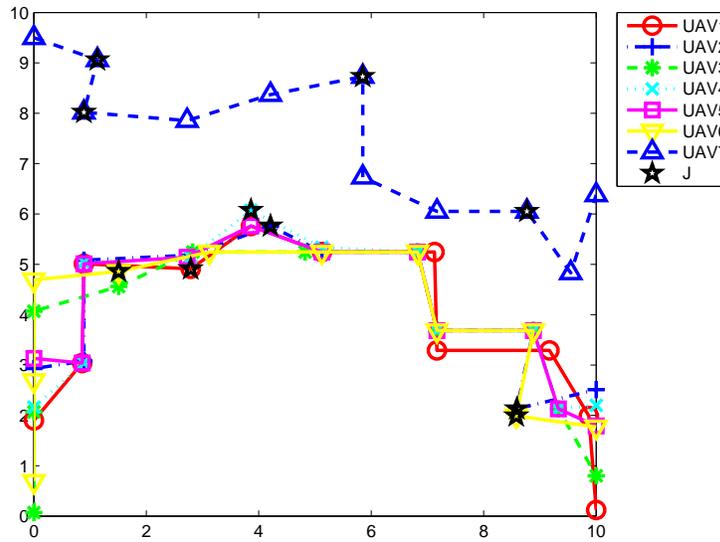


FIGURE 9. Example with 7 agents.

We have implemented the MILP formulation of the CCPM using the CPLEX<sup>TM</sup> optimization suite from ILOG [13]. CPLEX contains an implementation of the simplex method [18], and uses a branch and bound algorithm [28] together with advanced cutting-plane techniques [21, 24].

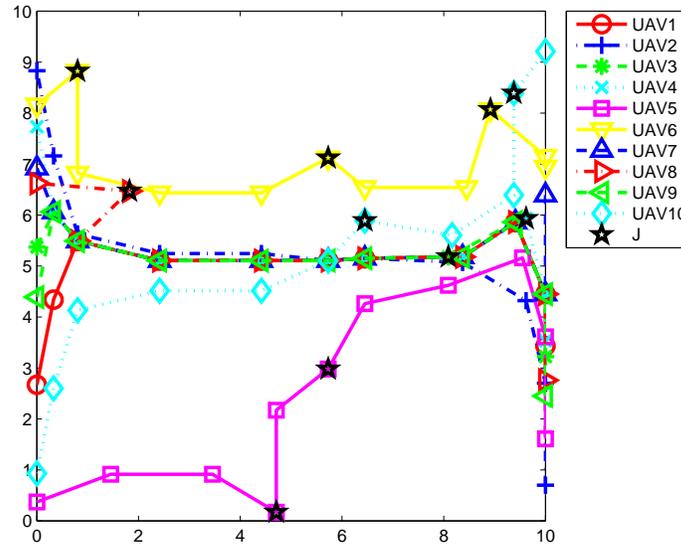


FIGURE 10. Example with 10 agents.

The instances were tested on grids of size 10. The set of coordinates,  $J$ , to be visited were generated uniformly at random. Three sets of coordinates were generated and each visited by three different sets of UAVs, numbering 5, 7, and 10. The  $y$ -coordinates of the starting and ending positions were also randomly generated using a uniform distribution and the  $x$ -coordinates were assumed to be 0 and 10 respectively. The scenarios were solved making use of the MILP formulation derived above. The optimal solutions were obtained for the instances with 5 UAVs. The instances with 7 UAVs and 10 UAVs were stopped at optimality tolerances of 10% and 25% respectively. A time-frame of 10 units was provided as an input and the minimum and maximum speed of the UAVs were 1 and 2 units respectively.

We have provided three graphical representations of the trajectories of the agents from each problem set. Figure 8 shows the paths traversed in one of the scenarios containing 5 agents. The movements of the agents are from left to right in the figure. The points which must be visited are denoted as stars. We see that from their starting points, the agents tend to converge into a tight formation. Notice that UAV2 separates from the group around point (7, 6) in order to visit three “must visit” points and arrive at its destination. The remaining agents travel together until they must diverge to reach their destinations.

In Figure 9 we have an example scenario containing 7 agents. As before, the agents quickly converge to a tightly coupled formation with agents leaving the group only to visit the points in  $J$  or arrive at their destinations. A 10 agent scenario is depicted in Figure 10 and similar behaviors of the agents can be observed. Figures 11-16 show how the paths of the agents change (from the 10 agent scenario in Figure 10) as agents are randomly removed from the scenario. As expected, the remaining agents are forced to spread out in order to ensure visiting the  $J$  target points.

The results presented are very promising. Indeed, the agents exhibit the exact behavior we would expect to see given the nature of the CCPM. As communication strength is inversely proportional to distance, the convergence to a common path clearly indicates that the UAVs are attempting to maximize the communication amongst the group. Consider the scenario in Figure 8. Notice that midway through the mission a very clear clustering

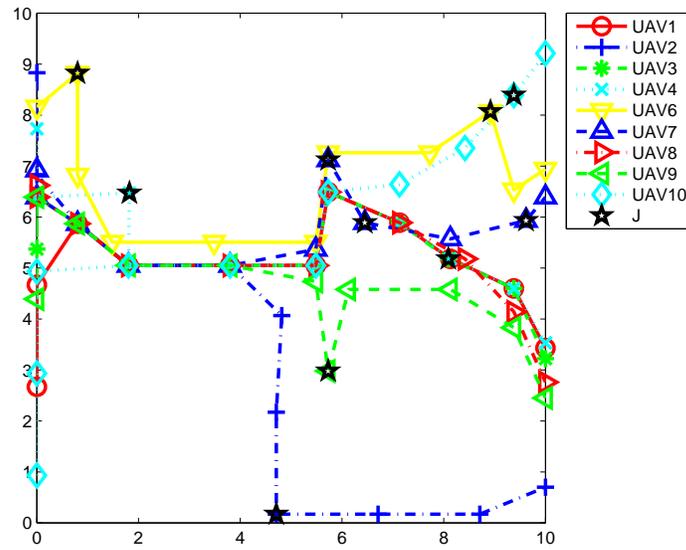


FIGURE 11. Example derived from 10 agent example, with one agent removed.

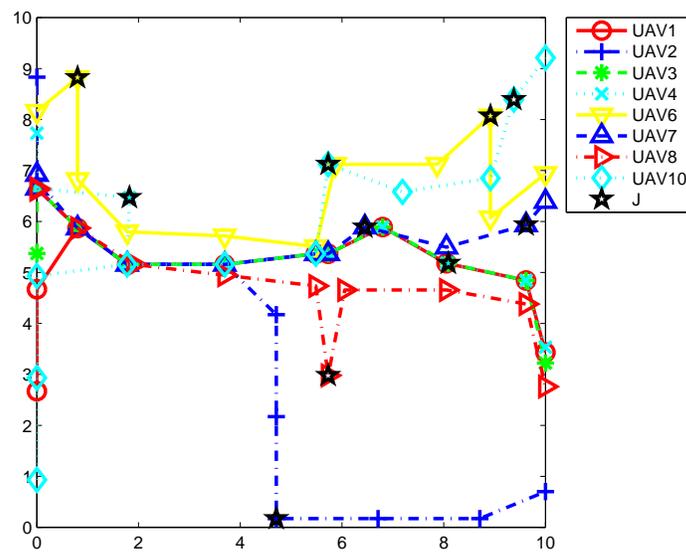


FIGURE 12. Example derived from 10 agent example, with two agents removed.

effect can be seen as two distinct groups of agents make their way towards the “must visit” points and ultimately, their destinations.

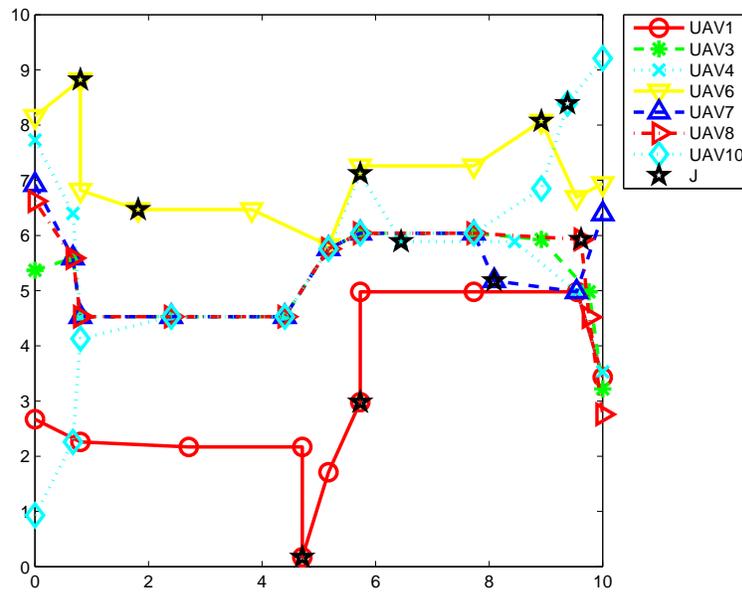


FIGURE 13. Example derived from 10 agent example, with three agents removed.

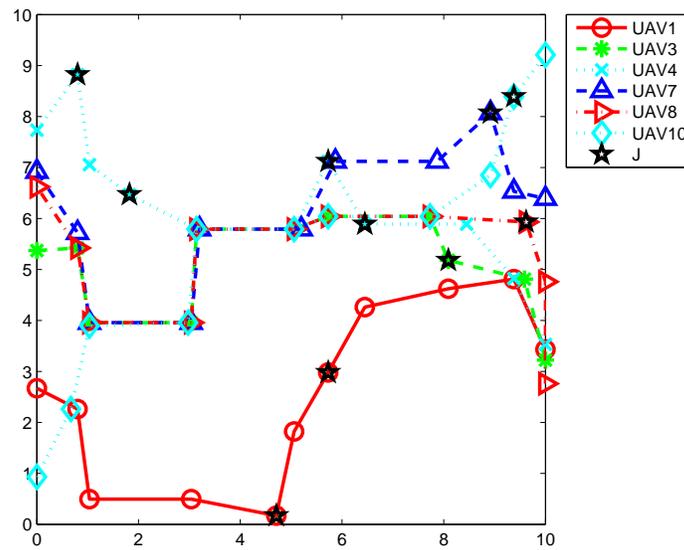


FIGURE 14. Example derived from 10 agent example, with four agents removed.

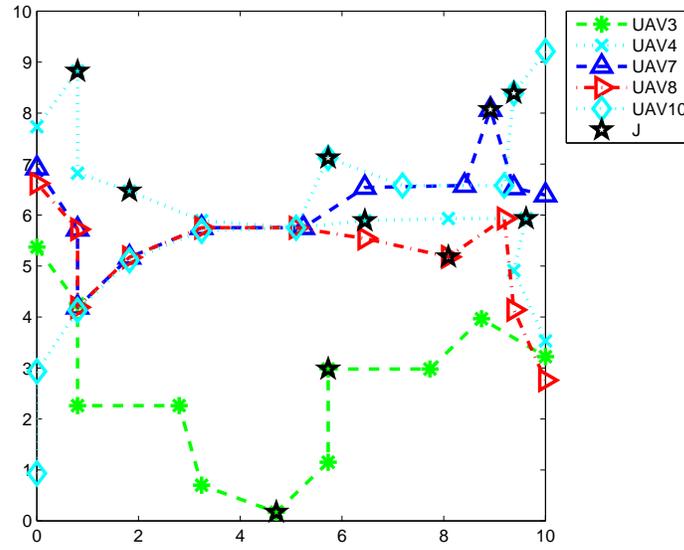


FIGURE 15. Example derived from 10 agent example, with five agents removed.

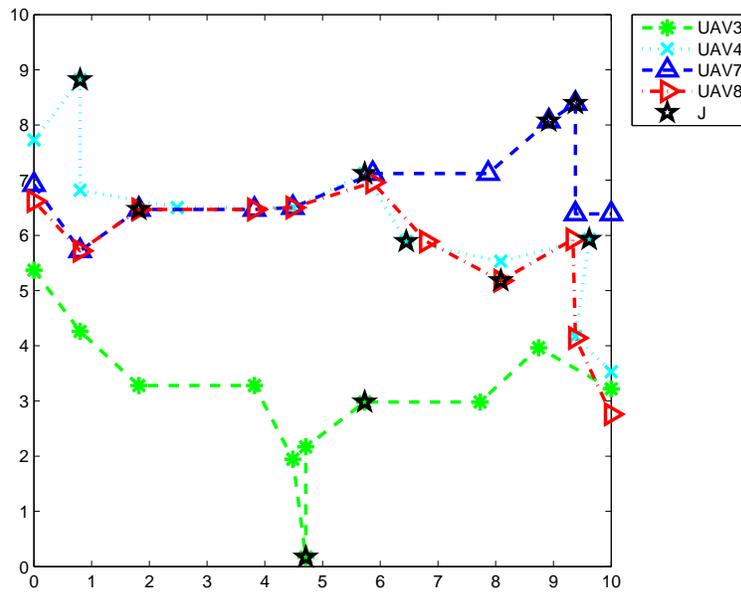


FIGURE 16. Example derived from 10 agent example, with six agents removed.

## 6. CONCLUSION

In this paper, we provided a review of recent work in the area of cooperative communication in mobile ad hoc networks. While inherently a problem of path planning, our formulations incorporated communication as a measure of the fitness of a given solution. We presented some discrete versions of the problem and derived two continuous formulations, the first time this has been considered. The advantage of the new models is that they ensure that a specified amount of the battlespace is explored by the agents. This addition is important in real-world applications particularly in the areas of surveillance, reconnaissance, and rescue operations. The preliminary numerical results demonstrate the effectiveness of the proposed models.

Due to the inherent complexity of the problem, future research will focus on continuous heuristic techniques for the newly proposed models, similar to those found in [19, 20]. Percentile risk constraints will be incorporated into the formulation. Commonly applied in financial applications, risk measures such as Value-at-Risk (VaR) and Conditional Value-at-Risk have proven to be effective tools for military applications as well [8, 11, 12]. We also plan to provide some theoretical results regarding feasibility of problem instances.

## REFERENCES

- [1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.
- [2] A. Arulselvan, C.W. Commander, M.J. Hirsch, P.M. Pardalos, and M.G.C. Resende. Cooperative communication in ad-hoc networks. In *National Fire Control Symposium Proceedings, 2007*.
- [3] A. Arulselvan, C.W. Commander, and P.M. Pardalos. A random keys based genetic algorithm for the target visitation problem. In M.J. Hirsch, P.M. Pardalos, R. Murphey, and D. Grundel, editors, *Advances in Cooperative Control and Optimization*. Springer, Inc., 2007.
- [4] S.I. Butenko, X. Cheng, C.A.S. Oliveira, and P.M. Pardalos. A new algorithm for connected dominating sets in ad hoc networks. In S. Butenko, R. Murphey, and P. Pardalos, editors, *Recent Developments in Cooperative Control and Optimization*, pages 61–73. Kluwer Academic Publishers, 2003.
- [5] S.I. Butenko, R.A. Murphey, and P.M. Pardalos, editors. *Cooperative Control: Models, Applications, and Algorithms*. Springer, 2003.
- [6] S.I. Butenko, R.A. Murphey, and P.M. Pardalos, editors. *Recent Developments in Cooperative Control and Optimization*. Springer, 2004.
- [7] B.N. Clark, C.J. Colbourn, and D.S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.
- [8] C.W. Commander. *Optimization Problems in Telecommunications with Military Applications*. Ph.D. dissertation, University of Florida, August 2007.
- [9] C.W. Commander, C.A.S. Oliveira, P.M. Pardalos, and M.G.C. Resende. A greedy randomized algorithm for the cooperative communication problem on ad hoc networks. In *8th INFORMS Telecommunications Conference, 2006*.
- [10] C.W. Commander, C.A.S. Oliveira, P.M. Pardalos, and M.G.C. Resende. A one-pass heuristic for cooperative communication in mobile ad hoc networks. In D.A. Grundel, R.A. Murphey, P.M. Pardalos, and O.A. Prokopyev, editors, *Cooperative Systems: Control and Optimization*. Springer, 2007.
- [11] C.W. Commander, P.M. Pardalos, V. Ryabchenko, S. Uryasev, and G. Zrazhevsky. The wireless network jamming problem. *Journal of Combinatorial Optimization*, to appear, 2007.
- [12] C.W. Commander, P.M. Pardalos, O. Shylo, and S. Uryasev. Recent advances in eavesdropping and jamming communication networks. In P.M. Pardalos D.A. Grundel, R.A. Murphey and O.A. Prokopyev, editors, *6th International Conference on Cooperative Control and Optimization*. World Scientific, 2006.
- [13] ILOG CPLEX. <http://www.ilog.com/products/cplex>, Accessed October 2006.
- [14] P. Festa and M.G.C. Resende. GRASP: An annotated bibliography. In C. Ribeiro and P.Hansen, editors, *Essays and surveys in metaheuristics*, pages 325–367. Kluwer Academic Publishers, 2002.
- [15] R.W. Floyd. Algorithm 97 (shortest path). *Communications of the ACM*, 5(6):345, 1962.
- [16] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.
- [17] D.A. Grundel, R.A. Murphey, and P.M. Pardalos, editors. *Theory and Algorithms for Cooperative Systems*. World Scientific, 2004.
- [18] F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research*. McGraw Hill, 2001.
- [19] M.J. Hirsch. *GRASP-based heuristics for continuous global optimization problems*. Ph.D. dissertation, University of Florida, December 2006.

- [20] M.J. Hirsch, C.N. Meneses, P.M. Pardalos, and M.G.C. Resende. Global optimization by Continuous GRASP. *Optimization Letters*, 1(2):201–212, 2007.
- [21] R. Horst, P.M. Pardalos, and N.V. Thoai. *Introduction to Global Optimization*, volume 3 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, 1995.
- [22] R.A. Murphey and P.M. Pardalos, editors. *Cooperative Control and Optimization*. Springer, 2002.
- [23] C.A.S. Oliveira and P.M. Pardalos. An optimization approach for cooperative communication in ad hoc networks. Technical report, School of Industrial Engineering and Management, Oklahoma State University, 2005.
- [24] C.A.S. Oliveira, P.M. Pardalos, and T.M. Querido. A combinatorial algorithm for message scheduling on controller area networks. *International Journal of Operations Research*, 1(1/2):160–171, 2005.
- [25] M.G.C. Resende and P.M. Pardalos. *Handbook of Optimization in Telecommunications*. Springer, 2006.
- [26] M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219–249. Kluwer Academic Publishers, 2003.
- [27] S. Warshall. A theorem on boolean matrices. *Journal of the ACM*, 9(1):11–12, 1962.
- [28] L. Wolsey. *Integer Programming*. Wiley, 1998.

(A. ARULSELVAN) CENTER FOR APPLIED OPTIMIZATION, DEPT. OF INDUSTRIAL AND SYSTEMS ENGINEERING, UNIVERSITY OF FLORIDA, GAINESVILLE, FL, USA.

*E-mail address:* ashwin@ufl.edu

(C.W. COMMANDER) AIR FORCE RESEARCH LABORATORY, MUNITIONS DIRECTORATE, AND, DEPT. OF INDUSTRIAL AND SYSTEMS ENGINEERING, UNIVERSITY OF FLORIDA, GAINESVILLE, FL USA.

*E-mail address:* clayton.commander@eglin.af.mil

(M.J. HIRSCH) RAYTHEON, INC., NETWORK CENTRIC SYSTEMS, ST. PETERSBURG, FL, USA.

*E-mail address:* michael\_j\_hirsch@Raytheon.com

(P.M. PARDALOS) CENTER FOR APPLIED OPTIMIZATION, DEPT. OF INDUSTRIAL AND SYSTEMS ENGINEERING, UNIVERSITY OF FLORIDA, GAINESVILLE, FL, USA.

*E-mail address:* pardalos@ufl.edu