Cooperative Real-Time Search and Task Allocation in UAV Teams

Yan Jin, Ali A. Minai, Marios M. Polycarpou
ECECS Department
University of Cincinnati
Cincinnati, OH 45221-0030

***Abstract*— In this paper, we consider a heterogeneous team of UAVs drawn from several distinct classes and engaged in a search and destroy mission over an extended battlefield. Several different types of targets are considered. Some target locations are suspected *a priori* with a certain probability, while the others are initially unknown. During the mission, the UAVs perform** *Search, Confirm, Attack* **and** *Battle Damage Assessment (BDA)* **tasks at various locations. The target locations are detected gradually through search, while the tasks are determined in real-time by the actions of all UAVs and their results (e.g., sensor readings), which makes the task dynamics stochastic. The tasks must, therefore, be allocated to UAVs in real-time as they arise. Each class of UAVs has its own sensing and attack capabilities with respect to the different target types, so the need for appropriate and efficient assignment is paramount.**

We present a simple cooperative approach to this problem, based on distributed assignment mediated through centralized mission status information. Using this information, each UAV assesses the task opportunities available to it, and makes commitments through a phased incremental process. This produces a simple, flexible, scalable and inherently decentralizable method for task allocation. Concurrently, every UAV also monitors the degree to which various parts of the environment have been searched, and accommodates this information in planning its paths. We study the effect of various decision parameters, target distributions, and UAV team characteristics on the performance of our approach.

I. INTRODUCTION

Motivated by recent advances in intelligent systems and cooperative control, many researchers have been studying large groups of *unmanned autonomous vehicles* or *UAVs* acting in teams to accomplish difficult missions in dynamic, poorly known or hazardous environments [15], [5], [8], [6], [7], [4], [1], [2], [3], [9], [11], [12], [13], [14], [16], [17], [19]. This paper presents an approach to this problem.

We consider a heterogeneous group of UAVs drawn from distinct classes and engaged in a search and destroy mission over an extended battlefield with both known and unknown targets. The UAVs must cooperatively search the environment, confirm suspected targets, discover and confirm new ones, attack them with appropriate munitions, and confirm their destruction. The task dynamics arises stochastically from the actions of the UAVs, and requires that tasks be assigned to appropriate UAVs as they arise. This creates a problem similar to the dynamic vehicle routing problem [18], [20], albeit of much greater complexity given the stochastic dynamics and several types of vehicles and tasks. In this paper, we present a simple model where UAVs choose tasks autonomously in real-time, using a central cognitive map that provides an instantaneous and accurate summary of the current situation as known to the UAV team. The approach we follow is motivated in part by the work of Chandler and Pachter, and their collaborators [5], [8], [6], [7], [4], [19]. It is also related to recent work by several other researchers [1], [2], [3], [9], [11], [12], [13]. A comprehensive overview of the research problems associated with UAV teams is available in [15].

A. Scenario and Model Description

We consider a $L_x \times L_y$ cellular environment, N UAVs , M stationary targets, γ_i, $i = 1,...,M$ with locations, (x_i^γ, y_i^γ), and no threats. Of the M targets, M_k are suspected initially, while $M_h = M - M_k$ must be discovered during search. The UAVs do not know the value of M_h or M, so the only way to ensure detection of all targets is to do a complete search. A canonical *task set*, $\mathbb{T}$, defines the tasks that the UAVs can undertake at a target location.

$$\mathbb{T} = \{Search, Confirm, Attack, BDA, Ignore\}$$

Each UAV, u_i, is characterized by an *expertise vector*, $\xi_i = \{\xi_{ij}\}, j = 1,\ldots,n, 0 \leq \xi_{ij} \leq 1$, where n is the number of possible tasks and ξ_{ij} indicates the UAV's expertise for task $T_j \in \mathbb{T}$. We number the tasks in the order given above, so, for example, $T_1 = search$ and $T_5 = Ignore$. The matrix Ξ with expertise vectors as rows is termed the *expertise matrix* for this problem.

UAVs move autonomously in the environment, scanning, communicating with other UAVs, making decisions, and performing tasks. At time, t, every cell, (x,y), in the environment has an associated *task status*, $T(x,y,t)$, indicating what needs to be done in that cell. The task status of all cells, $T(t) = \{T(x,y,t)\}$, represents the state of the environment, termed the *task state*. The dynamics of the task state is determined by the *target occupancy probability (TOP)*, $P(x,y,t)$, of each cell, (x,y), defined as the estimated probability that the cell contains a live target.

The *confirm*, *attack* and *BDA* tasks are called *assignable tasks*, i.e., tasks for which the UAVs are assigned explicitly. Such UAVs move purposively to the locations of their assigned tasks and perform them. The *search* and *ignore* tasks are termed *automatic tasks*, i.e., any UAV passing through a cell with one of these task statuses automatically performs the indicated task. However, UAVs do not actively bid for these tasks. The *search* task does have an effect on UAV movements as described below.

All locations with assignable tasks at time t form the set, $L(t)$, of *current target locations (CTLs)*. The task, τ_j, at each CTL, (x_j, y_j), has an *assignment status*, A_j, which can take on the values from the set

$$\mathbb{A} = \{available, associated, assigned, active, complete\}$$

The assignment status indicates whether the task is open for bidding (available), has been provisionally assigned to a UAV (associated), has been firmly assigned to a UAV (assigned), is being currently performed by a UAV at the location (active), or has been finished (complete). A completed task is accompanied by an immediate transition in the task status of the CTL.

B. UAV State

The *state*, $S_i(t)$, of a UAV, u_i, at time t has two parts:

- A *physical state*, including information on its position, $\lambda_i(t)$, and orientation, $\delta_i(t)$.
- A *functional state*, indicating the identity and location of the specific task (if any) to which the UAV is committed or has bid for, the corresponding *commitment status* (see below), and the UAV's expected cost for performing this task. The commitment status, $K_i(t)$, of UAV u_i takes values from the set:

  $$\mathbb{S} = \{open, competing, committed\}$$

  indicating whether the UAV has no commitment (open), has bid on a task or been associated with one (competing), or is assigned to a task and, possibly, is performing it (committed). The functional state of an open UAV has NULL values in its other fields. The *search* and *ignore* tasks require no commitment, and correspond to an *open* functional state.

C. Task Dynamics

As u_i moves in the environment, it performs an *action*, $a_i(t)$, in each cell, $(x_i(t), y_i(t))$, that it visits. A canonical action set is denoted by $\mathbb{D}$, which include sensor readings, firing of munitions and null actions. The action $a_i(t)$ is selected from $\mathbb{D}$ by an *action selection function*, G; i.e.,

$$a_i(t) = G(T(x_i(t), y_i(t), t), S_i(t)) \qquad (1)$$

If the action is a sensor reading, it returns an *observation value*, $b_i(t)$, which is a stochastic quantity. We also denote by $a(x,y,t)$ the set of actions performed by all UAV's in cell (x,y) at time t, and by $b(x,y,t)$ the set of observations (if any) taken by the UAVs. This determines the updates of the TOP value at (x,y) through a possibly stochastic *TOP update function*, F:

$$P(x,y,t+1) = F(P(x,y,t), T(x,y,t), a(x,y,t), b(x,y,t)) \qquad (2)$$

If multiple UAVs occupy the cell at time, updates due to their actions are applied sequentially. The TOP value, in turn, determines the dynamics of the cell's task status, modelled as a deterministic automaton whose transitions depend on threshold crossings in $P(x,y,t)$ (Figure 1):

$$T(x,y,t+1) = H(T(x,y,t), P(x,y,t+1); \theta) \qquad (3)$$

where the parameter vector θ represents the set of threshold values used for transitions.The dynamics is made stochastic by the stochasticity of $b(x,y,t)$ and the TOP update function, F. We define $F(\cdot)$ separately for each task status:

Task 1: Search: UAV, u_i, makes a sensor reading $b_i(x,y,t) = 1$ if the sensor detects a target and $b_i(x,y,t) = 0$ is it does not. The accuracy of the sensor is characterized by

$$\alpha = \frac{P(b_t|A)}{P(b_t|\overline{A})}$$

where A is the event that a target is actually located in the cell being scanned. The TOP is updated based on a Bayesian formulation [10]:

$$\begin{aligned} P(x,y,t+1) &= (1 - b_i(x,y,t))\frac{1 - P(x,y,t)}{1 + (\alpha - 1)P(x,y,t)} \\ &\quad + b_i(x,y,t)\frac{\alpha P(x,y,t)}{1 + (\alpha - 1)P(x,y,t)} \end{aligned} \qquad (4)$$

A cell with the *search* status transitions to *ignore* if its TOP falls below the *resolution threshold*, p_r, and to *confirm* if the TOP exceeds the *suspicion threshold*, p_s.

Task 2: Confirm: The cuing of a *confirm* task at cell (x,y) indicates that a UAV with the appropriate sensors should move towards the cell and scan it. All cells with *a priori* suspected targets are initialized with the *confirm* task and given a TOP of p_s. The *confirm* task is functionally identical to *search*, and the TOP update function is the same (Equation (4)). However, unlike *search*, it is assignable to UAVs with the appropriate expertise. The sensors used may also be different in the two cases. The cell transitions to *search* if its TOP falls below p_s (as a result of failure to confirm suspicions), and to *attack* if the TOP exceeds the *certainty threshold*, p_c.

Task 3: Attack: The *attack* status indicates that an appropriately armed UAV should proceed to the location and attack the target there with the correct munition. The attacking UAV then changes the TOP for the location as:

$$P(x,y,t+1) = P(x,y,t)(1 - P_s) \qquad (5)$$

where $0 \leq P_s \leq 1$ is the probability that the target is destroyed in the attack. Different types of UAVs can have different values of P_s for different target types. If the updated TOP exceeds the *exit threshold*, p_e, the cell transitions to status *BDA*.

Task 4: BDA: The purpose in the BDA task is to verify that the TOP has indeed fallen below p_e. Like *search* and

$confirm$, this is a purely observational task, and uses the same update equation (4). If the result of the update produces $P(x,y,t+1) \geq p_e$, the cell transitions back to *attack*; if $p_r \leq P(x,y,t+1) \leq p_e$, it transitions to *search*; and if $P(x,y,t+1) < p_r$, the cell transitions to *ignore*. The last designation allows search to focus in regions where targets are likelier. It also allows a mission termination condition to be defined concisely.

Task 5: Ignore: This status applies to cells that are known *a priori* to harbor no targets or where the absence of targets has been confirmed.

Figure 1 shows the transitions between states using an automaton formulation.

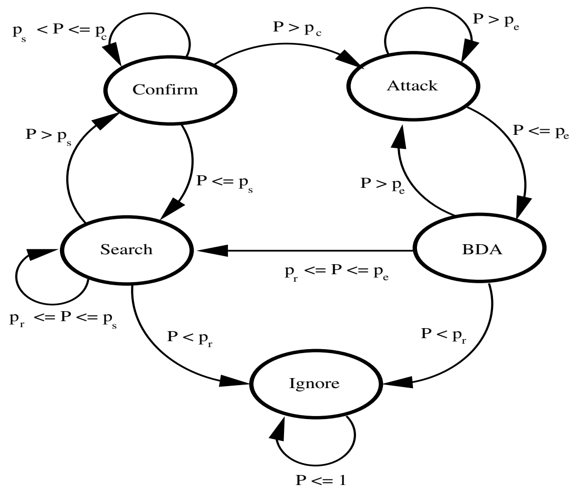

Fig. 1. Task Dynamics. Where, p_s=suspicion threshold, p_c=certainty threshold, p_e=exit threshold, p_r=resolution threshold.

D. Certainty Dynamics

In order to direct the search for targets, it is important to quantify the degree to which each cell's TOP is based on knowledge rather than ignorance. We do this by defining a *certainty* variable, $\chi(x,y,t) \in [0,1]$, for each (x,y). The initial value, $\chi(x,y,0)$, is based on the *a priori* information about the occupancy of (x,y) (e.g., if all targets are land-based, locations corresponding to water may begin with $P(x,y,0) = 0$ and $\chi(x,y,0) = 1$). Locations where targets are possible begin with $\chi(x,y,0) = 0$. Each observation by a UAV in (x,y) updates the certainty as:

$$\begin{aligned} \chi(x,y,t+1) &= \chi(x,y,t) + 0.5(1 - \chi(x,y,t)) \\ &= 0.5(1 + \chi(x,y,t)) \end{aligned}$$

This formulation [22] provides a simple way to track the number of useful "looks" each location has had and captures the notion of diminishing returns with each look.

II. Assignment Algorithm

In this paper, we consider UAVs drawn from two classes: Target recognition (TR) UAVs; and attack (A) UAVs. All UAVs are assumed to have sensors needed for search. UAVs of class C have a class-specific expertise vector, $[\xi_j^C]$, with respect to the five tasks, T_j, $j = 1,...,5$, in the task set. The expertise of class C UAV u_i for task T_j is, therefore, denoted by ξ_{ij}^C. All ξ_{ij}^C are between 0 and 1. In keeping with the capability designations, we set: $\xi_1^{TR} = \xi_1^A$, $\xi_2^{TR} > \xi_2^A$, $\xi_3^{TR} < \xi_3^A$, $\xi_4^{TR} > \xi_4^A$, $\xi_5^{TR} = \xi_5^A$. Thus, TR UAVs are needed for the $confirm$ and *BDA* tasks, while A UAVs are needed for attack.

The UAVs' mission is to *search* all cells that are not designated *ignore*, and to perform $confirm$, $attack$ and *BDA* tasks on each target known or discovered through *search*. For each task, the team must try to use UAVs best suited to it.

All UAVs have instantaneous and noise-free access to a centralized *information base (IB)*, which comprises the following items:

1) The TOP map $P(x,y,t)\ \forall(x,y)$.
2) The certainty map $\chi(x,y,t)\ \forall(x,y)$.
3) The task status map $T(x,y,t)\ \forall(x,y)$.
4) The assignment status map $A(x,y,t)\ \forall CTL(x,y)$.
5) The UAV state vector, $S(t) = \{S_i(t)\}\ \forall u_i$.

Each UAV reads and updates the IB at each step.

The initial TOP map has $P(x,y,0) < p_s$ (*search*) for cells where targets are not suspected, $P(x,y,0) = p_s$ ($confirm$) for suspected target locations, and $P(x,y,0) < p_r$ (*ignore*) for locations where targets are impossible. The UAVs' initial positions are also given. All UAVs initially have the open status.

The current set of assignable tasks is $T_s = \{\tau_k\}$, and j_k denotes the *identity* of task τ_k, i.e., whether it is $confirm$ ($j_k = 2$), *attack* ($j_k = 3$), or *BDA* ($j_k = 4$). The initial assignment is done as follows:

Each UAV u_i of class C calculates a *cost value*, h_{ik}, with respect to all *available* or *associated* assignable tasks, τ_k:

$$h_{ik} = \omega_1 * d_{ik} + (1 - \omega_1) * \exp(-\xi_{ij_k}^C) \tag{6}$$

where $0 \leq \omega_1 \leq 1$, d_{ik} is the normalized distance between UAV u_i and the location of task τ_k, and $\xi_{ij_k}^C$ is the expertise of UAV u_i for task j_k. When a UAV has no task to choose (it happens when all the targets are neutralized or the UVA is ineligible for all *available* or *associated* assignable tasks), it can choose the neighboring cell with the lowest certainty value(**search driven algorithm**) or choose a neighboring cell randomly(**non-search driven algorithm**).

Each UAV reports its best choice to the central controller. UAVs that are sufficiently close to their preferred tasks are assigned these tasks, while other UAVs compete for the remaining choices until every UAV has an initial assignment. When two UAVs prefer the same task, the conflict is resolved in favor of the UAV with smaller cost. Thus, initial assignment is purely cooperative and semi-greedy.

After the initial assignment, each UAV moves towards its *assigned* or *associated* task, updating the TOP in each cell it passes. When it reaches its *assigned* task, it performs the task and updates the TOP there. A new task (possibly the same as the previous one) is then cued at the CTL, and the UAV's status reverts to *open*. Locations can become CTLs if *search* raises their TOP above p_s, corresponding to the "discovery" of a new target. Each new assignable task is cued with an $available$ status.

At all times, all *open* and *competing* UAVs are considered for all *available* and *associated* tasks. The UAVs are processed in a randomized sequence according to the algorithm used for the initial assignment. The process continues until all locations have an *ignore* status or some time threshold is met.

While *committed* UAVs move directly to their tasks, *open* UAVs move by following the most locally productive search direction, as determined through the certainty variable: The UAV compares the certainty values for all possible next positions and always moves to the one with the lowest certainty. Ties are broken randomly. In other work [22], [21], we have considered more sophisticated approaches for determining efficient search paths.

III. Performance Measures

The goal for the UAV team is to cover the environment as rapidly as possible in such a way that all cells reach the *ignore* task status, i.e., all cells are completely searched and all targets neutralized. Specifically, we consider two measures to quantify performance:

1) The **target neutralization time (TNT)**, which is the time needed to neutralize all a priori known targets.
2) The **total mission time (TMT)**, which is the total number of steps needed to bring all cells to the *ignore* status.

IV. Simulation Results

In the first simulation (Figure 2), we consider a 20×20 cellular environment with 10 *a priori* suspected targets. The total number of UAVs is 10 (fixed), but λ (the ratio of TR UAVs to A UAVs) is varied from $1:9$ to $9:1$. The data for each case is averaged over 100 independent runs with random target configurations. Figure 2 shows that the value of λ has a significant effect on the TNT but little effect on TMT. This reflects the fact that all UAVs are equally capable of *search*, but have differential abilities for $confirm$ and *attack* tasks. Furthermore, there exists a "best" λ such that the TNT is smallest. Thus, given the total number of UAVs, one can find a UAV team composition that provides maximal target neutralization efficiency while giving up nothing in search efficiency.

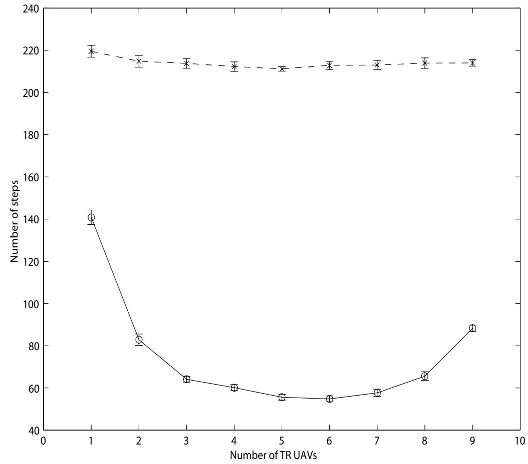

Fig. 2. The effect of the ratio of two types of UAVs on TNT and TMT. 20×20 cellular environment, 10 suspicious targets, 10 UAVs. Solid line for TNT, and dashed line for TMT.

In the second simulation, we compare the performance of search driven and non-search driven algorithms (defined in Section *II*). A 20×20 cellular environment is considered here. There are 8 TR UAVs and 8 A UAVs. The total number of targets is 20, and the ratio of suspected targets and unknown targets is varied. Figure 3 shows the search driven algorithm is more efficient than the non-search driven one at neutralizing all the targets, especially when most targets are unknown initially. And figure 4 shows that the search driven algorithm is also more efficient than the non-search driven one at searching the whole region. The result is reasonable, since the UAVs are more likely to search the unknown region by using the search driven algorithm.

In the third simulation, we consider a structured environment with simplified dynamics for which the optimal behavior of a UAV team can be determined explicitly. We then compare the performance of our cooperative search and task allocation algorithm with this optimal performance, and also with a non-cooperative version of our algorithm. The specific environment is an $L \times L$ cellular one as shown in Figure 5. There is one TR UAV located at each corner along the main diagonal, while one A UAV is located at each corner along the antidiagonal. Four suspected targets are located at the vertices of a centered square with sides of size X. For simplicity, we assume that once a *committed* UAV reaches its task location and executes its task (in one time step), the task status of that location transitions deterministically to the next one. This allows an explicit determination of the optimal plan for each UAV so that overall mission time is minimized.

The theoretical optimal TNT can be achieved if the search and task allocation problem is solved off-line *a priori* in a centralized manner. Thus, each UAV is assigned a trajectory and a task schedule, which it executes without further

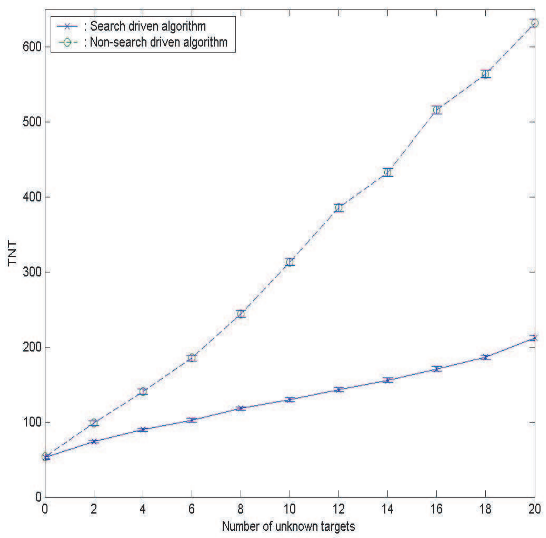

Fig. 3. Comparison of TNT of search driven and non-search driven algorithms. 20×20 cellular environment, 8 TR UAVs and 8 A UAVs, 20 targets.

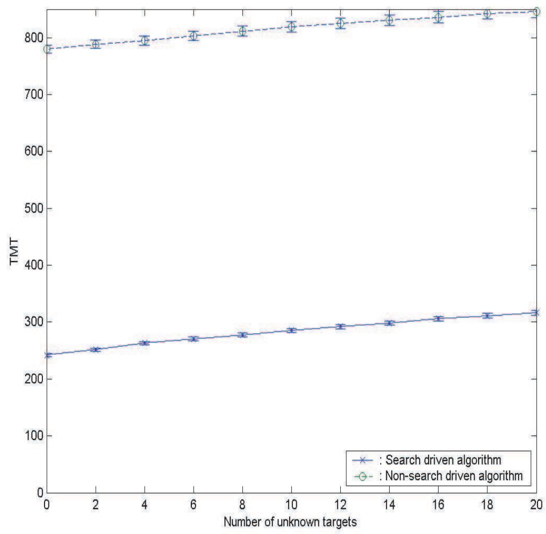

Fig. 4. Comparison of TMT of search driven and non-search driven algorithms. 20×20 cellular environment, 8 TR UAVs and 8 A UAVs, 20 targets.

decision-making. The optimal TNT in this case is given by $(L+5X+2)/2$. This obviously neglects the stochastic aspects of the complete model, but it does provide a valuable benchmark for evaluating the relative performance of our real-time algorithm.

We also evaluate the effect of cooperation in our task allocation algorithm by comparing it with a non-cooperative method. In this case, we assume that each UAV goes to its preferred task without considering the decisions of other UAVs, and moves away from it only when the task's status changes in response to another UAV accomplishing it first.

In the simulation, $L = 21$, while X is varied. The term "co-operative algorithm" refers to the task allocation algorithm presented in Section II. Figure 6 shows that cooperation significantly improves the performance of the UAVs, moving it roughly halfway towards the optimum.

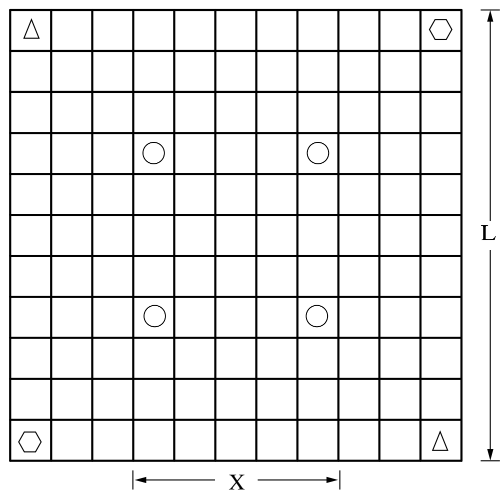

Fig. 5. The environment of the given example. Circles denote targets, triangulares denote TR UAVs , and sexangulares denote A UAVs.

V. Conclusion and Future Work

The model presented above is a simple attempt to formalize the UAV search-and-destroy problem in a way that is amenable to decentralization. The results are promising, and suggest several avenues for further exploration. These include:

- Inclusion of threats.
- Allowing each UAV bid simultaneously for several tasks, and using a multi-stage assignment process.
- Including prediction in the assignment process, so that UAVs can anticipate tasks likely to become available in the near future and include these in their plans.
- Incorporating learning and adaptation at the UAV and team levels, so that decision-making can improve with experience, and individual UAVs can develop specialized expertise.
- Using a dynamic expertise matrix to model losses in UAV capabilities due to munitions use or damage.
- Decentralizing the information base and the assignment process.

Work on these areas will be reported in the future.

VI. Acknowledgments

This work was supported by the AFRL/VA and AFOSR Collaborative Center of Control Science (Grant F33615-01-

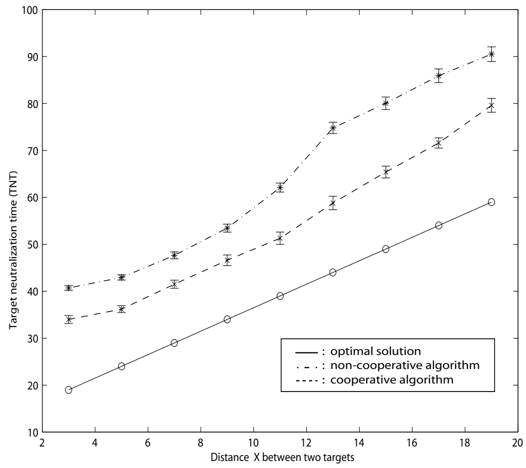

Fig. 6. *TNT* for the given example. 21×21 cellular environment, 2 *TR* UAVs and 2 *A* UAVs, 4 targets.

2-3154).

VII. REFERENCES

[1] R.W. Beard, T.W. McLain, and M. Goodrich. Coordinated target assignment and intercept for unmanned air vehicles. *Proc. ICRA'2002*, 3:2581–2586, 2002.

[2] R.W. Beard, T.W. McLain, M. Goodrich, and E.P. Anderson. Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Trans. On Robotics and Automation*, 18:911 –922, 2002.

[3] J. Bellingham, M. Tillerson, A. Richards, and J. How. Multi-task allocation and path planning for cooperative UAVs. In S. Butenko, R. Murphey, and P.M. Pardalos, editors, *Cooperative Control: Models, Applications and Algorithms*, pages 23–41. Kluwer Academic Publishers, 2003.

[4] P.R. Chandler and M. Pachter. Research issues in autonomous control of tactical UAVs. *Proc. ACC'1998*, pages 394–398, 1998.

[5] P.R. Chandler and M. Pachter. Hierarchical control for autonomous teams. *Proc. GNC'2001*, pages 632–642, 2001.

[6] P.R. Chandler, M. Pachter, and S. Rasmussen. UAV cooperative control. *Proc. ACC'2001*, 1:50–55, 2001.

[7] P.R. Chandler, M. Pachter, D. Swaroop, J.M. Fowler, J.K. Howlett, S. Rasmussen, C. Schumacher, and K. Nygard. Complexity in UAV cooperative control. *Proc. ACC'2002*, 3:1831 –1836, 2002.

[8] P.R. Chandler, S. Rasmussen, and M. Pachter. UAV cooperative path planning. *Proc. GNC'2000*, pages 1255–1265, 2000.

[9] M. Flint, M.M. Polycarpou, and E. Fernandez-Gaucherand. Cooperative control of multiple autonomous UAV's search for targets. *Proc. of the 41st IEEE Conference on Decision and Control*, pages 2823–2828, 2002.

[10] D.R. Jacques. Search, classification and attack decisions for cooperative wide area search munitions. In S. Butenko, R. Murphey, and P.M. Pardalos, editors, *Cooperative Control: Models, Applications and Algorithms*, pages 75–93. Kluwer Academic Publishers, 2003.

[11] Y. Jin, M.M. Polycarpou, and A.A. Minai. Cooperative real-time task allocation among groups of UAVs. *Proc. Cooperative Control and Optimization Workshop 2002*, (to appear).

[12] S.-M. Li, J.D. Boskovic, S. Seereeram, R. Prasanth, R. Amin, R.K. Mehra, and R.W. amd Mclain. T.W. Beard. Autonomous hierarchical control of multiple unmanned combat air vehicles (UCAVs). *Proc. ACC'2002*, pages 274–279, 2002.

[13] T.W. McLain and R.W. Beard. Trajectory planning for coordinated rendezvous of unmanned air vehicles. *Proc. GNC'2000*, (AIAA-2000-4369), 2000.

[14] T.W. McLain, R.W. Beard, and J.M. Kelsey. Experimental demonstration of multiple robot cooperative target intercept. *Proc. GNC'2002*, (AIAA-2002-4678), 2002.

[15] A. Moitra, R.J. Szczerba, V.A. Didomizio, L.J. Hoebel, R.M. Mattheyses, and B. Yamrom. A novel approach for the coordination of multi-vehicle teams. *Proc. GNC'2001*, pages 608–618, 2001.

[16] K. Passino. An introduction to research challenges in cooperative control for uninhabited autonomous vehicles. (preprint).

[17] M.M. Polycarpou, Y. Yang, and K. Passino. A cooperative search framework for distributed agents. *Proc. IEEE ISIC'2001*, pages 1–6, 2001.

[18] T. Schouwenaars, B. De Moor, E. Feron, and J. How. Mixed integer programming for multi-vehicle path planning. *Proc. European Control Conference 2001*, pages 2603–2608, 2001.

[19] C. Schumacher, P.R. Chandler, and S.R. Rasmussen. Task allocation for wide area search munitions via network flow optimization. *Proc. GNC'2001*, pages 619–626, 2001.

[20] K.C. Tan, L.H. Lee, Q.L. Zhu, and K. Ou. Heuristic methods for vehicle routing problem with time windows. *Intelligent in Engineering*, pages 281–295, 2002.

[21] Y. Yang, A.A. Minai, and M.M. Polycarpou. Decentralized opportunistic learing in UAV's performing cooperative search. *Proc. GNC'2002*, 2002.

[22] Y. Yang, M.M. Polycarpou, and A.A. Minai. Opportunistically cooperative neural learning in mobile agents. *Proc. IJCNN'2002*, pages 2638–2643, 2002.