Impact of System Design Parameters on Performance of Cooperative Agent Teams

Yan Jin[1], Yan Liao[1], Ali A. Minai[1], Marios M. Polycarpou[1,2]
[1]Dept. ECECS, University of Cincinnati, Cincinnati, OH 45221-0030
[2]Dept. ECE, University of Cyprus, 75 Kallipoleos, Nicosia 1678, Cyprus

***Abstract*— During the last few years several models of cooperative control among a team of agents have been proposed. In this paper, we study the impact of several system design parameters on a cooperative control methodology. An event-driven simulation model is developed to investigate how system design parameters affect the search and task performance of cooperative multi agent teams. In this study we focus on the problem of Uninhabited Air Vehicles (UAVs) searching for targets in an uncertain environment. We have previously presented a model-based decision-making and task-allocation methodology for this problem. In this paper, we investigate how the choice of two critically important parameters in the algorithm influence the performance of the UAV team. We evaluate the effect of these parameters on six different performance measures: 1) total mission time; 2) classification error rate; 3) target neutralization time; 4) task attempt times; 5) target miss ratio; and 6) task-driven traveling distance. The results provide valuable intuition on the selection of system design parameters based on different mission requirements.**

I. Introduction

The problem of cooperation and coordination among a team of agents has attracted considerate attention during the past few years [1], [2]. The applications range from homeland security to disaster surveillance. Advances in intelligent systems, computing power and wireless communications have made feasible the development of communities of Uninhabited Air Vehicles (UAVs) with the capacity to perform cooperative tasks in a coordinated manner [3], [4].

In this paper, we consider a group of heterogenous UAVs engaged in a target search, confirmation, neutralization and damage assess mission. Information about the location and type of some targets (called suspected targets) are assumed to be known a priori, while the location of other targets is assumed unknown and needs to be determined through cooperative search by the UAVs. Search path planning has been addressed in the robotics literature in the field of robot motion planning [5] and, in particular, within the subfields of terrain acquisition [6] and coverage path planning [7]. In recent years, there has been a great deal of work on search schemes for UAVs. Genetic algorithms [8], opportunistic neural learning [9], surrogate optimization [10])and dynamic programming [11] are some of the techniques that have been proposed for this problem. A search-theoretic approach based on the previous work of [12] is taken in [13]. This problem is posed as a pursuit-evasion game in [14].

The tasks of the UAV team include locating a target, confirmation, classification, neutralization of each target with the appropriate munitions, and verification of the damage impacted on the target. Since all the tasks are created and accomplished through the actions of the UAVs, the task dynamics evolves stochastically throughout the environment, and requires that tasks be assigned to appropriate UAVs as they pop up. This creates a problem similar to the dynamic vehicle routing problem [15], but of greater complexity due to the stochastic dynamics and uncertainty. Mixed-integer linear programming [16], Tabu search [17] and market-based allocation methods [18] have been considered for the problem. This assignment problem has also been viewed as a cooperative scheduling (resource allocation) problem in [19].

In previous work, we have reported results from a cooperative UAV team model where the tasks are chosen autonomously and cooperatively by the UAVs using a central subjective model, called a *cognitive map*, on the status of all tasks and UAVs [20], [21]. In this approach, UAVs carry out search as the default behavior, and take on target-specific tasks through a process of *cooperative gradual commitment*, beginning with volunteering and ending in assignment. UAVs assigned to specific tasks proceed directly to the task location instead of searching for new targets. While we have shown that the proposed algorithm works correctly and efficiently, it has not been clear how the basic parameters used in decision-making affect team performance. In this paper, we use the proposed model to study various issues associated with the effect of the most significant design parameters using Monte Carlo simulations.

For the purpose of completeness, the system models and cooperative search and task allocation algorithms are described briefly in Section II-IV. The readers can refer to our previous papers [20], [21] for more detail. The investigation of the impact of system design parameters on the performance of cooperative agent teams and Monte Carlo simulations are presented in Section V. Finally, the conclusion is given in Section VI.

II. Scenario Description

The mission environment is taken to be a continuous rectangular region of size $L_x\ km$ by $L_y\ km$. In this paper, lower-case x and y represents the Cartesian coordinates of

This work was supported by the AFRL/VA and AFOSR Collaborative Center of Control Science (Grant F33615-01-2-3154). Please address all correspondence to Ali A. Minai (aminai@ececs.uc.edu).

the continuous environment. For the purposes of sensing and cognitive map representation, the environment is divided into $N_x \times N_y$ cells. Upper case X and Y denotes the integer coordinates of the discretized cellular representation.

There are M stationary targets located in the environment. Each target, ν_i, $i = 1, \ldots, M$, has location, (x_i^ν, y_i^ν), and fixed orientation Φ_i^* relative to a globally defined frame of reference. These targets are drawn from N_T different target classes. Among the M targets, M_k are suspected initially, while $M_h = M - M_k$ need to be discovered gradually by search. There are n UAVs, u_i, $i = 1, \ldots, n$, operating in the environment, with the mission of discovering and destroying all targets. The tasks that the UAVs can undertake at a target location are in a canonical task set $\mathcal{T} = \{Search, Confirm, Attack, BDA\}$.

Each UAV, u_i, is characterized by two *expertise vectors*: 1) *sensing expertise matrix* $\xi_i^S = \{\xi_{ij}^S,\ j = 1, \ldots, N_T\}$, where ξ_{ij}^S indicates the UAV's expertise for sensing and identifying a target of type j; 2) *attack expertise matrix* $\xi_i^A = \{\xi_{ij}^A,\ j = 1, \ldots, N_T\}$, where ξ_{ij}^A indicates the UAV's capability for attacking a target of type j.

UAVs move autonomously through the environment in continuous time, 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*, $T(X, Y, t) = \{T_j(X, Y, t), j = 0, 1, \ldots, N_T\}$, indicating what needs to be done for each possible target type j. Here, $j = 0$ corresponds to the "no target" case. Each T_j can take values 1 (*search*), 2 (*confirm*), 3 (*attack*), and 4 (*BDA*). The task status of all cells, $T(t) = \{T_j(X, Y, t)\}$, represents the *task state* of the environment from the UAV team's viewpoint. The dynamics of the task state is determined by the *target occupancy probability (TOP)*, $P_j(X, Y, t)$, defined as the estimated probability that the cell contains a live target of type j, $j = 1, \ldots N_T$, and $P_0(X, Y, t)$ represents the estimated probability that there is no live target there. It is assumed that there is at most one live target located at a cell and no target crosses the boundary between two or more cells. The TOP of all cells, $P(t) = \{P_j(X, Y, t)\}$, is called the *TOP map* of the environment, and represents the UAV team's subjective estimates of target occupancy throughout the environment.

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. $Search$ is termed an *automatic task*, i.e., any UAV passing through a cell with *search* task status automatically performs search but UAVs do not actively bid for these tasks.

All cells with known assignable tasks at time t form the set, $L(t)$, of *current target locations* (CTLs). The task, τ_i, at each CTL, (X_i, Y_i), has an *assignment status*, A_i, which can take on the values from the set $\mathcal{A} = \{available, associated, assigned, active, complete\}$. A completed task is accompanied by a transition in the task status of the CTL.

III. Cognitive Map Definition

A. TOP Dynamics

The TOP map is updated in an event driven fashion by UAV *observations* and *actions*. When UAV u_i takes a sensor reading at time t, observations $b_i(X, Y, t) \in \{0,\ 1,\ \ldots,\ N_T\}$ are produced for all cells (X, Y) in its sensor field at the time. These are stochastic quantities, with $b_i(X, Y, t) = j$ indicating that UAV u_i detected a target of type j in cell (X, Y) at time t (recall that $j = 0$ corresponds to detecting no target). When a UAV u_i located within cell (X, Y) fires a munition at time t, it is denoted as an action, $a_i(X, Y, t)$. The observations and actions that occur in cell (X, Y) at time t are denoted, respectively, by $b(X, Y, t)$ and $a(X, Y, t)$. Together, they determine the updates of the TOP value at (X, Y) through a possibly stochastic ***TOP update function***, F:

$$\begin{aligned} &P(X, Y, t) \\ &\quad = F(P(X, Y, t^-), T(X, Y, t^-), a(X, Y, t), b(X, Y, t)) \end{aligned} \tag{1}$$

where t^- indicates the time immediately preceding the action or observation. If multiple UAVs take observations or actions in the same cell simultaneously, the updates are applied sequentially. F is defined as follows for the cases of observation and action:

Observation-Triggered TOP Update Function F_o**:** If UAV u_i makes a sensor reading $b_i(X, Y, t)$ using sensor resources $\zeta^i(t)$, the TOP for each target type is updated based on the Bayesian formulation [22]:

$$\begin{aligned} &P_j(X, Y, t) \\ &\quad = \frac{\lambda^i_{j, b_i(X,Y,t)}(\theta_S(t), \zeta^i(t)) P_j(X, Y, t^-)}{\sum_{l=0}^{N_T} \lambda^i_{l, b_i(X,Y,t)}(\theta_S(t), \zeta^i(t)) P_l(X, Y, t^-)} \end{aligned} \tag{2}$$

where θ_S is the *relative angle of observation (RAO)*, given by the angle between the UAV's heading and the target's estimated orientation, and $\lambda^i_{j,k}$ is a function characterizing the accuracy of the sensors used by u_i, defined as:

$$\lambda^i_{j,k}(\theta_S, \zeta^i) = Prob(b_i = k | E_j; \theta_S, \zeta^i).$$

Here, E_j is the event that a target of type j is actually located in the cell being scanned. So $\lambda^i_{j,k}(\theta_S, \zeta^i)$ quantifies the probability of observing a type j target as a type k target from an RAO of θ_S using sensor resources ζ^i. It is assumed that, given a target type and a sensor, there are some optimal angles of observation. A high-quality sensor would have $\lambda^i_{k,k}$ close to 1 for all k when the observation is made from an optimal angle for that target type, but not necessarily from another angle. This models the real situation when the accuracy of a given sensor depends on the type of target and angle of observation.

Action-Triggered TOP Update Function, F_a**:** If UAV u_i with munition resources $\mu^i(t)$ executes an attack in cell

(X, Y), the TOP for the cell is updated as:

$$P_j(X,Y,t) = (1-\beta_j^i(\theta_A(t),\mu^i(t)))P_j(X,Y,t^-),\quad for\ j = 1,2,\ldots,N_T \tag{3}$$

$$P_0(X,Y,t) = 1 - \sum_{l=1}^{N_T} P_l(X,Y,t) \tag{4}$$

where $0 \le \beta_j^i \le 1$ is the probability that the target of type j is destroyed by UAV u_i, $\theta_A(t)$ is the *relative angle of attack (RAA)*. As with observation-triggered updates, the TOP for all target types is updated after an action.

B. Task Dynamics

Changes in the TOP map determine the dynamics of the cell's task state. This is modeled as a deterministic automaton, H, whose transitions depend on threshold crossings in $P(X,Y,t)$ (Fig. 1):

$$T(X,Y,t) \; = \; H(T(X,Y,t^-), P(X,Y,t); \bar{\rho}) \tag{5}$$

where the parameter vector $\bar{\rho}$ represents the set of threshold values used for transitions. The dynamics is made stochastic by the random nature of $a(X,Y,t)$ and $b(X,Y,t)$. Fig. 1 shows the transitions between states using an automaton formulation. The task update function, H, is defined separately for each task status (refer to [20]).

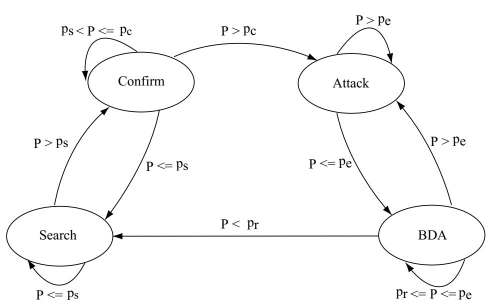

Fig. 1. Automaton formulation of the task dynamics.

C. Uncertainty Dynamics

In order to direct the search for targets efficiently, it is important to quantify how much is known about the existence of a live target in each cell as well as the type of the target. We do this by defining two types of uncertainty as follows:

- *Target Occupancy Uncertainty*, $\chi_o(X,Y,t)$, given by:

$$\chi_o(X,Y,t) = -P_0(X,Y,t)\log P_0(X,Y,t) - (1-P_0(X,Y,t))\log(1-P_0(X,Y,t))$$

- *Target Type Uncertainty*, $\chi_t(X,Y,t)$, given by:

$$\chi_t(X,Y,t) = -\sum_{l=1}^{N_T} \frac{P_l(X,Y,t)}{1-P_0(X,Y,t)} \log \frac{P_l(X,Y,t)}{1-P_0(X,Y,t)}$$

Based on the above definition, a uncertainty variable, $\chi(X,Y,t)$, for each cell (X,Y), is defined as follows:

$$\chi(X,Y,t) = \frac{\omega_\chi}{\log 2}\chi_o(X,Y,t) + \frac{(1-\omega_\chi)}{\log N_T}\chi_t(X,Y,t) \tag{6}$$

where, ω_χ is called the *uncertainty coefficient*, which takes values between 0 and 1. This combined entropy-like formulation provides a measure that represents how uncertain the UAV team is that a target exists in (X,Y) and about its type.

IV. Assignment Algorithm

The UAVs' mission is to $search$ all cells for all possible target types, 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 the UAV best suited to it.

All UAVs have instantaneous and noise-free access to a centralized *information base (IB)*, which comprises the TOP map, orientation map, uncertainty map, task status map, assignment status map and UAV state vector. Each UAV reads and updates the Information Base continuously. Please refer to [21] for the initialization of IB.

A. Initial Assignment

The initial assignment is done as follows:

Step 1: Each UAV u_i calculates a *cost value*, h_{ik}, with respect to all *available* or *associated* assignable tasks, τ_k:

$$h_{ik} = \omega_c d_{ik} + (1-\omega_c)\exp(-\xi_{im_k}) \tag{7}$$

where, ω_c is termed the *cost coefficient*, which takes values between 0 and 1, d_{ik} is the normalized distance between u_i and the location of task τ_k, and ξ_{im_k} is the expertise of UAV u_i for task m_k.

Step 2: Each UAV reports its cost for all tasks on the CTL for which it is eligible (UAV u_i is eligible for task m_k if the expertise $\xi_{im_k} \ge \xi_{min}$, where ξ_{min} is a non-negative parameter) to the central controller. Then a semi-greedy bipartite matching algorithm is employed to match UAVs with tasks.

Step 3: UAVs that are within distance D_a of their matched task are $assigned$ the task and are given the $committed$ status, while UAVs that are further away are $associated$ with their matched tasks and are given the $competing$ status. Only one UAV is allowed to be $assigned$ to a task but up to n_a UAVs can be $associated$ with a task. Similarly, each UAV can only be $committed$ to a single task, but can $compete$ for up to m_a tasks.

When a UAV has no task, it has $open$ status and follows a *path of maximum local uncertainty*, i.e., one that takes it through cells with the highest uncertainty in its local neighborhood — within turning constraints. The purpose is to maximize the benefit from *search*, and the path followed is termed a *search path*. UAVs assigned to a task determine the best RAO (or RAA) with respect to their sensors (or munitions) and plan a path to approach the target from that angle.

After the initial assignment, each UAV with an $assigned$ task moves towards that task, UAVs with no $assigned$ task move towards their lowest-cost $associated$ task, while the rest follow search paths. All UAVs take sensor readings as they move and update the TOP. When a UAV 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 according to the transition function, 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.

B. Assignment Update

At all times, all $open$ and $competing$ UAVs monitor the CTL, and report their costs for all $available$ and $associated$ tasks. When a $competing$ UAV reaches a point within distance D_a of its $associated$ task, that task is $assigned$ to it and its status is switched to $committed$. All other UAVs $competing$ for this task are dis-$associated$ from it. The controller continually monitors all the costs reported, and allows as many as n_a $competing$ or $open$ UAVs to be $associated$ with it while ensuring that no UAV is $associated$ with more than m_a different tasks. Sometimes, a task may be done opportunistically by a UAV that happens to pass by, in which case all UAVs $assigned$ to or $associated$ with it are released. The process continues until the region is completely searched and all targets are neutralized, or some time limit is reached.

V. Simulation Results and Discussion

To investigate the impact of system design parameters on the performance of the proposed algorithm, Monte Carlo simulations have been conducted using an event-driven simulator. In the simulations, we consider two types of targets, which are characterized by different optimal observation and attack angles. There are two classes of UAVs: *target recognition* (*TR*) UAVs and *attack* (*A*) UAVs. All UAVs are equipped with sensors needed for search, but with different sensing capabilities. The sensing capabilities of UAVs from each class are defined using functions $\lambda_{j,k}^{C}$, the probability that a UAV of type C detects a type k target given that the true target type is j. The λ-functions are chosen to reflect the fact that target identification is dependent on the observation angle and the sensor resources used. We have used phenomenologically reasonable definitions of these functions for the purpose of simulation (e.g., giving the lowest probability of error at the optimal observation angle, and lower probability of identification error for *TR* UAVs compared to *A* UAVs). In practice, the λ-functions will be derived from a knowledge of the targets, sensors, and autonomous target recognition (ATR) systems. For the case of attack, the same considerations apply to the β_k^C functions, the probability that a UAV with type C destroy a type k target.

The first simulation (Fig. 2 and 3) considers the impact of the certainty coefficient ω_χ in Equation 6 on the performance of the UAV team. As ω_χ increases, the uncertainty value used to guide search is weighted more strongly towards uncertainty about the existence of targets, and less towards uncertainty about target type. Operationally, this means that, at high values of ω_χ, searching for targets in new locations will be preferred to spending effort verifying the type of already discovered targets. This, in turn, is expected to affect overall performance, since targets differ in optimal observation angles according to type, and targets of uncertain type are likelier to be observed (and later attacked) from the wrong direction, possibly by the wrong type of UAV. Since ω_χ is primarily a driver for search, the simulations here focus only on the search and identification tasks using only *TR* UAVs. It should be noted, however, that in a full mission, efficiency of search would also affect the time to neutralize targets. The UAV team comprises 10 *TR* UAVs in a $150\ km$ by $150\ km$ environment with 10 initially unknown targets (5 of each type). The mission is to do a complete search and to classify all the targets. The locations of UAVs and targets are generated randomly in each run. Each data point is averaged over 200 independent runs. We do two separate sets of simulations, one where UAVs have a high sensing capability (the probability of correct classification is 0.98 from optimal observation angles and 0.7 from other angles) and another where it is lower (the probability of correct classification is 0.9 from optimal observation angles and 0.6 from other angles). Performance is measured using two quantities:

- Total Mission Time (TMT): the time taken to classify all targets and bring the uncertainty of each cell within the mission region down to a pre-defined threshold (complete search). It is a subjective measure based on the viewpoint of the UAV team, and the final cognitive map and target classification might not be correct.
- Classification Error Rate (CER): the fraction of targets missed or classified incorrectly. This is an objective measure of search quality, addressing the possibility that the UAVs' assessment may differ from reality.

Fig. 2 shows that the total time for complete search has a shallow U-shaped dependence on ω_χ. The value of the optimal uncertainty coefficient appears to depend slightly on sensing capability, but is close to $\omega_\chi = 0.5$, indicating that equal weight to target presence and type is a good policy. As expected, UAVs with higher sensing capability finish faster.

Fig. 3 shows that the classification error rate is virtually independent of ω_χ, though, as expected, UAVs with lower sensing capability make more errors. Thus, the overall conclusion is that, given the sensing capability of the UAV team, ω_χ can be chosen to optimize TMT without concern for its effect on classification error.

The second simulation experiment (Fig. 4) studies how

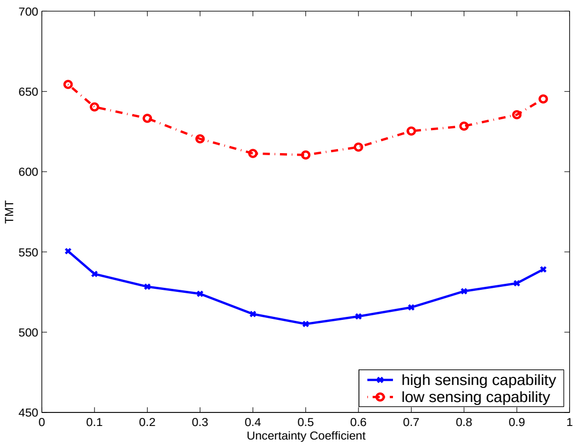

Fig. 2. Uncertainty coefficient effect on TMT. $150km$ by $150km$ environment, 10 suspected targets, 10 *TR* UAVs.

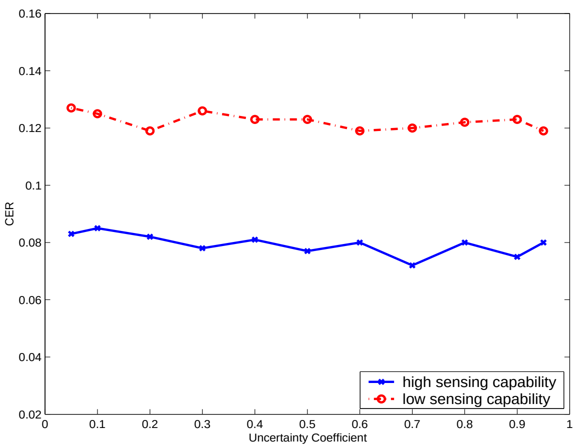

Fig. 3. Uncertainty coefficient effect on CER. $150km$ by $150km$ environment, 10 suspected targets, 10 *TR* UAVs.

the cost coefficient, ω_c in Equation 7 effects the performance of the UAV team. In contrast with ω_χ, this parameter affects mainly the assignable tasks *confirm*, *attack* and *BDA*. A larger ω_c means that, in assigning UAVs to tasks, greater weight is given to the target distance, d_{ik}, and less to the expertise of the UAVs in the pool relative to the task. Thus, a nearby UAV with lower expertise is likelier to be assigned in this case, whereas in the low ω_c situation, a better UAV would be summoned from far away. This creates an interesting trade-off between speed and quality: are more, quicker, lower-quality attempts better than fewer, delayed, higher-quality ones?

This simulation uses a UAV team comprised by 5 *TR* UAVs and 5 *A* UAVs in a 150 km by 150 km environment with 10 initially suspected targets. The goal is to confirm and verifiably neutralize all the targets. In order to elucidate the effects of distance and expertise as well as investigate the overall performance, the following performance measures are tracked:

- Target Neutralization Time (TNT): time to neutralize all targets. Again, this is a subjective measure from the viewpoint of UAVs. It is possible that some targets are still alive although UAVs think all the targets have been neutralized.
- Task-driven Traveling Distance (TTD): the UAVs' total traveling distance while associated/assigned to assignable tasks. The traveling distance involved in *search* is not counted, so that the effect of assignment policy can be unmasked. Otherwise, since all UAVs travel at constant speed, each covers the same distance during a mission.
- Task Attempt Times (TAT): total number of attempts UAVs make at assignable tasks during the mission. This is intended to evaluate whether less attention to expertise entails more attempts.
- Target Miss Ratio (TMR): the fraction of live targets after the UAVs consider the mission completed.

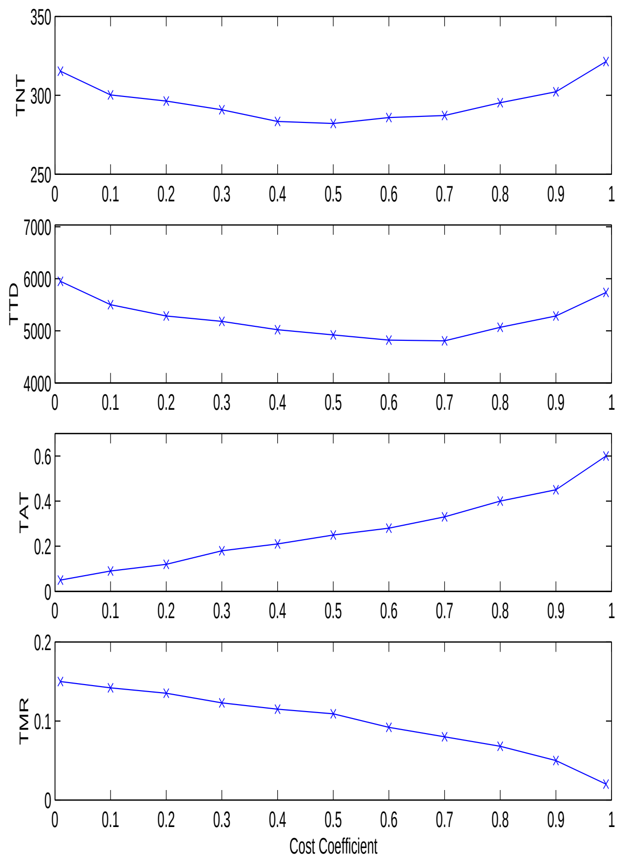

Fig. 4. Cost coefficient effect. $150km$ by $150km$ environment, 10 suspected targets, 5 *TR* and 5 *A* UAVs.

The first subplot of Fig. 4 presents the effect on target neutralization time: as ω_c increases from a small number close to zero, the TNT first decreases reaching its lowest (best) value around $\omega_c = 0.5$ (though this may vary for other mission scenarios), then increases. This clearly indicates that, in terms of mission time, paying attention to both distance and expertise is preferable to focusing on only one.

The second graph shows that the task-driven traveling distance decreases as more weight is put on the distance factor, it achieves the minimum around $\omega_c = 0.7$ (again, this may vary for other mission scenarios), and increases afterwards. This suggests that putting greater penalty of distance costs does reduce task-driven travel up to a point, but too much weight eventually begins to extract a price, presumably because tasks need more attempts due to being handled by inexpert UAVs.

This intuition is confirmed by the third subgraph, which shows that the total number of task attempts increases monotonically as ω_c increases. Clearly, expertise results in an economy of attempts, which must be balanced by considerations of distance and time.

The last graph shows that the target miss rate is relatively small and actually decreases as more weight is put on the distance cost factor. We believe that this is an ancillary consequence of the larger number of attempts (as shown in the third graph). While the attempts are by UAVs with lower expertise, their increased numbers induces a kind of "noise reduction" by averaging, ensuring that tasks considered complete are, in fact, done. Essentially, an expert, confident of success in one or two attempts, may erroneously declare a task accomplished and move on, whereas inexpert agents, less confident, make more attempts, making it possible to eliminate occasional errors in sensing and avoiding a premature declaration of success. This is the most intriguing result from these simulations, since it indicates that a low-quality (in terms of UAV's expertise) solution at the UAV-level can, in fact, have the benefit of greater success at the team-level. The implication is that a simple assignment process that pays little attention to expertise can, at a little extra cost, achieve better results than a highly optimized process that tries to match tasks and agents at great computational cost. Of course, ultimately, this will depend on the disparity between the expertise of agents and the sensing/attack qualities of the experts. If the experts are much better and always right, the high-ω_c, "nearest-first" approach is not likely to be very rewarding, but if the disparity between agents is small and even the experts have significant probability of error, the approach may represent the best option.

VI. Conclusion

The impact of the uncertainty and cost coefficients on the performance of a cooperative UAV team has been investigated. The results show that there may be an optimal choice for the uncertainty coefficient such that the total mission time is minimized, given the sensing capabilities of the UAVs. The results for the cost coefficient indicate a more complex set of trade-offs, where the choice of the best value would depend on the capabilities of UAVs and the goals of the mission. More extensive analysis of these dependencies will be reported in the future.

References

[1] R. Beard, T. McLain, M. Goodrich, and E. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Trans. Robot. Automat.*, vol. 18, pp. 911–922, 2002.

[2] P. Chandler, M. Pachter, D. Swaroop, J. Fowler, J. Howlett, S. Rasmussen, C. Schumacher, and K. Nygard, "Complexity in UAV cooperative control," in *Proc. of American Control Conference*, 2002, pp. 1831–1836.

[3] S. Kanchanavally, R. Ordonez, and J. Layne, "Mobile target tracking by networked uninhatited autonomous vehecles via hospitability maps," in *Proc. of the 2004 American Control Conference*, June 30 - Jyly 2 2004, pp. 5570–5575.

[4] J. Finke, K. Passino, and A. Sparks, "Cooperative control via task load balancing for networked uninhabited autonomous vehicles," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Dec 2003, pp. 31–36.

[5] J.-C. Latombe, *Robot Motion Planning*, ser. the Kluwer International Series in Engineering and Computer Science. Boston: Kluwer Academic Publishers, Aug 1991, vol. 124.

[6] V. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Trans. on Robotics and Automation*, vol. 6, pp. 462–472, Aug 1990.

[7] S. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Transactions on Systems, Man and Cybernetics (Part B)*, vol. 34, pp. 718–724, Feb 2004.

[8] J. Marin, R. Radtke, D. Innis, D. Barr, and A. Schultz, "Using a genetic algorithm to develop rules to guide unmanned aerial vehicles," in *1999 IEEE International Conference on Systems, Man, and Cybernetics*, Oct 1999, pp. 1055–1060.

[9] Y. Yang, A. Minai, and M. Polycarpou, "Analysis of opportunistic method for cooperative search by mobile agents," in *Proc. of the 41st IEEE Conference on Decision and Control*, Dec 2002, pp. 576–577.

[10] C. Zhang and R. Ordonez, "Decentralized adaptive coordination and control of uninhabited autonomous vehicles via surrogate optimization," in *Proc. of the 2003 American Control Conference*, Jun 2003, pp. 2205–2210.

[11] M. Flint, E. Fernandez-Gaucherand, and M. Polycarpou, "Cooperative control for UAV's searching risky environments for targets," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Dec 2003, pp. 3567–3572.

[12] L. Stone, *Theory of Optimal Search*. New York: Academic Press, 1975.

[13] M. Baum and K. Passino, "A search-theoretic approach to cooperative control for uninhabited air vehicles," in *Proc. of the 2002 AIAA Conference on Guidance, Navigation, and Control*, 2002.

[14] J. Hespanha, M. Prandini, and S. Sastry, "Probabilistic pursuit-evasion games: a one-step Nash approach," in *Proc. of the 39th IEEE Conference on Decision and Control*, Dec 2000, pp. 2272–2277.

[15] O. Laporte, "The vehicle routing problem: an overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, pp. 345–358, 1992.

[16] A. Richards, J. Bellingham, M. Tillerson, and J. How, "Co-ordination and control of multiple UAVs," in *Proc. of 2002 AIAA Conference on Guidance, Navigation, and Control*, no. AIAA-2002-4288, Aug 2002.

[17] K. O'Rourke, T. Bailey, R. Hill, and W. Carlton, "Dynamic routing of unmanned aerial vehicles using reactive tabu search," in *Proc. of the 67th MORS Symposium*, Nov 1999.

[18] A. Pongpunwattana, R. Rysdyk, and J. Vagners, "Market-based co-evolution planning for multiple autonomous vehicles," in *Proc. of 2003 AIAA Conference on Guidance, Navigation, and Control*, no. AIAA-2003-6524, May 2003.

[19] A. Gil, K. Passino, and A. Sparks, "Cooperative scheduling of tasks for networked uninhabited autonomous vehicles," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Dec 2003, pp. 522–527.

[20] Y. Jin, A. Minai, and M. Polycarpou, "Cooperative real-time search and task allocation in UAVs teams," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Dec 2003, pp. 7–12.

[21] Y. Jin, Y. Liao, M. Polycarpou, and A. Minai, "Balancing search and target response in cooperative UAV teams," in *Proc. of the 43rd IEEE Conference on Decision and Control*, Dec 2004, pp. 2923–2928.

[22] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *Artificial Intelligence Magazine*, vol. 9, pp. 61–74, 1988.