Chapter 1

COOPERATIVE REAL-TIME TASK ALLOCATION AMONG GROUPS OF UAVS *

Yan Jin
Marios M. Polycarpou
Ali A. Minai
Department of Electrical & Computer Engineering and Computer Science
University of Cincinnati
Cincinnati, OH 45221

Abstract Uninhabited autonomous vehicles(UAVs) are an increasingly important part of battlefield environments, and may soon be common in civilian applications such as disaster relief, environmental monitoring and planetary exploration. Such vehicles may be airborne, land-based or aquatic, though the focus so far has been on airborne vehicles for military applications, and this is the focus of the research presented here. We consider a heterogeneous group of UAVs drawn from several distinct classes and engaged in a search and destroy mission over an extended battlefield. During the mission, the UAVs perform *Search, Confirm, Attack* and*Battle Damage Assessment (BDA)* tasks at various locations. The tasks are determined in real-time by the actions of all UAVs and their consequences (e.g., sensor readings), so that the task dynamics are stochastic. The tasks must, therefore, be allocated to UAVs in real-time as they arise, while ensuring that appropriate vehicles are assigned to each task. Each class of UAVs has its own sensing and attack capabilities, so the need for appropriate assignment is paramount.

We present a simple cooperative approach to this problem, based on distributed assignment mediated by centralized mission status information. We also suggest methods for decentralizing the allocation process using a minimum disturbance approach termed MiDAS (Minimum Disturbance Allocation Strategy).

*This work was supported by the AFRL/VA and AFOSR Collaborative Center of Control Science (Grant F33615-01-2-3154).

1. Introduction

Over the last decade, unmanned vehicles such as airborne drones or minesweeping robots have become an increasingly feasible part of the battlefield environment, and may soon be common in civilian applications such as disaster relief, environmental monitoring and planetary exploration. However, the unmanned vehicles in current use, such as the Predator, are not autonomous, and require remote guidance by a team of human operators. Not only is this expensive and risky, it also places a fundamental limit on the scalability and range of such systems. Motivated by recent advances in intelligent systems and cooperative control, many researchers have been studying large groups of truly autonomous vehicles — called *unmanned autonomous vehicles* or *UAVs* — acting in concert to accomplish difficult missions in dynamic, poorly known and/or hazardous environments (Passino, 2002; Chandler and Pachter, 1998; Chandler et al., 2000; Chandler and Pachter, 2001; Chandler et al., 2001; Chandler, 2002; Beard et al., 2000; Beard et al., ; Bellingham et al., 2001; Jacques, 1998; Li et al., 2002; McLain and Beard, 2000; McLain et al., 2002; Moitra et al., 2001; Polycarpou et al., 2001; Polycarpou et al., 2002; Schumacher et al., 2001). The work reported here presents an approach to this problem.

We consider a heterogeneous group of UAVs drawn from several distinct classes and engaged in a search and destroy mission over an extended battlefield with known and unknown targets. The UAVs must cooperatively search the environment, confirm known targets, discover and confirm new ones, attack them with appropriate munitions, and confirm their destruction. This defines a time-varying set of tasks over the environment that must be accomplished by UAVs of the appropriate classes such that the overall mission completion time is minimized. Since the tasks are determined by the actions of the UAVs and their stochastic consequences, the process of task generation is stochastic and not predictable ahead of time. This, in turn, requires that the UAVs select tasks in the battlefield as they arise. The primary determinants of this selection are: 1) The locations of tasks and UAVs; and 2) the UAV capabilities required for each task. This creates a problem similar to many other assignment problems, but most closely related to the dynamic vehicle routing problem (Schouwenaars et al., 2001; Tan et al., 2002), albeit of much greater complexity given the several types of vehicles and tasks. Our goal, ultimately, is to develop a truly decentralized approach to the efficient heuristic solution of this problem. However, in this paper, we present a simple, somewhat preliminary model where UAVs choose tasks autonomously in a decentralized way but use a central cognitive map that provides an instantaneous and accurate summary of the current situation as known to the UAV team. Future work will address the problem of decentralizing the cognitive map and the consequences of this decentralization.

The approach we follow is motivated by the seminal work of Chandler and Pachter, and their collaborators (Chandler and Pachter, 1998; Chandler et al., 2000; Chandler and Pachter, 2001; Chandler et al., 2001; Chandler, 2002; Schumacher et al., 2001). It is also related closely to recent work by several other researchers (Beard et al., 2000; Beard et al., ; Bellingham et al., 2001; Jacques, 1998; Li et al., 2002; McLain and Beard, 2000; McLain et al., 2002). A comprehensive overview of the research problems associated with UAV teams is available in (Passino, 2002).

1.1. Scenario

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. Though we require that the UAVs completely search the entire environment, we do not include any hidden targets in the simulations reported here.

A series of *tasks* need to be accomplished at each target location as described below. A canonical *task set*, TT, is defined to comprise all the tasks that the UAVs can undertake at a target location. In this formulation, we have:

$$TT = \{Search, Confirm, Attack, BattleDamageAssess(BDA), Ignore\}$$

1.2. UAV Model

Every UAV, u_i, is characterized by an *expertise vector*, that gives information on the UAV's capabilities with regard to the tasks in the task set, TT. The *expertise vector* for UAV u_i is $\xi_i = \{\xi_{ij}\}, j = 1, \ldots, n, 0 \leq \xi_{ij} \leq 1$, where ξ_{ij} indicates the UAV's expertise for task T_j. The expertise reflects the quality of UAV's capabilities. This formulation can be used to specify classes of UAVs with specific functional repertoires (e.g., reconnaissance UAVs and attack UAVs), but UAVs can have individual expertise profiles in the general case. These profiles may also change with experience — representing learning. 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.

1.3. System Dynamics

At any time, t, every cell, (x, y), in the environment has an associated *task status*, $T(x, y, t)$, indicating what a UAV needs to do 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 mediated 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.

UAVs move in the environment seeking to accomplish the tasks cued at the cells they occupy. In the current (centralized) model, it is assumed that all UAVs know the current task state, and actively plan their paths to perform tasks suited to their capabilities, bidding for tasks and making commitments in concert with the team. Actions and observations by UAVs change the TOP, and the task status of each cell is updated based on TOP thresholds as described in the next section.

The $confirm$, *attack* and BDA tasks form the set of *assignable tasks*, i.e., tasks for which the UAVs are assigned explicitly. These UAVs move purposively to the locations of their assigned tasks and perform them. *search* and *ignore* comprise the set of *automatic tasks*, i.e., any UAV passing through a cell with one of these task statuses automatically performs the indicated task — possibly with varying quality among different UAVs for *search*. However, UAVs do not actively bid for these tasks. The *search* task does have an effect on UAV movements as described later.

All locations whose task status at time t corresponds to an assignable task 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 $\{$*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, possibly to the same task.

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

- A *physical state*, which includes information on its position, $\lambda_i(t)$, and orientation, $\delta_i(t)$.

- A *functional state*, which indicates 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 on values from the set:

$$\{open,\ competing,\ committed\}$$

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 a NULL functional state.

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 at time t. The actions are drawn from a canonical list, and include such acts as taking various kinds of sensor readings and firing various types of munitions. Doing nothing is also a possible action, and is termed the *null action*. The action performed by the UAV in cell $(x_i)t), y_i)t))$ is selected from this canonical list by an *action selection function*, G

$$a_i(t) = G(T(x_i(t), y_i(t), t), S_i(t)) \tag{1}$$

Thus, the selected action is based on the current task status of the cell and the UAV's own state in terms of capabilities and commitment. If the action is a sensor reading, it returns an *observation value*, $b_i(t)$, which is a stochastic quantity.

While actions are performed by UAVs, we also denote by $a(x, y, t)$ the set of actions (including null 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.

$$a(x, y, t) = \{a_i(t) \forall i \ s.t.\ \lambda_i(t) = (x, y)\} \tag{2}$$

$$b(x, y, t) = \{b_i(t) \forall i \ s.t.\ \lambda_i(t) = (x, y)\} \tag{3}$$

This determines the updates of the TOP value at (x, y) through a possibly stochastic *TOP update function*, F:

$$P(x, y, t + 1) = F(P(x, y, t), T(x, y, t), a(x, y, t), b(x, y, t)) \tag{4}$$

If $a(x, y, t)$ and $b(x, y, t)$ have several elements (because the cell was occupied by more than one UAV at time t), the TOP update iterates over them. The TOP value, in turn, determines the dynamics of the cell's task status, which is updated via a deterministic automaton whose transitions depend on threshold crossings in $P(x, y, t)$ (Figure 1):

$$T(x, y, t + 1) = H(T(x, y, t), P(x, y, t + 1); \theta) \tag{5}$$

where the parameter vector θ represents the set of threshold values used for transitions. Together, Equations (1) — (5) define the dynamics of the system. Note that the actions of the UAVs cause state transitions in the environment which, in turn, drives the actions of the UAVs. The dynamics is made stochastic by the stochasticity of $b(x, y, t)$ and the TOP update function, F (see below).

The TOP Update Function. As described above, the task status of each cell is updated based on the crossing of preset thresholds by its TOP. Thus, the TOP update function, $F(\cdot)$, is crucial for the system's dynamics. As indicated in Equation 4, the TOP update depends on a cell's current task status, and we define $F(\cdot)$ separately for each case.

1 *Task 1: Search:*

A UAV, u_i, engaged in the *search* task makes a sensor reading intended to detect targets. The resulting observation is given by $b_i(x, y, t) = 1$ if the sensor detects a target and by $b_i(x, y, t) = 0$ is it does not. The sensor is assumed to be imperfect, with a parameter α characterizing its detection accuracy:

$$\alpha = \frac{P(b_t \,|A)}{P(b_t \left| \overline{A} \right.)}$$

where A is the event that a target is actually located in the cell being scanned.

When the UAV's search sensor reports a target present in cell (x, y), i.e., $b_i(x, y, t) = 1$, the update is:

$$P(x, y, t+1) = \frac{\alpha P(x, y, t)}{1 + (\alpha - 1)P(x, y, t)} \tag{6}$$

When the UAV's search sensor reports there is no target in cell (x, y), i.e., $b_i(x, y, t) = 0$: (x, y),

$$P(x, y, t+1) = \frac{1 - P(x, y, t)}{1 + (\alpha - 1)P(t)} \tag{7}$$

These update equations are derived based on Bayesian inference (see Appendix).

A cell with the *search* status remains in this status until its TOP falls below the *resolution threshold*, p_r, or exceeds the *suspicion* *threshold*, p_s. In the former case, it transitions to *ignore* and in the latter to $confirm$.

2 *Task 2: Confirm:*

This status is invoked when the TOP, $P(x, y, t)$, for a cell (x, y) with status *search* reaches the suspicion threshold, p_s, indicating a significant possibility of a target being present there. The cuing of a $confirm$ task at cell (x, y) indicates that a UAV with the appropriate sensors should move purposively towards the cell and scan it. All cells where targets are suspected at mission inception are initialized with the $confirm$ task and given a TOP of p_s. The $confirm$ task is functionally identical to *search*, but is assignable to UAVs with the appropriate expertise whereas *search* is not.

The TOP update function is as given in Equations (6) and (7). However, the sensors used need not be the same as those used in *search*, and may have a different value of α. The cell remains in the $confirm$ status until its TOP falls below p_s (as a result of failure to confirm suspicions) or exceeds the *certainty threshold*, p_c. In the former case, the status changes back to *search*, in the latter to *attack*.

3 *Task 3: Attack:*

If $P(x, y, t)$ at cell (x, y) with status $confirm$ exceeds the certainty threshold, p_c, its status transitions to *attack*, indicating that an appropriately armed UAV should proceed to the location and attack it with the correct munition. Once this action is performed, the UAV changes the TOP for that location in accordance with an internal model (see Appendix).

The update function given by the simple model is:

$$P(x, y, t + 1) = P(x, y, t)\,(1 - P_s) \tag{8}$$

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.

A cell with status *attack* remains in this status until its TOP exceeds the *exit threshold*, p_e, which causes a transition to status *BDA*.

4 *Task 4: BDA:*

If, as a result of an attack in a cell (x, y) with an *attack* status, the cell's TOP falls below the exit threshold, $p_e \geq p_s$, the cell transitions to the BDA status. The task here is to verify that the TOP has indeed fallen below p_e.

The BDA task, like *search* and $confirm$ is purely observational, and uses the same update equations (6) and (7). 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 value p_r (resolution threshold) corresponds to a value so low as to exclude any possibility of a target. Note that setting $p_r = 0$ effectively eliminates the *ignore* state. However, *ignore* is a useful state since it allows the specification of regions that should specifically be excluded from the search. It also allows a mission termination condition to be defined concisely.

5 *Task 5: Ignore:*

As discussed above, this state applies to cells that do not even need to be scanned. This may be because they are known *a priori* to harbor no targets, because they have been scanned sufficiently to be excluded, or because known targets there have been destroyed verifiably.

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

1.4. Certainty Dynamics

As TOP estimates are updated via search, it is important also to quantify the "reliability" of these estimates. For example, if $P(x, y, t)$ for a location is close to zero after several UAVs have scanned it, one can be more certain that it has no target than if the TOP is based on an initial guess. Indeed, while the $confirm$, *attack* and BDA tasks are driven by the TOP, the search task must be driven by this confidence factor We quantify it 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 a lake may begin with $P(x, y, 0) = 0$ and $\chi(x, y, 0) = 1$). Most locations would typically begin with a certainty of zero. Each time a UAV visits (x, y) and makes an observation, the certainty changes as

$$\chi(x, y, t + 1) = \chi(x, y, t) + 0.5(1 - \chi(x, y, t)) = 0.5(1 + \chi(x, y, t)) \quad (9)$$

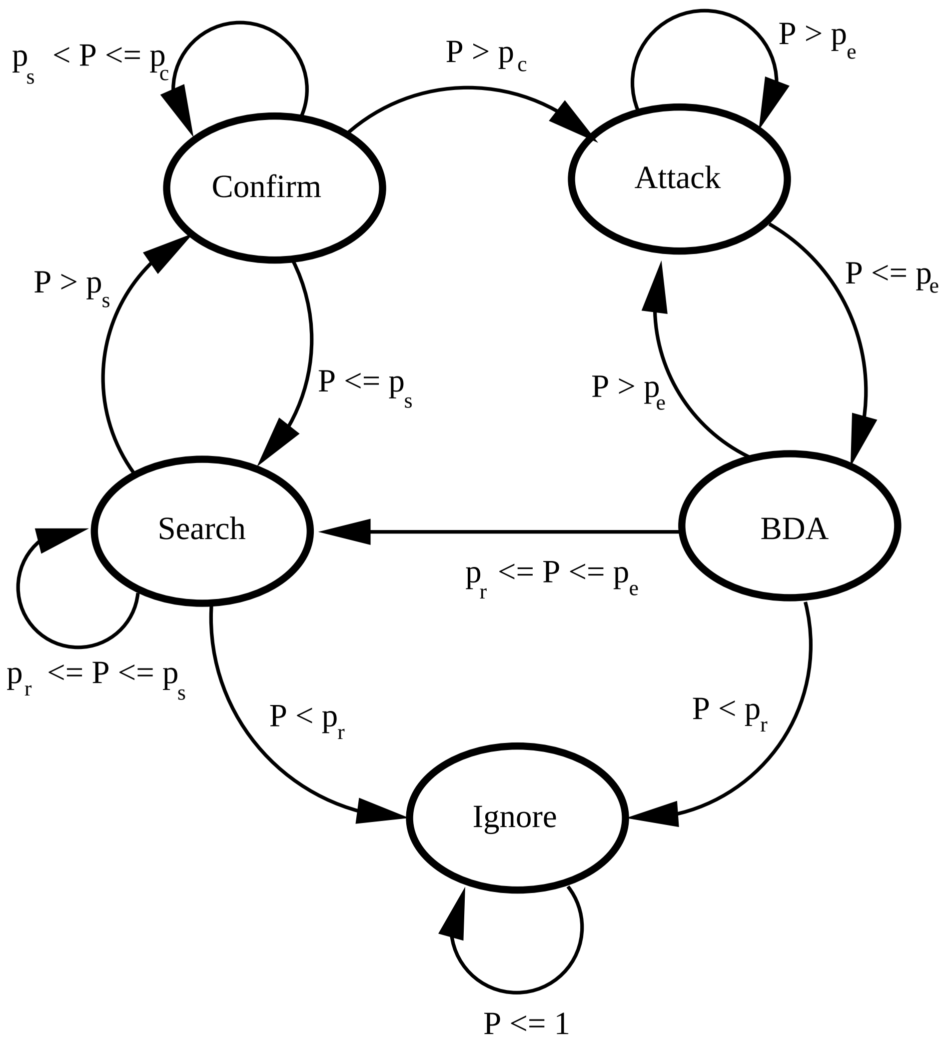

Figure 1.1. Task Dynamics. Where, Ps=suspicion threshold, Pc=certainty threshold, Pe=exit threshold, Pr=resolution threshold.

This formulation, originally proposed in (Yang et al., 2002b), 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.

2. Algorithm Description

We consider a team of UAVs, u_i, drawn from two known classes with well-defined capabilities (target recognition (TR) and attack (A)). All UAVs are assumed to have sensors needed for search. The UAV classes are represented by distinct expertise vectors, ξ_{ij}, with respect to the five tasks in the task set:

- **TR Class UAVs** have $\xi_{ij} = \{\xi_1^S, \xi_1^C, \xi_1^A, \xi_1^B, \xi_1^I\}$.
- **A Class UAVs** have $\xi_{ij} = \{\xi_2^S, \xi_2^C, \xi_2^A, \xi_2^B, \xi_2^I\}$.

All ξ_k^T are between 0 and 1. In keeping with the capability designations, we set: $\xi^S > \xi_2^S$, $\xi^C > \xi_2^C$, $\xi^A < \xi_2^A$, $\xi^B > \xi_2^B$, $\xi^I = \xi_2^I$.

The UAVs operate in a region where certain targets are suspected to exist. No threats are considered in this preliminary model. 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.

2.1. Information Base

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 $A(x, y, y)\ \forall CTL(x, y)$.

5 The UAV state vector, $S(t) = \{S_i(t)\}\ \forall u_i$.

Each UAV reads and updates the IB at each step.

2.2. Initialization

The mission begins with an externally supplied TOP map for the environment. Typically, this would include $P(x, y, 0) \leq p_s$ for regions where targets are unlikely (or impossible), $P(x, y, 0) = p_s$ for suspected target locations, and $p_s < P(x, y, 0) < p_r$ for other locations. Thus, the suspected target locations are cued with available $confirm$ tasks and the rest with *search* or *ignore* tasks. The UAVs' initial positions are also given. All UAVs initially have the open status.

2.3. Initial Assignment

Given the initial tasks and UAV positions, the first step is to assign an initial task to each UAV. We denote the current set of assignable tasks by $T_s = \{\tau_k\}$, and use j_k to denote the *identity* of task τ_k, i.e., whether it is $confirm$ ($j_k = 2$), *attack* ($j_s = 3$), or *BDA* ($j_s = 4$). The assignment is done as follows:

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

$$h_{ik} = \omega_1 * d_{ik} + (1 - \omega_1) * \exp(-\xi_{ij_k}) \tag{10}$$

where ω_1 is a positive parameter, $0 \leq \omega_1 \leq 1$, d_{ik} is the normalized distance between UAV u_i and the location of task τ_k. ξ_{ij_k} is the expertise of UAV u_i for task j_k.

2 Each UAV, u_i, chooses a task $\tau_{k_i^*}$, which satisfies $h_{ik_i^*} < h_{ik}$ for all the other τ_k's. Note that already *assigned* tasks are not considered in this process.

3 Each UAV, u_i, reports its preferred task $\tau_{k_i^*}$, the corresponding value of the distance $d_{ik_i^*}$, and the cost value $h_{ik_i^*}$ to the information base.

- If the status of the task $\tau_{k_i^*}$ is *available* (i.e., the task has not been bid for by any UAV), then u_i is considered for this task. If the distance to the task location is within a threshold ($d_{ik_i^*} \leq \theta_A$), task $\tau_{k_i^*}$ is *assigned* to UAV u_i, whose status changes to *committed*. The task is removed from the pool over which UAVs are competing. If $d_{ik_i^*} > \theta_A$, task $\tau_{k_i^*}$ is *associated* with UAV u_i, whose status is set to *competing*. The task, in this case, still remains open for competition.
- If the status of the task $\tau_{k_i^*}$ is *associated* (i.e., some other UAV, u_l, has been provisionally given this task), then the costs $h_{ik_i^*}$ and $h_{lk_i^*}$ are compared and the UAV with the smaller cost is assigned or associated with the task under the threshold rule. The UAV that loses the competition stays in the competing pool for the remaining tasks.

4 The process continues iteratively until all UAVs have been assigned a task (we assume that the number of tasks exceeds the number of UAVs).

2.4. Assignment Update

When the initial assignment is completed, each UAV begins to move towards its assigned or associated task. As it passes each cell, the TOP in that cell is updated in accordance with the update dynamics described above. When it gets to its assigned task, it performs the task and leads to a TOP update there. A new task is then cued at the CTL, and the UAV's status reverts to *open*. Depending on whether the last action caused the TOP to cross a task transition threshold, the new ask may be the same as the previous one or not. Locations that previously did not have suspected targets can become CTLs if *search* raises their TOP above p_s. This corresponds to the "discovery" of a new target. Each new assignable task — whether at an existing CTL or a new one — is cued with an *available* status.

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

2.5. UAV Movement

At all times, *open* UAVs move by following the most locally productive search direction, which is determined via the certainty variable. We use a particularly simple model in this paper, where 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 (Yang et al., 2002b; Yang et al., 2002a), we have considered more sophisticated approaches for determining efficient search paths. *Competing* and *committed* UAVs follow the most direct path to their target locations.

3. 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 measure two times to quantify performance:

1 The **target neutralization time (TNT)**, which is the time needed to neutralize all a priori known targets.

2 The **jotal mission time (TMT)**, which is the total number of steps needed to bring all cells to the *ignore* status.

4. Simulation Results

One of the primary issues considered in the simulations we report is the effectiveness of search in combination with target neutralization. As described earlier, the movement of uncommitted UAVs in the environment is driven by the need to search, mediated by the *certainty* variable. We use simulations to quantify the effectiveness of this **search-driven (SD) policy** in comparison with the naive *random movement (RM) policy*. Note that the different policies apply only to UAVs that are not associated with or committed to a task; other UAVs always take the shortest path to their designated target location.

In the first simulation (Figures 2 and 3), we consider a 15×15 cellular environment with 10 UAVS — 5 ATR units and 5 attack units. The number of targets is varied systematically from 10 to 50. The data for each case is averaged over ten independent runs with random target configurations. Figure 2 shows that the SD and RM policies lead to no significant difference in the TNT. The time taken to neutralize all known targets appears to scale linearly with the number of targets, which is to be expected. Figure 3 shows the TMT with the two policies, and here it is clear that the SD policy provides an extremely significant improvement. Thus, using the SD approach gives up nothing in attack effectiveness while greatly increasing search efficiency.

Figures 4 and 5 show the results for the case with 8 ATR UAVs and only 2 attack UAVs. While the actual mission duration is different because of the small number of attack UAVs, the results are qualitatively similar to the previous case.

5. Decentralization Approach

As described above, the current formulation is a partially centralized one in that all UAVs use the same, globally and instantaneously updated cognitive map. However, the UAVs make their commitment decisions autonomously, and this is the basis for the possibility of decentralization. We have been developing a decentralization approach that we term the **minimum disturbance allocation strategy (MiDAS)**, which has several components:

1 *Optimal, off-line initial assignment*: In this stage, UAVs are assigned to all known targets using a powerful — but possibly expensive — optimization procedure such as a genetic algorithm or integer programming. However, this is feasible because it is done off-line using powerful computers. Since the actual dynamics of the mission is stochastic and not known a priori, this initial as-

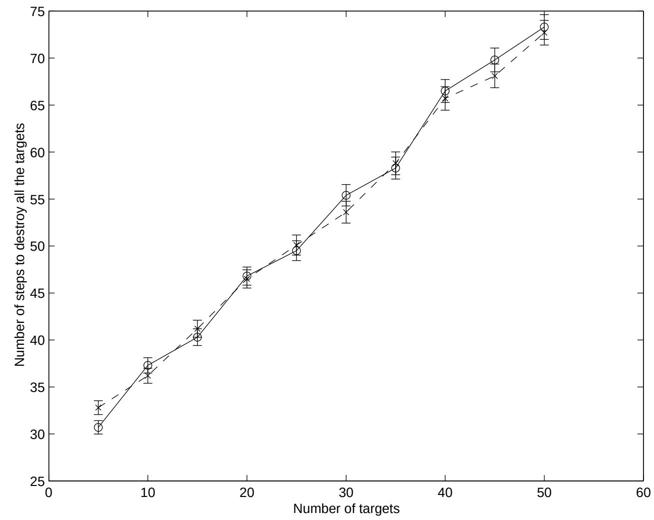

Figure 1.2. TNT for 15*15 cellular environment, 5 ATR UAVs and 5 Attack UAVs. Solid line is for the search-driven policy and dashed line for random movement.

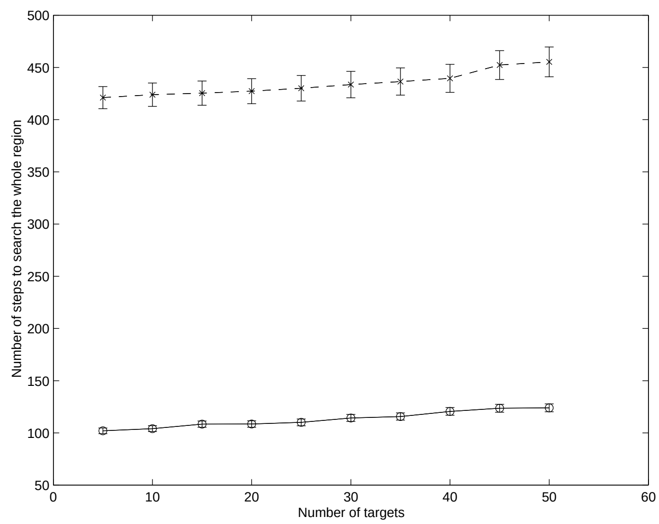

Figure 1.3. TMT for 15*15 cellular environment, 5 ATR UAVs and 5 Attack UAVs. Solid line is for the search-driven policy and dashed line for random movement.

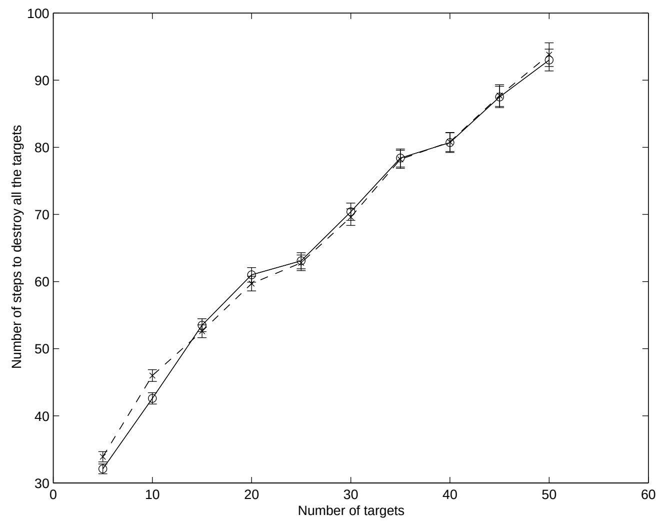

Figure 1.4. TNT for 15*15 cellular environment, 8 ATR UAVs and 2 Attack UAVs. Solid line is for the search-driven policy and dashed line for random movement.

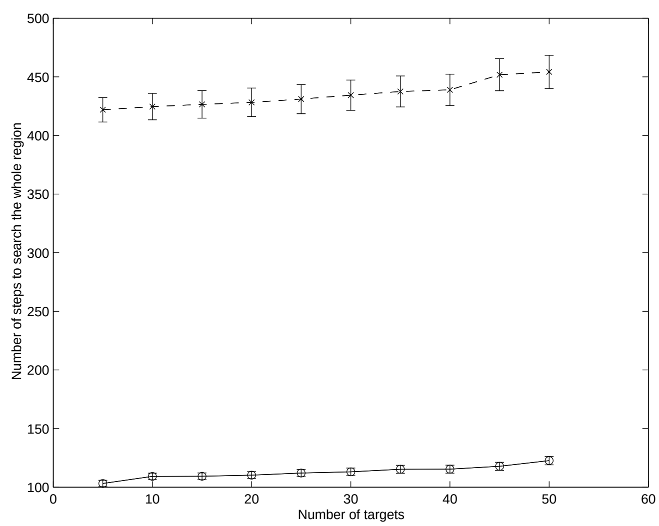

Figure 1.5. TMT for 15*15 cellular environment, 8 ATR UAVs and 2 Attack UAVs. Solid line is for the search-driven policy and dashed line for random movement.

signment would use a “typical” unfolding of the mission based on a model using known targets, etc.

2 *Decentralized, opportunistically updated cognitive map*: Rather than a single centralized map, each UAV would carry its own cognitive map built based on its own experience and on information communicated by UAVs that it happened to pass close to. The latter is what we term “opportunistic”. Clearly, any individual UAV’s map will be variously incomplete, inaccurate and out-of-date, leading to greater performance challenges compared to the centralized map.

3 *Opportunistic, decentralized in-field adjustment and assignment*: After the UAVs enter the environment and begin to follow their initial assignments, these assignments are modified locally by individual UAVs in response to developing circumstances. As the mission unfolds, it creates the actual task dynamics; new targets are discovered; new threats emerge. Each UAV, using its own limited, possibly incorrect, cognitive map, proposes changes to its plan that would not alter that plan drastically. Then, as it gets closer to its new target, it negotiates with other UAVs that may also have independently volunteered themselves for the same task. The MiDAS principle dictates that, in such a negotiation, the UAV whose assignment would lead to the least overall disruption of the initial plan is preferred. The main issue here is to develop a negotiation protocol and triggering mechanism that leads to minimal overall disruption. For example, this may involve looking at different conditions for volunteering and distance thresholds for commitment.

The system described in this paper can be seen as a preliminary version of the third component of the MiDAS approach. However, the crucial element of incremental negotiation is not yet included.

6. Conclusion and Future Work

The model presented above is only a simple, first-cut 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 initially unknown targets and pop-up threats.
- Use of more comprehensive cost functions, accounting for UAV-specific capabilities.
- Considering the existence of threats.

- Letting each UAV bid for more than one target.
- Using two or more stages of commitment for UAVs, and multiple thresholds for transition of commitment.
- Using more realistic UAV expertise profiles and target behavior.

Work on these areas will be reported in the future, as will the work on the decentralization with MiDAS.

Appendix: Derivation of TOP Update Equations

To obtain the update functions (6) and (7), consider the case where a UAV takes a measurement in cell (x, y) at time t. Define the following for a cell (x, y):

- A is the event that a target is located in cell (x, y).
- b_t is the binary sensor reading taken by the UAV, where $b_t = 1$ indicates target detection and $b_t = 0$ non-detection.
- B_{t-1} is the vector of all sensor readings for cell (x, y) by all UAVs taken up to time $t - 1$ (i.e., before time t).

Based on the above definitions, $P(A|B_{t-1})$ is the probability of target existence in cell (x, y) at time $t - 1$ and $P(A|B_{t-1}, b_t)$ is the updated probability after obtaining the new reading, b_t. Thus we have

$$P(t-1) = P(A|B_{t-1}) \tag{A.1}$$

$$P(t) = P(A|B_{t-1}, b_t) \tag{A.2}$$

We assume that the sensors' measurements in any cell are conditionally independent given the state of the cell, i.e.

$$P(b_1, b_2, ..., b_n|A) = \prod_{i=1}^{n} P(b_i|A) \tag{A.3}$$

Based on the above definitions and assumptions, the updating function (6) and (7) follow directly from Bayes' rule (Moravec, 1988). According to Bayes' rule,

$$\frac{P(A\,|B_{t-1}, b_t\,)}{P(\overline{A}\,|B_{t-1}, b_t\,)} = \frac{P(A\,|B_{t-1}\,)}{P(\overline{A}\,|B_{t-1}\,)} \cdot \frac{P(b_t\,|A\,|B_{t-1}\,)}{P(b_t\,\left|\overline{A}\,\right|B_{t-1}\,)} \tag{A.4}$$

which can be simplified by virtue of the conditional independence assumption to:

$$\frac{P(A\,|B_{t-1}, b_t\,)}{P(\overline{A}\,|B_{t-1}, b_t\,)} = \frac{P(A\,|B_{t-1}\,)}{P(\overline{A}\,|B_{t-1}\,)} \cdot \frac{P(b_t\,|A, B_{t-1}\,)}{P(b_t\,\left|\overline{A}, B_{t-1}\right.\,)} \tag{A.5}$$

$$= \frac{P(A\,|B_{t-1}\,)}{P(\overline{A}\,|B_{t-1}\,)} \cdot \frac{P(b_t\,|A\,)}{P(b_t\,\left|\overline{A}\right.\,)} \tag{A.6}$$

By solving (A.6) for $P(A|B_{t-1}, b_t)$ using the fact that $P(\bar{A}|B_{t-1}, b_t) = 1 - P(A|B_{t-1}, b_t)$, we get

$$P(A|B_{t-1}, b_t) = 1 - \left[1 + \frac{P(b_t\,|A)}{P(b_t\,\left|\overline{A}\right.)} \cdot \frac{P(A\,|B_{t-1}\,)}{P(\overline{A}\,|B_{t-1}\,)}\right]^{-1} \tag{A.7}$$

Defining $\frac{P(b_t|A\,)}{P(b_t\left|\overline{A}\right.)} = \alpha$ and using equation (A.7), (A.1), (A.2), we can obtain the update equation (6) and (7) by exchanging $P(A|B_{t-1})$, $P(A|B_{t-1}, b_t)$ with $P(t)$ and $P(t + 1)$ correspondingly.

The update function in (8) is obtained as follows:

$$
\begin{aligned}
P(x,y,t+1) &\equiv Prob(target\ present\ at\ (x,y)\ at\ step\ t+1 \mid target\ attacked) \\
&= Prob(target\ present\ at\ (x,y)\ at\ step\ t\ AND\ not\ destroyed \mid target\ attacked) \\
&\equiv Prob(target\ present\ at\ (x,y)\ at\ step\ t) Prob(target\ not\ destroyed \mid target\ attacked) \\
&= P(x,y,t)[\ 1 - Prob(target\ destroyed \mid target\ attacked)\ ] \\
&= P(x,y,t)(1-P_s) \qquad \text{(A.8)}
\end{aligned}
$$

where P_s is the probability that the target is destroyed in the attack.

References

Beard, R., McLain, T., and Goodrich, M. (2000). Coordinated target assignment and intercept for unmanned air vehicles. *Proc. ICRA'2000*, pages 2581–2586.

Beard, R., McLain, T., Goodrich, M., and Anderson, E. Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Trans. On Robotics and Automation*.

Bellingham, J., Tillerson, M., Richards, A., and How, J. (2001). Multi-task allocation and path planning for cooperative uavs. *Conference on Coordination, Control and Optimization*.

Chandler, P. and Pachter, M. (1998). Research issues in autonomous control of tactical uavs. *Proc. ACC'1998*, pages 394–398.

Chandler, P. and Pachter, M. (2001). Hierarchical control for autonomous teams. *Proc. GNC'2001*, pages 632–642.

Chandler, P., Pachter, M., and Rasmussen, S. (2001). Uav cooperative control. *Proc. ACC'2001*.

Chandler, P., Rasmussen, S., and Pachter, M. (2000). Uav cooperative path planning. *Proc. GNC'2000*, pages 1255–1265.

Chandler, P. e. a. (2002). Complexity in uav cooperative control. *Proc ACC'2002*.

Jacques, D. (1998). Search, classification and attack decisions for cooperative wide area search munitions. *Proc. Cooperative Optimization and Control Workshop*.

Li, S.-M., Boskovic, J., Seereeeram, S., Prasanth, R., Amin, R., Mehra, R., and Beard, R. a. M. T. (2002). Autonomous hierarchical control of multiple unmanned combat air vehicles (ucavs). *Proc. ACC'2002*, pages 274–279.

McLain, T. and Beard, R. (2000). Trajectory planning for coordinated rendezvous of unmanned air vehicles. *Proc. GNC'2000*, pages 1247–1254.

McLain, T., Beard, R., and Kelsey, J. (2002). Experimental demonstration of multiple robot cooperative target intercept. *Proc GNC'2002*.

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

Moravec, H. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9:61–74.

Passino, K. (2002). An introduction to research challenges in cooperative control for uninhabited autonomous vehicles. *preprint*.

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

Polycarpou, M., Yang, Y., and Passino, K. (2002). Cooperative control of distributed multi-agent systems. *IEEE Control Systems Magazine*.

Schouwenaars, T., De Moor, B., Feron, E., and How, J. (2001). Mixed integer programming for multi-vehicle path planning. *Proc. ACC'2001*.

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

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

Yang, Y., Minai, A., and Polycarpou, M. (2002a). Decentralized opportunistic learing in uav's performing cooperative search. *Proc. GNC'2002*.

Yang, Y., Polycarpou, M., and Minai, A. (2002b). Opportunistically cooperative neural learning in mobile agents. *Proc. IJCNN'2002*.