

Complexity in UAV Cooperative Control

Phillip R. Chandler
Flight Control Division
AFRL/VACA
WPAFB, OH 45433-7531

Dharba Swaroop
Dept of Mech Engr
Texas A&M University
College Station, TX 77843-3123

Jason K. Howlett
Dept of Mechanical Engineering
Brigham Young University
Provo, Utah 84602

Steven Rasmussen
Veridian, Inc
5200 Springfield Pike
Dayton, OH 45431-1289

Kendall Nygard
Dept of Computer Science and Operations Research
North Dakota State University
Fargo, ND 58105-5164

Meir Pachter
Dept of Elec & Computer Engr
AFIT/ENG
WPAFB, OH 45433-7765

Jeffrey M. Fowler
Sibley School of Mech and Aero Engr
Cornell University
Ithaca, New York 14853

Corey Schumacher
Flight Control Division
AFRL/VACA
WPAFB, OH 45433-7531

Abstract

This paper addresses complexity and coupling issues in cooperative decision and control of distributed autonomous UAV teams. In particular, the recent results obtained by the inhouse research team are presented. Hierarchical decomposition is implemented where team vehicles are allocated to subteams using set partition theory. Results are presented for single assignment and multiple assignment using network flow and auction algorithms. Simulation results are presented for wide area search munitions where complexity and coupling are incrementally addressed in the decision system, yielding radically improved team performance.

1 Introduction

In the operations research and weapon target assignment literature, fast and efficient static linear allocation algorithms are available for hundreds, even thousands of vehicles (n) and tasks (m). These are globally optimal algorithms and require that the complete cost matrix be centrally available. The auction algorithm, discussed in a later section, is a distributed form of these static linear algorithms. Sheer size is one form of complexity.

However, coupling induced complexity, and not necessarily size, dominates the wide area search munition problem: The problems addressed to date are of modest size, $n \leq 8$, but there are $m = 4$ tasks (search, classification, attack, verification) per target and an arbitrary number of targets (< 10 to date). The vehicles each have a default task of performing cooperative search, which introduces extensive coupling in their search trajectories. When an object is detected it needs to be classified. Once a target is attacked, the vehicle is destroyed. After attack, the target is viewed by another vehicle to ensure it has been destroyed. The tasks must be correctly ordered and sequenced in the shortest time due to severe fuel constraints.

To address this complexity, a hierarchical decomposition is used [1]. Figure 1 illustrates a general architecture for cooperative control and task apportionment among multiple vehicles.

Sub-teams collapse the complexity of the overall team optimization problem so that only those vehicles that have benefit in servicing the objects are considered. The partitioning uses a limited horizon minimum weight spanning tree [2]. The smaller single or multiple assignment problem can then be addressed.

Multiple assignment removes much of the myopic

penalties of single assignment. A longer planning horizon is essential for good performance in highly coupled systems. Iterative network flow [4], binary linear programming, and auction [5] are addressed in the paper as algorithms for multiple task assignment.

The paper is organized as follows: decomposition into sub-teams is addressed in §2; §3 addresses multiple assignment; §4 presents simulation results; and §5 the conclusions.

2 Sub-Teams

A set of UAVs can be clustered into a team if they service the same target, or a set of targets are close to one another. A finer assignment of which UAV should be assigned to which task may be made by considering the maneuvering constraints of the UAVs.

Suppose S_i is an ordered subset of tasks $\{\Gamma_{i_1}, \dots, \Gamma_{i_{p_i}}\}$, then we refer to S_i in the context of UAV V_j performing the tasks in S_i in the order in which they appear in S_i . The first and last tasks performed by V_j are Γ_{i_1} and $\Gamma_{i_{p_i}}$ respectively. One can associate a cost with V_j performing tasks in S_i . A feasible partition of tasks is an allocation of disjoint subset of tasks for each of the UAVs to perform, so that every task is performed by some UAV and all timing (coordination) constraints on the tasks are met. The problem of resource allocation may be posed as finding the minimum cost partition of the set of tasks, where \mathcal{P} is any feasible partition of tasks.

2.1 Graph Approach

This approach combines the ideas of iterative resource allocation with those from graph theory.

The resource allocation is performed in two stages:

- In the first stage, a classical assignment is performed to allocate resources to m_1 tasks that must be serviced/performed as soon as possible. To proceed with the classical assignment, targets requiring multiple services are replicated an appropriate number of times and treated as distinct targets that are collocated. For the purpose of replication, classification followed by attack will be considered a composite task requiring only one vehicle and will be treated as a terminal task. To avoid distinguishing between the replica and the original target, we refer to the replica by its task. The procedure is as follows:

1. Suppose there are m_1 tasks to be serviced and n UAVs with ($n \geq m$), where binary linear programming solves the classical assignment problem:

$$\min \sum_{i=1}^n \sum_{j=1}^{m_1} T_{ij} x_{ij}, \text{ where}$$

$$\sum_{i=1}^n x_{ij} = 1, \sum_{j=1}^{m_1} x_{ij} \leq 1,$$

2. There is inefficiency in this assignment, since, of the m_1 tasks, only m_2 are terminal. This implies that more than one task of the set of $m_1 - m_2$ tasks can be assigned to a UAV resulting in a faster servicing of the tasks. This leads to multiple assignment.
 3. We are concerned with multiple verification assignments for a UAV that results in smaller total service time. Timing constraints do not appear at this stage, since UAVs that are not assigned in the first stage arrive necessarily later at a target than their counterparts in the first stage.
- In the second stage, some inefficiency in resource allocation is weeded out using graph theory.

2.2 Graph Construction

Let D_i be the distance traveled by a UAV to arrive at the i th of the $m - m_1$ targets under the classical assignment.

To specify a graph, one provides the set of nodes or vertices and the set of edges/arcs connecting the nodes. The targets (or verification tasks) are nodes of the graph.

We first construct a fully connected symmetric graph in such a way that the weight of an edge/arc connecting Γ_i and Γ_j is the Euclidean distance, d_{ij} , between the nodes. We then construct a benefit graph as follows: The benefit b_{ij} is the weight of an arc/edge connecting nodes Γ_i and Γ_j and is defined to be $D_j - d_{ij}$. The benefit, b_{ij} represents the saving in distance traveled in having a UAV that visits Γ_i also visit Γ_j . Clearly, it will be beneficial to have a UAV that visits Γ_i also visit Γ_j only if $b_{ij} > 0$. Clearly, the benefit graph is asymmetric, since D_i may not necessarily be equal to D_j , although $d_{ij} = d_{ji}$.

The problem of sub-teaming can be thought of as partitioning the directed benefit graph into subgraphs so that: 1) every node is covered by one and only one

subgraph; 2) no two edges of a subgraph either start from the same node or end in the same node; and 3) the sum of the benefits of all edges in all subgraphs is maximum.

2.3 Preliminary Results

We used a minimum weight spanning tree in [2] to solve this problem. This algorithm decomposes the graph into isolated nodes and/or directed unary subgraphs where no two edges in the subgraphs share the same in-node or out-node.

The results are shown in Figure 2 which indicates how the verification tasks are partitioned and allocated to different UAVs. In the plots, the numbers in black indicate the index of the UAV and its location is given by its coordinates in the plot. The numbers in blue indicate the index of the target requiring verification and its location is specified by the coordinates in the plot. There is a line starting from a UAV and joining different targets. UAVs that are not connected to any targets are released for search operations.

3 Multiple Assignment

The three criteria to which the chosen assignment should conform are: 1) no task should be assigned to more than one UAV; 2) no task should be assigned unless any prerequisite tasks are also assigned; and 3) the estimated time at which a task is accomplished should not be before the estimated time of the immediately prerequisite task. Also, attacking a target is a terminal task. The assignment meeting all of these criteria with the minimum cost (or maximum benefit) is desired. Solutions using enumeration, BLP, relative benefit, and iterative network flow are discussed below.

3.1 Enumeration

A simplified problem may be solved with an exhaustive search. The distribution of feasible assignments and their values was calculated using a representative objective function. For each combination of two to three targets with two to four UAVs, the optimal assignment had a value about three times the mean value of feasible assignments. The distributions were qualitatively normal with a standard deviation approximately equal to the mean. As this would suggest, many feasible assignments had negative value. Figure 3 shows the distribution of values for a three-target, three-vehicle scenario, and the statistics for similar distributions with two targets. These findings demonstrate two ways in which the problem is difficult. First, the number of feasible assignments is drastically smaller than the num-

ber of possible assignments. Second, most feasible assignments have far less value than the optimum, so a random search over feasible assignments is not likely to find a near-optimal solution with a small sample size.

3.2 Binary Linear Programming

In [3] a Generalized Assignment Program framework was used to assign multiple identical UAVs to targets. A Binary Linear Programming (BLP) framework can encode all of the same information in a much smaller problem which is faster to set up and to solve. The general form of a BLP is as follows:

$$\begin{aligned} & \text{Optimize} && A \cdot \mathbf{x} \\ \text{Subject to} & (1) && x_j \in \{0, 1\} \forall j \\ & (2) && (F \cdot \mathbf{x})_i \leq d_i \forall i \\ & (3) && (F_{eq} \cdot \mathbf{x})_i = d_{eq_i} \forall i \end{aligned}$$

The optimal solution to the Binary Linear Programming problem will optimize the objective function among all feasible assignments in which no UAV is assigned more than the predetermined number of tasks. There are general solvers for BLPs, but a specialized solver is needed. The problem does not scale as efficiently as the Generalized Assignment Problem. The number of columns each scale with the product of the number of UAVs and the number of possible tours, though with a smaller coefficient. The number of constraints scales only with the number of tasks or the number of UAVs. A formulation with tours of three tasks would have more variables than a formulation with tours of two tasks, but the same number of constraints. As before, some possible tours could be heuristically eliminated in the formulation stage.

3.3 Graph Methods

An intermediate approach, between single assignment and exhaustive search, is to start with the single assignment result and look for nearby assignments with relatively better objective values. When the objective function is Euclidean distance, the marginal benefit of one UAV assuming the duties of another UAV in addition to its own has a simple form. If Target i is separated from Target j by a distance d_{ij} , and the cost associated with servicing Target j by single assignment is D_j , the benefit of the UAV assigned to Target i subsequently servicing Target j is $D_j - d_{ij}$. A matrix of benefits can be readily constructed according to

$$b_{ij} := \begin{cases} 0, & \text{if } i = j \\ D_j - d_{ij}, & \text{otherwise.} \end{cases}$$

Any of several algorithms can operate on this matrix. One method investigated uses an auction mechanism to

make the temporary assignment resulting in D_j , uses another auction to produce a permutation of targets, evaluates the cost for each UAV to service each of the resulting loops, then uses one more auction to assign UAVs to these loops. Another method uses a greedy algorithm. At each stage, Target j , is designated to be serviced by the UAV servicing Target i . The ordered pair of targets chosen at each stage maximizes b_{ij} over all ordered pairs of targets satisfying the conditions. The process is repeated until the maximum such b_{ij} is negative, or no such b_{ij} exists. The assignment by either of these algorithms could be used directly, or as a guide to partitioning targets into groups for service by sub-teams.

Good results were produced by an algorithm that does not preserve the order of targets when one UAV assumes the assignment of another, but chooses the order in a myopic manner as in Section 3.4. The ordered pair of UAVs maximizing the marginal benefit is chosen at each stage until the benefit is negative or only one UAV is assigned. This heuristic allows for general objective functions of a UAV assigned to a set of targets. Furthermore, inclusion of dummy UAVs will allow the assignment algorithm to leave some targets unserved, if the benefits do not outweigh the cost.

3.4 Iterative Network Flow

Tours can be compiled by an iteration of the linear transshipment algorithm [3]. At each stage, UAVs have a planned position and heading for the end of their assigned tour of multiple tasks. The transshipment algorithm makes a temporary assignment of these UAVs to subsequent tasks. Of these temporary assignments, the one with the earliest estimated time is fixed, and the planned UAV position and target state are updated. The process repeats until all target tasks are fully assigned. Fixing the earliest estimated time at each stage discourages, but does not prevent, non-feasible assignments. The authors also implemented the iterative algorithm as an auction, then addressed the issue of conflicting task order. One method is to associate a large cost with any potential assignment which would create a conflict. The other method is to adjust the cost of a potential assignment to include loitering, so that the dependent task occurs at the earliest admissible time. The implementation of the latter adjustment currently addresses only the order of the completion of the tasks.

4 Simulation Results

The authors have developed a MatLab Simulink based multi-UAV simulation of a wide area search munition

scenario with a hierarchical distributed decision and control system as depicted in Figure 1. The vehicles cooperatively search for and destroy high value targets. The vehicles low maneuverability and endurance are critical constraints.

The scenario, shown in Figure 4, has 8 vehicles following a preset serpentine search pattern. Figure 4 is a snapshot 98sec into the scenario where the vehicles path and footprint are color-coded. There are 3 high value rectangular targets in the search space that have an arbitrary orientation on the ground. When these potential targets pass completely through the footprint, the object is declared detected. The object cannot be attacked until it has been classified, with sufficiently high confidence. The probability of classification is a function of the aspect angle at which it is viewed. Additional views by the same, or other vehicles, may be necessary. These views are then combined statistically. More details of the cooperative classification are covered in [6].

Figure 4 uses the decision and control structure of Figure 1 where all the vehicles and objects are always assigned to one sub-team. The sub-team agent here solves a static binary assignment problem using a network flow analogy [4]. The n vehicle by m task matrix includes the costs for every surviving vehicle to perform a task that transitions each object to the next state (detect, classify, attack, verify). The vehicle agent calculates these costs by generating minimum time trajectories that satisfy kinematic constraints with specified terminal position and heading.

Figure 4 shows the typical consequences of a single step look ahead decision and control system. Following their pre-specified waypoints, vehicles 1,2 detect targets 1-3. Vehicles 5-7 are assigned to classify targets 1-3 respectively. These vehicles have the lowest cost (time) to view the objects at aspect angles that have the highest probability of classifying the objects. Once classified, the classifying vehicles are assigned to attack the target. The 3 "racetracks" are a consequence of the single assignment binary optimization and the policy of returning the vehicles to the point of search departure. The optimization is triggered only when an object changes state or a task is completed. The near circular trajectory of vehicle 1 is an example of a vehicle being reassigned and not completing a task. This "myopic" optimization uses 6 vehicles to service 3 targets.

Iteration 2 of the design process is shown in Figure 5 where the full 3 level decision hierarchy in Figure 1 is used. Here, the team agent allocates resources to an attack sub-team or a search sub-team. Initially, all the

vehicles are assigned to the search subteam. As more objects are detected, vehicles are allocated to the attack sub-team for a maximum of 4 vehicles, which is the minimum needed to attack and verify 3 high value targets. Which vehicles are assigned is based on time to target. The graph based set partition with flyable trajectories yielded vehicles 4-8 as "closest". At the sub-team level, the binary network flow assignment algorithm is sequentially applied as resources are allocated to the sub-team. Now, a maximum of 4 vehicles are considered for each task, yielding a more tractable problem. Vehicle 8 does the verification for all 3 targets.

Iteration 3 of the design uses multiple assignments to remove the trajectory inefficiency, but at an appreciable increase in complexity. Here, the generalized assignment, graph based, and the iterative network, all with tours of maximum length 3, have been used. The iterative network assignment is by far the most computationally efficient, but though not guaranteed to yield the optimal, the results have been very good. Figure 6 shows the 3 task planning horizon for the same 4 vehicle attack sub-team. Now, vehicle 8 has been assigned a 3 task verification tour. The coupling of target tasks with search tasks is also seen in this scenario.

5 Conclusions

The primary attack upon coupling-induced complexity is hierarchical decomposition. This yields a sub-optimal solution, but the optimal solution can only be found by direct enumeration, which is computationally prohibitive. The graph partition technique with flyable trajectories is promising for determining sub-team composition.

The solution of the binary linear program for the single assignment problem is the starting point for much of what has been done. Network flow solvers, auction implementations, and binary linear programming algorithms all yield the optimal solution for the binary linear single assignment problem. All are fast, but the auction mechanism is more readily implemented in a distributed fashion.

Multiple assignment extends the planning horizon and yields significant improvements in performance and robustness. The iterative network flow optimization algorithm is a short planning horizon heuristic that has been found to work well in practice. Robustness and feasibility are issues, and the algorithm could occasionally give poor performance. Task order can be enforced

in the generalized assignment problem, the graph partition, and binary linear programming, but timing is problematic. Auction algorithms may be devised as well, but share the same issues. However, all these algorithms are more computationally intensive.

References

- [1] Chandler, P. and M. Pachter, "Hierarchical Control for Autonomous Teams", AIAA GNC 2001, Montreal, Quebec, Canada, Aug 2001.
- [2] Nemhauser, G., and B. Wolsey, "Integer and Combinatorial Optimization", Wiley, 1999.
- [3] Guo, W. and K. Nygard, "Combinatorial Trading Mechanism for Task Allocation", 13th International Conference on Computer Applications in Industry and Engineering, June, 2001.
- [4] Nygard, K., P. Chandler, and M. Pachter, "Dynamic Network Optimization Models for Air Vehicle Resource Allocation", ACC 2001, Arlington, VA, June 2001.
- [5] Bertsekas, D., "Auction algorithms for network flow problems: A tutorial introduction", Computational Optimization and Applications, Vol. 1, pp. 7-66, 1992.
- [6] Chandler, P., M. Pachter, K. Nygard, and D. Swaroop, "Cooperative Control for Target Classification", *Cooperative Control and Optimization*, Kluwer, 2001.

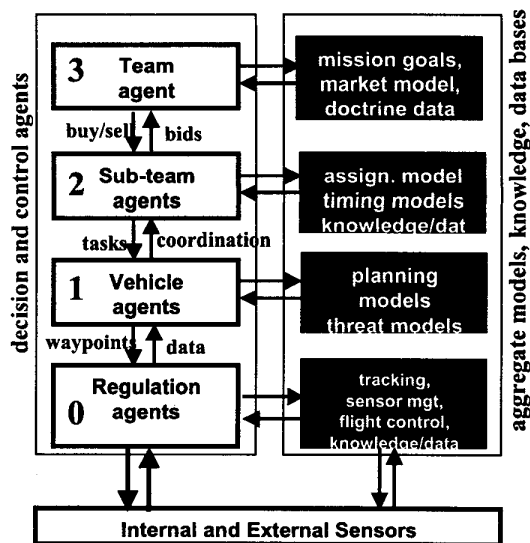


Figure 1: Hierarchical Decomposition

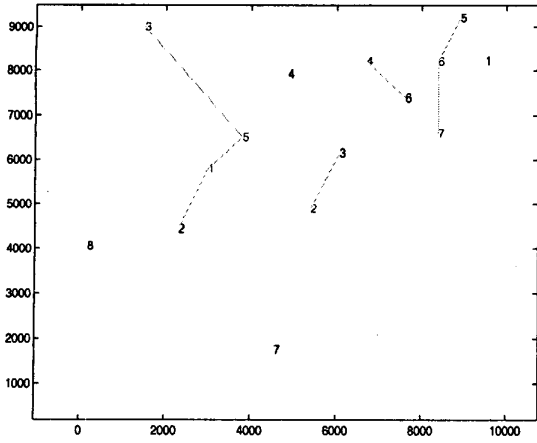


Figure 2: Illustration of team decomposition and verification assignments when UAVs and targets are interspersed

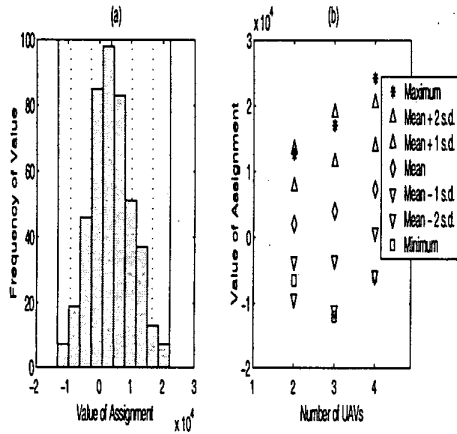


Figure 3: Distribution of values of feasible assignments for a particular scenario with three unclassified targets and three UAVs.

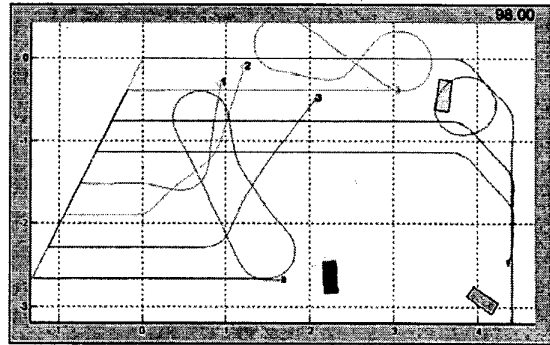


Figure 4: Decision Hierarchy with no Sub-Teams

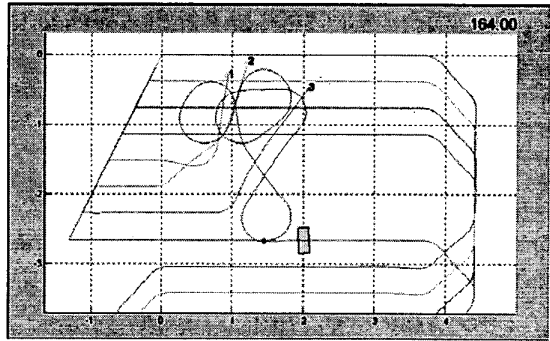


Figure 5: Decision Hierarchy with Sub-Teams

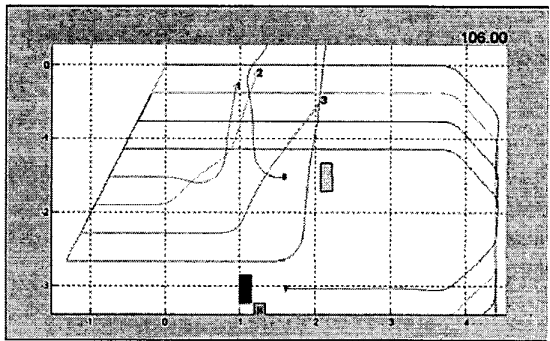


Figure 6: Decision Hierarchy with Sub-Teams and Multiple Assignment