Proceedings of the American Control Conference
Anchorage, AK May 8-10, 2002

TASK ALLOCATION FOR WIDE AREA SEARCH MUNITIONS

Corey Schumacher, Phillip R. Chandler
Flight Control Division
Air Force Research Laboratory (AFRL/VACA)
Wright-Patterson AFB, OH 45433-7531

Steven R. Rasmussen
Veridian Inc.
Wright-Patterson AFB, OH

Abstract[1]

This paper addresses the problem of task allocation for wide area search munitions. The munitions are required to search for, classify, attack, and perform battle damage assessment on potential targets. It is assumed that target field information is communicated between all elements of the swarm. A network flow optimization model is used to develop a linear program for optimal resource allocation. Periodically re-solving this optimization problem results in coordinated action by the search munitions. The network optimization model can be initialized such that multiple vehicles can be assigned to service a single target. Memory of previous task assignments is included in the task benefit calculations to reduce churning due to frequent reassignments. Simulation results are presented for a swarm of eight vehicles searching an area containing three potential targets. All targets are quickly classified, attacked, and verified as destroyed.

1.0 Introduction

Autonomous wide area search munitions (WASM) are small, powered air vehicles, each with a turbojet engine and sufficient fuel to fly for a short period of time. They are deployed in groups, or "swarms," from larger aircraft flying at higher altitudes. They are individually capable of searching for, recognizing, and attacking targets. Cooperation between munitions has the potential to greatly improve their effectiveness in many situations. The ability to communicate target information to one another will greatly improve the capability of future search munitions.

In this paper we describe a time-phased network optimization model designed to perform task allocation for a group of powered munitions each time it is run. The model is run simultaneously and independently on all munitions at discrete points in time, and assigns each vehicle a task each time it is run. The model is solved each time new information is brought into the system, typically because a new target has been discovered or an already-known target's status has been changed. A network model for task allocation was studied in [6], but the present work improves on that in [6] in two ways. One limitation of the work in [6] is that only one vehicle can be assigned to each target at a time. This is inefficient, because it does not make use of all available information. When an attack is performed, a BDA task will be needed after the attack. Knowledge of this additional task was not used, in [6], until the attack had been completed. In the present work, the network optimization model is modified so that multiple vehicles can be assigned to a single target at one time. When a target is classified, both the attack and the BDA tasks for that target are included in the next task allocation, resulting in two vehicles being assigned and the target being serviced more quickly. Another limitation of the previous work in [6] is that no memory is included of previous task assignments. This means that successive task assignment calculations could result in a searching vehicle being initially assigned to service a target, and then being reassigned back to search before completing the previous task, resulting in wasted time and fuel. The previous assignment of a vehicle is now a factor in the task benefit calculations, with a slightly increased weight on servicing the target to which the vehicle is presently assigned. A small increase in the relevant benefit has been found to greatly reduce churning, while still allowing vehicles to change assigned tasks if new information, such as a new target being found, becomes available.

The cooperative control algorithm is being implemented in a simulation with up to ten wide area search munitions and ten potential targets. This simulation has six degree-of-freedom dynamics for the search munitions and the capability to include a variety of target types. This paper presents simulation results for a swarm of vehicles searching an area containing a cluster of targets. The vehicles have limited flight times due to fuel constraints, and have an ATR capability. The vehicles are assumed to be able to communicate target state information to each

[1] This paper is declared a work of the U. S. Government and is not subject to copyright protection in the United States.

other, as well as the calculated "benefits" for each vehicle performing each possible task.

2.0 Scenario

We begin with a set of N vehicles, deployed simultaneously, each with a life span of 30 minutes. We index them i = 1, 2, ..., N. Targets that might be found by searching fall into known classes according to the value or "score" associated with destroying them. We index them with j as they are found, so that j = 1, 2, ...M and Vj is the value of target j. We assume that at the outset there is no precise information available about the number of targets and their locations. This information can only be obtained by the vehicles carrying out searches and finding potential targets using Automatic Target Recognition (ATR) methodologies. The ATR process is modeled using a system that provides a probability that the target has been correctly classified. The probability of a successful classification is based on the viewing angle of the vehicle relative to the target. At this time, the possibility of incorrect identification is not modeled, but targets are not attacked unless a 90% probability of correct identification is achieved. Further details of the ATR methodology can be found in [2], and a detailed discussion is available in [3].

3.0 Network Optimization Model

Network optimization models are typically described in terms of supplies and demands for a commodity, nodes that model transfer points, and arcs that interconnect the nodes and along which flow can take place. To model weapon system allocation, we treat the individual vehicles as discrete supplies of single units, tasks being carried out as flows on arcs through the network, and ultimate disposition of the vehicles as demands. Thus, the flows are 0 or 1. We assume that each vehicle operates independently, and makes decisions when new information is received. These decisions are determined by the solution of the network optimization model. The receipt of new target information triggers the formulation and solving of a fresh optimization problem that reflects current conditions, thus achieving feedback action. At any point in time, the database onboard each vehicle contains a *target* set, consisting of indexes, types and locations for targets that have been classified above the probability threshold. There is also a *speculative* set, consisting of indexes, types and locations for potential targets that have been detected, but are classified below the probability threshold and thus require an additional look before striking. Figure 1 provides an illustration of this model.

The model is demand driven, with the large rectangular node on the right exerting a demand-pull of N units (labeled with a supply of –N), so that each of the munition nodes on the left (with supply of +1 unit each) must flow through the network to meet the demand. In the middle layer, the top M nodes represent all of the targets that have been identified with the required minimum classification probability at this point in time and thus are ready to be attacked. An arc exists from a specific vehicle node to a target node if and only if it is a feasible vehicle/target pair. At a minimum, the feasibility requirement would mean that there is enough fuel remaining to strike the target if tasked to do so. Other feasibility conditions could also enter in, if, for example, there were differences in the onboard weapons that precluded certain vehicle/target combinations, or if the available attack angles were unsuitable. The bottom R nodes of the middle layer represent all of the potential targets that have been identified, but do not meet the minimum classification probability. We call them *speculatives*. The minimum feasibility requirement for an arc to connect a vehicle /speculative pair is sufficient fuel for the vehicle unit to assume a position in which it can deploy its sensor to assist in elevating the classification probability beyond threshold. The lower tier models alternatives for battle damage assessment for targets that have been struck. Finally, each node in the vehicle set on the left has a direct arc to the far right node labeled sink, modeling the option of continuing to search. The capacities on the arcs from the target and speculative sets are fixed at 1. Due to the integrality property, the flow values are constrained to be either 0 or 1. Each unit of flow along an arc has a "benefit" which is an expected future value. The optimal solution maximizes total value.

The network optimization model can be expressed as:

$$\max J = \sum_{i,j} c_{ij}x_{ij} \tag{1}$$

Subject to:

$$\sum_{i,j} x_{ij} + x_{jk} = 1, \qquad ,\forall i = 1,\ldots,n \tag{2}$$

$$\sum_{i} x_{is} + \sum_{j} x_{jk} = n, \qquad ,n = \#UAVs \tag{3}$$

$$x \geq 0 \tag{4}$$

$$x_{is} \leq 1 \tag{5}$$

This particular model is a capacitated transshipment problem (CTP), a special case of a linear programming problem. Constraint (2) enforces a condition that flow-in must equal flow-out for all nodes. Constraint (3) forces the number of assigned tasks to be equal to the number of available vehicles. Constraints (4) and (5) help enforce the binary nature of the problem. Any particular flow is either active or inactive (0 or 1). Restricting these capacities to a value of one on the arcs leading to the sink, along with the integrality property, induces binary values for the decision variables x_{ij}. Due to the special structure of the problem, there will always be an optimal solution that is all integer [1]. Solutions to this problem pose a small computational burden, making it feasible for implementation on the processors likely to be available on disposable wide area search munitions.

The goal of the optimization problem is to maximize the value of the tasks performed by the vehicles at the time the model is solved. Solving the model whenever new target information is available attempts to maximize the value of the targets destroyed over the life of the munitions.

Due to the integrality property, it is not normally possible to simultaneously assign two vehicles to the same target. However, creating multiple instances of the same target allow this to be done. In the following results, whenever a target is classified and thus available for attack, two instances of the target are created in the network flow model, one needing to be attacked, and one needing BDA. In this way, two vehicles can be assigned to the target, with the first reaching it performing the attack and the second performing BDA. Normally the assignment will be made such that the classifying vehicle will subsequently perform the attack, but that will not always be the case, especially if the available vehicles have different remaining flight times. One potential hazard in this approach is that the sensor footprint of the vehicle performing BDA can overfly the target before the attack is performed. This will be a rare event, and can be avoided by comparing the vehicle ETA's and modifying the BDA vehicle's flight path at necessary.

4.0 Simulation

This network flow model has been implemented in our multi-vehicle, multi-target coordinated-control simulation. The scenario has eight Wide Area Search Munitions performing a search for targets in a rectangular area. The WASM are using a simple "moving the grass" search pattern. There are up to 5 different target types possible in the simulation, including a "non-target" target type for objects that appear similar to targets but which may be distinguishable as non-targets by the ATR.

One of the critical questions involved in using the network flow model for coordinated control and decision-making for WASM is how the values of the weights *c(i,j)* are chosen. Different values will achieve good results for different situations. For example, reduced warhead effectiveness greatly increases the importance of battle damage assessment and potential repeated attacks on an individual target. A simplified scheme has been developed which does not attempt to address the full probabilistic computation of the various Expected Values suggested by (1)-(4) above. It is intended to assign the highest value possible to killing a target of the highest-valued type, with other tasks generating less of a benefit. The values of different tasks are calculated as follows:

C(i,j) = Expected value of vehicle I attacking target j
= (Probability target type has been correctly identified)*(Probability of destroying target j) * (Value of target j)*(Time weighting)*(Previous task weighting)
= $P_{id}*P_k*V_j*\min_j(\text{ETAMatrix})/\text{ETAMatrix}(i,j)*\gamma$

C(i,s) = Value of vehicle i continuing to search
= (Maximum Target Value)*(Remaining flight time)/(Maximum flight time)
= $\max(\text{target values})*T_f/T_m$

C(i,k) = Expected value of vehicle i assisting in classifying speculative k
= ((Probability successful ATR)*(Expected value of target being attacked after classification) + Value of continued search after classification) *(Previous task weighting)
= $(Patr*Pk*Vj+\max(\text{target values})*(T_f - T_{classify})/T_m)*\gamma$

C(i,g) = Expected value of vehicle i performing BDA on target g
= ((Probability successful BDA)*(Probability target was not killed)(Probability of correct target ID)(Value of target j) + Value of continued search after classification) *(Previous task weighting)
= $(P_{bda}*(1-P_k)*P_{id}*V_j + \max(\text{target values})*(T_f - T_{bda})/T_{mg})*\gamma$

There are five possible target types with different values, and different ATR characteristics. P_{id} is an input based on the quality of the ATR recognition. ETAMatrix contains the required flight times for each vehicle *i* to fly to each target *j* . T_f is the remaining available flight time of a vehicle, and T_m is the maximum flight time of the vehicle. For the following simulation results, some of the parameters were set as constants: $P_k = 0.90$, $P_{bda} = 0.90$. $T_{classify}$ and T_{bda} are equal to the flight time to reach the specified target, plus the time needed to return to search after the task is completed. The additional weighting γ is used to encourage vehicles to continue on to service targets to which they have already been assigned, and thus reduce the "churning" effect which can occur if vehicle-target assignments change frequently. We have found that γ=1.05 greatly reduces the churning effect, while still allowing changes in task assignments when new information, such as a newly-discovered target, is available.

The value of attacking a target is weighted with the time required for a vehicle to perform that attack, so that a higher value is assigned to a vehicle that can attack a target sooner. The value of continuing to search is set such that the value of searching is equal to the value of killing a high-value target initially, and degrades linearly with search time remaining. This will tend to result in vehicles with less flight time remaining being used to kill targets, and vehicles with more fuel left being used to search, classify, and perform BDA. Determining precise appropriate values for the probabilities of successful ATR and BDA is difficult, and requires substantial modeling of those processes, which this paper does not address in substantial detail. Simplified models giving reasonable values for these parameters are used. The value of all possible tasks, vehicle, and target assignment combinations are calculated and sent to the capacitated transshipment problem solver. The values are multiplied by 10,000 before being sent to the solver, as it only works with integers and rounding will result in poor results without the scaling factor.

For the simulation results presented, eight vehicles are searching an area containing five targets of different types, and hence of different values. The target information is as follows:

Target	Type	Value	Location (X,Y)
1	1	10	(9500,-500)
2	1	10	(15000,-2500)
3	2	7	(15000,-14000)

The targets also have an orientation (facing) that has an impact on the ATR process and desired viewing angles, but this will not be discussed as it does not directly affect the task allocation. The search vehicles are initialized in a staggered row formation, with fifteen minutes of flight time remaining, out of a maximum thirty minutes. This assumes that the vehicles have been searching for fifteen minutes and then find a cluster of potential targets.

As vehicles are assigned non-search tasks, the possibility arises of failing to locate targets, but that does not occur in this instance. We do not attempt to compensate for that possibility in this paper. Search issues are complex in and of themselves, and beyond the scope of this paper. Figure 2 shows simulation results with $\gamma = 1$ (no memory weighting). The colored rectangles represent the sensor footprints of the searching vehicles, and the numbers are the target locations. Colored lines show flight paths. Targets are numbered 1-5. As soon as each target is classified, one vehicle is assigned to attack it, and another is assigned to perform battle damage assessment on that target. Since the task allocation algorithm is performed each time a task is completed, the assignments are recalculated immediately after a target is struck. There are three instances where a vehicle is pulled off its search path to perform an attack task, and then reassigned to search before completing that task. This "churning" occurs due to small variations in the length of the path that is calculated for each iteration of the task allocation, and results in wasted vehicle fuel and potentially more gaps in the search pattern. All of the targets are still fully serviced (found, classified, attacked, and BDA'd) in this example.

Simulation results with $\gamma = 1.05$ (a small "memory" weighting) are shown in Figure 3. In this case, the small additional weight on servicing a target to which a vehicle is already assigned results in reduced wasted effort. Each time a vehicle is assigned to service a target it maintains that assignment during later assignment calculations. This could change due to new information, such as a new target being found, but the algorithm is no longer sensitive to minor variations in task values due to changes in the calculated path lengths. Combining both the use of multiple instances of a target in the task allocation computation and the memory weighting allows the immediate use of all available information about the targets and tasks to be performed. Monte Carlo runs with established performance metrics would be required to carefully evaluate the advantages of using this memory factor, but the initial results are promising.

5.0 Conclusions

In this paper we presented a solution to the problem of task allocation for wide area search munitions. The vehicles are capable of searching for targets, performing ATR to classify targets, attack targets, and perform BDA on targets. A linear program based on the capacitated trans-shipment problem is used to solve the task allocation problem. Simulation results are presented for eight vehicles searching and attacking three targets of different values within the search area. The network optimization results in an optimal allocation of vehicle resources to the required tasks. Multiple vehicles are simultaneously assigned to a single target, resulting in faster completion of BDA tasks after an attack. A memory factor is included in the task benefit calculations to reduce churning due to frequent modification of task assignments. Further work is needed in this area, to refine the methods for computing the relative benefits of each task. The method is still limited, in that each vehicle can only be assigned one task at a time. Nonlinear or iterative methods, which will not have this limitation, need to be investigated. Metrics are also needed to allow more precise evaluation of competing techniques.

6.0 References

1. Nygard, K. E., Chandler, P R. , Pachter, M.., "Dynamic Network Flow Optimization Models for Air Vehicle Resource Allocation," Proceedings of the Automatic Control Conference, June 2001.
2. Chandler, Phillip R., Pachter, Meir, ``UAV Cooperative Classification", to appear in Workshop on Cooperative Control and Optimization, Klewer Academic Publishers, 2001.
3. Chandler, Phillip R., Pachter, Meir, "Hierarchical Control of Autonomous Teams," Proceedings of the 2001 AIAA Guidance, Navigation, and Control Conference
4. Ford, L. R Jr. and D. R. Fulkerson, "Flows in Networks," Princeton University Press, Princeton, NJ, 1962

5. Murphy, R.A., "An Approximate Algorithm For a Weapon Target Assignment Stochastic Program," Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems, editor: P.M. Pardalso

6. Schumacher, C, Chandler, P.R, Rasmussen, S. R., "Task Allocation for Wide Area Search Munitions Via
7. Network Flow Optimization", Proceedings of the 2001 AIAA Guidance, Navigation, and Control Conference.

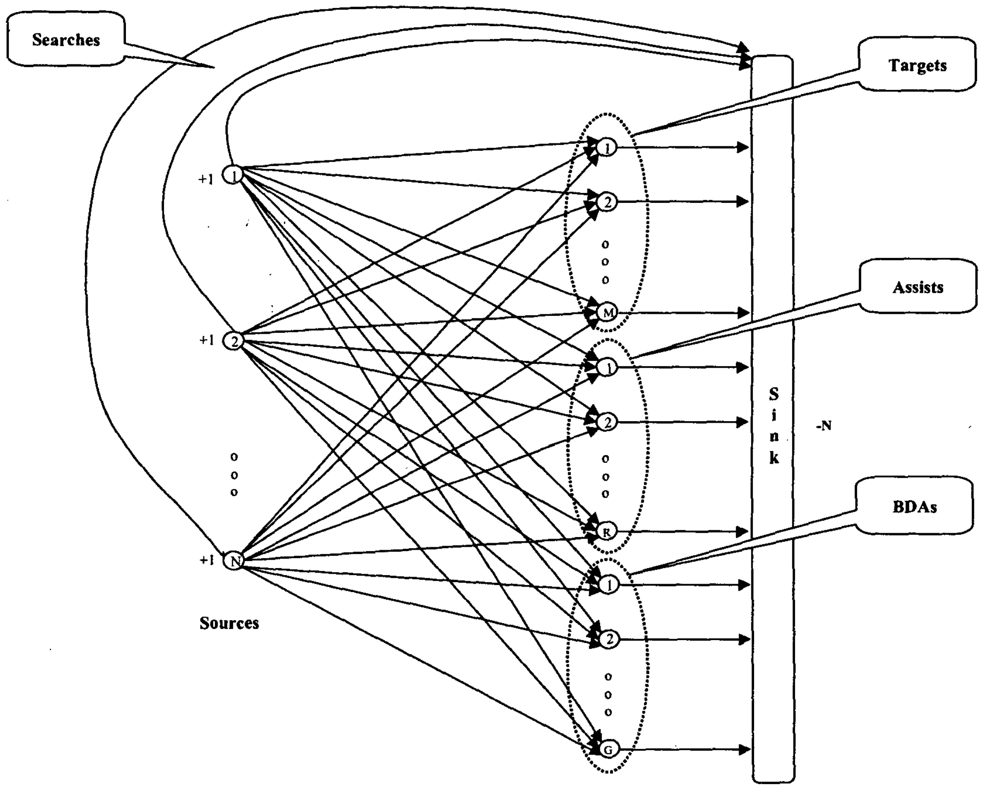

Figure 1: Network Flow Model for Task Allocation

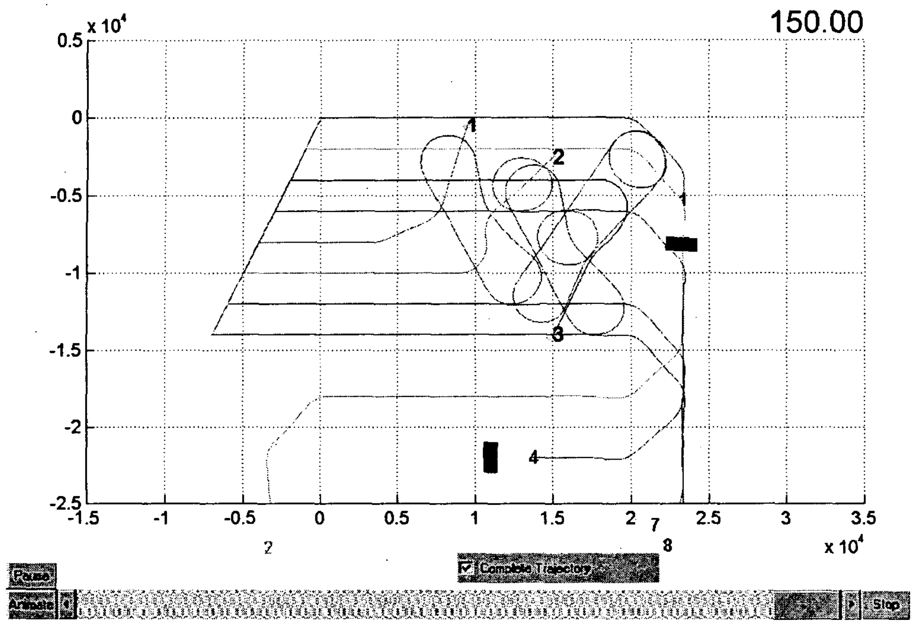

Figure 2: Vehicle Flight Paths and Target Locations without Task Memory

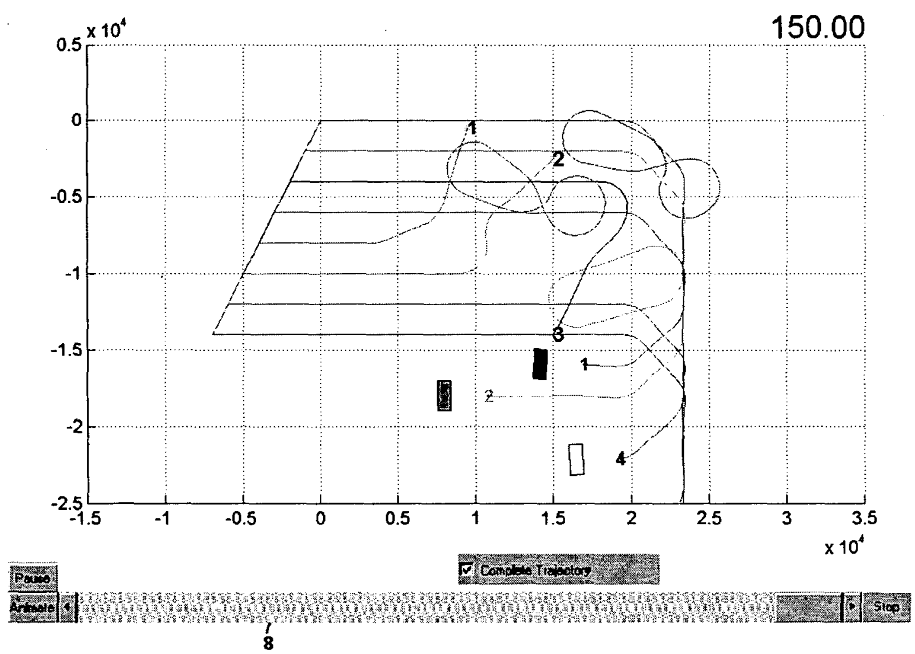

Figure 3: Vehicle Flight Paths and Target Locations with Task Memory