

State-Space Search for Improved Autonomous UAVs Assignment Algorithm

S. J. Rasmussen, T. Shima, J. W. Mitchell, A. G. Sparks, P. Chandler
Air Vehicles Directorate, Air Force Research Laboratory, Wright-Patterson AFB

Abstract—This paper describes an algorithm that generates vehicle task assignments for autonomous uninhabited air vehicles in cooperative missions. The algorithm uses a state-space best-first search of a tree that incorporates all of the constraints of the assignment problem. Using this algorithm a feasible solution is generated immediately, that monotonically improves and eventually converges to the optimal solution. Using Monte Carlo simulations the performance of the search algorithm is analyzed and compared to the desirable assignment algorithm attributes. It is shown that the proposed deterministic search method can be implemented for given run times, providing good feasible solutions.

I. INTRODUCTION

Advances in technology have made it possible to field autonomous uninhabited air vehicles (UAVs) that can be deployed in teams to accomplish important missions such as suppression of enemy air defenses and combat intelligence surveillance and reconnaissance. While it is technically possible to field these types of vehicles, work is needed to develop implementable strategies/algorithms to allow UAVs to cooperate with each other in order to perform these types of missions. Major portions of proposed missions can be preplanned, but due to limited information about enemy positions and assets in the battlefield area, the UAVs will have to react to changes in perceived enemy state during execution of the mission plan. Cooperating, the UAV team will be able to optimize the use of their combined resources to accomplish the goals of their mission. If the UAVs are unable to cooperate with each other in online planning and execution of the mission, then either group autonomy will be traded for high levels of manned intervention or more vehicles/resources will be required to perform the mission. While cooperation of this kind is desirable, it can be very complicated to implement. To perform these missions, acceptable algorithms must be solved with given time constraints and be robust to uncertainties arising from elements such as sensors, communication, and plan execution.

Steven Rasmussen is a General Dynamics on-site Contractor. steven.rasmussen@wpafb.af.mil

Tal Shima is a National Research Council (NRC) Visiting Scientist. shima_tal@yahoo.com

Jason Mitchell was a Visiting Scientist. He is Currently an Aerospace Scientist for Emergent Space Technologies, Greenbelt, MD. Jason.Mitchell@emergentspace.com,

Andrew Sparks is a Senior Aerospace Engineer. Associate Fellow IEEE. andrew.sparks@wpafb.af.mil

Phillip Chandler is a Senior Controls Engineer. philip.chandler@wpafb.af.mil

Many different candidate cooperative control algorithms have been developed, implemented, and simulated [1]–[6]; but, due to the complexity of this problem, all of these algorithms have been heuristic in nature. Many of these algorithms also do not meet all of the requirements of the assignment problem, i.e. assignment coordination, task precedence, and flyable trajectories. In order to judge the effectiveness of these algorithms a tree generation algorithm was developed [7] that produces optimal solutions to the assignment problem based on piecewise optimal trajectories. This algorithm generates a tree of feasible assignments and then by exhaustive search finds the optimal assignment. During generation of the tree all of the requirements of the mission are met, but since enumeration of all of the feasible assignments is needed, direct use of this approach is only reasonable for relatively low dimensional scenarios and off-line applications.

In this work a dynamic state-space search algorithm is proposed that has many desirable qualities such as providing a fast feasible solution that monotonically improves and, eventually, converges to the optimal solution. Using the tree to represent the decision state-space makes it possible to incorporate many different types of constraints into the solution of the problem. Since the state-space is traversed dynamically, i.e. only the states discovered in the search are instantiated, the algorithm can efficiently find feasible solutions. Given enough time the essentially branch and bound search algorithm will converge to the optimal assignment without a complete enumeration of all of the states.

The remainder of this manuscript is organized as follows: In the next section, the UAV task assignment problem is reviewed. This is followed by a description of the state space best first search algorithm for the studied problem. A Monte Carlo simulation study is then presented and concluding remarks are offered in the last section.

II. ASSIGNMENT PROBLEM

Since UAV teams and missions that require cooperative decision and control can be varied with many different requirements and capabilities, a generic assignment problem is defined. Let

$$T = \{1, 2, \dots, N_t\} \quad (1)$$

be the set of targets and let

$$V = \{1, 2, \dots, N_v\} \quad (2)$$

be a set of UAVs performing tasks on these targets. In this assignment problem the UAVs are required to perform three

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE DEC 2004		2. REPORT TYPE		3. DATES COVERED 00-00-2004 to 00-00-2004	
4. TITLE AND SUBTITLE State-Space Search for Improved Autonomous UAVs Assignment Algorithm				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Research Laboratory, Air Vehicles Directorate, Wright Patterson AFB, OH, 45433				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES The original document contains color images.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 6	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

tasks to prosecute each target, i.e. classify, attack, and verify kill of the targets, while maintaining forward air speed. Thus the set of missions is

$$M = \{Classify, Attack, Verify\} \quad (3)$$

This type of assignment problem is termed bounded speed task assignment problem (BSTAP).

A. Assignment Requirements

Each of the tasks has requirements governing its execution. Target classification is required to ensure that the subject object is the intended target and not a decoy or some other non-target. To complete a classification task a vehicle must follow a trajectory that places its sensor footprint on the target at a selected heading angle with respect to the target. After a target has been successfully classified one or more UAVs attack it with restrictions on their trajectory. Then the UAV team must verify that the target was killed using their onboard sensors with given footprints.

The tasks for each target must be accomplished in order, i.e. the target must be classified before it can be attacked and attacked before it can be verified. Thus, any algorithm that produces cooperative assignments must enforce precedence of the tasks. In order to utilize the UAVs in an efficient manner, each task must be accomplished once, i.e. UAVs are not allowed to attack a target twice, unless the target is verified alive after an attack or there is a predefined need for multiple attacks. This means that task coordination must be enforced in any optimal solution to this assignment problem. In order to guarantee that the task precedence requirements are met, the assigned trajectories must be flyable, e.g. a fixed-wing UAV has a minimum turning radius. If the trajectories assigned to a UAV are not flyable, the timing and geographical coordination of the cooperative mission may be invalidated. The scale of the scenario is important in determining the impact of the flyable trajectories constraint, e.g. if the turn radius of the vehicle is very small when compared to the distance between the targets, then this constraint may not have a negative impact on the mission.

Based on the analysis above, any optimal cooperative control algorithm for solving the BSTAP must comply with the following constraints:

- (i) *Task Coordination* - Vehicle task assignments must be coordinated to ensure that every task $k \in M$ is completed exactly once on each target $j \in T$.
- (ii) *Task Precedence* - The Tasks performed on each target must be in the following order: classify, attack and verify.
- (iii) *Flyable Trajectories* - The UAVs must be assigned trajectories that they can follow.

B. Combinatorial Optimization Problem

Even for relatively low numbers of vehicles and targets the BSTAP is a very large combinatorial problem. Table I shows the number of nodes in the decision space for various vehicle/target engagements that require three tasks. Because

Targets/Vehicles	2	3	4
2	2,229	2,0833	106,569
3	1,465,507	46,816,228	570,031,453

TABLE I
TOTAL NUMBER OF NODES IN THE DECISION STATE-SPACE.

of the size of the problem and the need to implement these algorithms on-line, desirable qualities of candidate cooperative decision and control algorithms are: fast feasible solutions, improved solution over time, and incorporation of vehicle dynamics constraints.

BSTAP analyzed in this paper, the performance metric is defined as the cumulative distance travelled by the vehicles to perform all of the required tasks

$$J = \sum_{i=1}^{N_v} r_i \quad (4)$$

where r_i is the distance travelled by UAV $i \in V$ until finishing his part in the group task plan. At that point in time the UAV has no more group tasks to fulfill and can resume a default task, e.g. searching for new targets. The group objective is to minimize Eq. 4 subject to the constraints (i)-(iii).

An optimal solution to the BSTAP, complying with all of the constraints, can be obtained using the mixed integer linear programming (MILP) method. However, for most significant problems, this algorithm can take a long time to set up and to execute. Heuristics, such as using Euclidean distances instead of the restriction of flyable trajectories, have been proposed to speed up the MILP algorithm on the expense of optimality [8], [9]. Incorporating the restriction of flyable trajectories but allowing the trajectory to be piece-wise optimal, a tree generation algorithm has been recently developed [7]. While this algorithm is easy to set up, for most significant problems it also takes a long time to exhaustively search for the optimal solution. In the next section a best first search algorithm for such a tree is proposed allowing fast feasible solutions that monotonically converge, eventually, to the piece-wise optimal solution.

III. STATE-SPACE SEARCH ALGORITHM

In [7] it was demonstrated that the BSTAP can be represented by a tree. This tree not only spans the decision space of the BSTAP, but it also incorporates the state of the problem in its nodes. The tree is constructed by generating nodes that represent the assignment of a vehicle $i \in V$ to a task $k \in M$ on a target $j \in T$ at a specific time. The child nodes are found by enumerating all of the possible assignments that can be made, based on the remaining tasks and requirements of the BSTAP. Nodes are constructed until all of the combinations of vehicles, targets, and tasks, that represent feasible assignments, have been found.

The choice of a search algorithm can greatly effect the rate at which feasible assignments are improved. To search

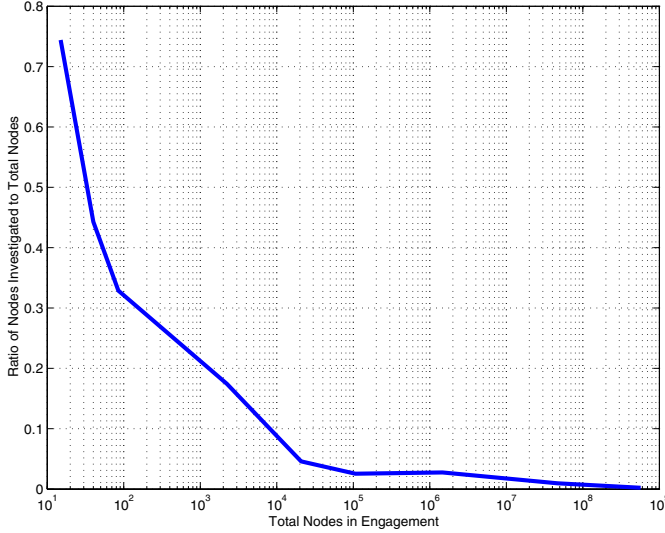


Fig. 1. Mean nodes required to compute optimal assignment.

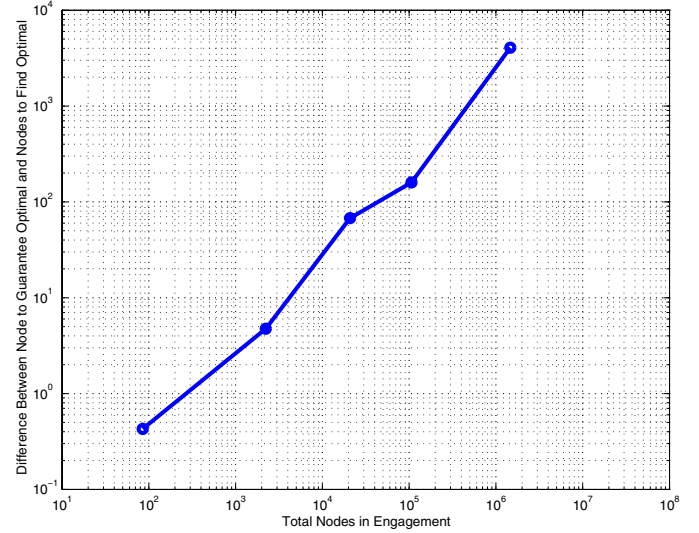


Fig. 2. Mean difference between number of nodes required to guarantee optimal and those required to find optimal assignments.

trees the choices are essentially, breadth-first, depth-first, and heuristic searches. Since the depth of the tree is very shallow compared to its width, a depth first search will generate feasible assignments quickly. Heuristic searches can also take advantage of the depth of the tree while also including known and predicted information into the conduct of the search. One such heuristic algorithm that is simple to implement is the state space best-first search (SSBFS) algorithm. With the SSBFS algorithm, the costs of the children nodes are calculated and the lowest cost (best) child node is expanded. This algorithm causes more nodes to be calculated than a depth-first search, but tends to arrive at better solutions earlier. Since the search is dynamic, only those nodes investigated need to be instantiated. This means that large portions of the tree can be trimmed based on previously discovered lower cost assignments. As shown in Figure 1, as the total number of nodes in the space increases, the ratio of nodes investigated to total nodes decreases. This makes it possible to find optimal assignments for larger dimensional problems than is possible with an exhaustive search of the nodes. Finding an optimal assignment takes less nodes than guaranteeing that the optimal assignment has been found. That is, after the optimal assignment is found, all of the uninvestigated nodes must be investigated or pruned. The difference between finding the optimal solution and guaranteeing that it is the optimal is shown in Figure 2 for the test scenarios.

The depth of the tree, i.e. the number of nodes from the root node to the leaf nodes is

$$D = N_t N_m \quad (5)$$

where N_m is the number of tasks that need be performed on each target (in the investigated problem $N_m = 3$). Traversing the tree from a root node to a leaf node produces a feasible assignment for UAVs to tasks. This makes it

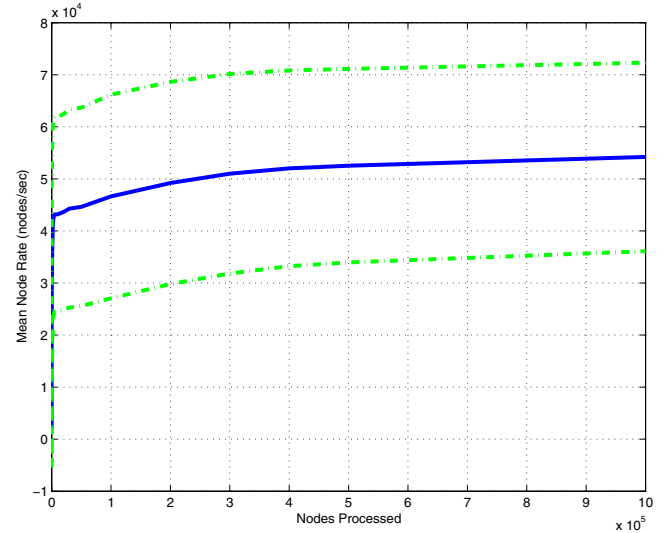


Fig. 3. Node processing rate statistics.

possible to find feasible assignments in a known time

$$t = D/n \quad (6)$$

where n is the node processing rate. Figure 3 shows the mean of a node processing rate n as a function of the number of total nodes processed. Note that although this quantity is computer platform based (a Pentium IV-2400Mhz in this case) the qualitative nature of this parameter is that it converges to a constant.

Once a feasible assignment is discovered, its cost J is saved as a candidate optimal solution. As the search progresses more nodes of the tree are evaluated and compared against the cost of the candidate optimal assignment. If the new nodes are of lower cost than the optimal candidate solution then they become the new optimal candidate solution. If the cost is higher, then the node and all its children nodes

are pruned. The search is terminated when all nodes have been investigated or pruned.

IV. RESULTS

To test the SSBFS algorithm a number of different engagements were constructed using the MultiUAV simulation [10]. For each simulation run the vehicles started at the same location and searched a given area with a given search pattern. At the beginning of each simulation run the position and heading of each target were selected using random draws from a uniform distribution. The simulation was run 100 times. Each time the assignment algorithm was needed during the simulation, a new engagement was declared. The state of the vehicles and targets was saved for every engagement. For the purpose of this test, all of the initial required tasks for the targets were set to the initial task, i.e. classify, which enabled all of the engagements to be compared with each other. The SSBFS algorithm was then executed for each engagement in the saved data. This produced sets of feasible assignments, optimal assignments, nodes required to guarantee optimal assignments and algorithm run time (in seconds and number of nodes evaluated).

The run time plots of the solution quality for a 4 vehicles, 2 targets case and a 4 vehicles, 3 targets case are shown in Figures 4 and 5, respectively. Note in these figures that the run time is enumerated in nodes at the bottom and seconds at the top of the plot. These plots represent individual engagements, but they are representatives of the results from the other engagements. As can be seen the initial solutions are found as quickly as possible, after D nodes have been processed; 6 nodes for the two targets case and 9 nodes for the three targets case. Both of the initial solutions are roughly twice the optimal solutions; both are monotonically improving; and both converge to the the optimal assignment solution. Each of the step improvements in the plots indicate that a better feasible solution to the BSTAP was found.

Figure 6 shows trajectories for feasible and optimal assignments for a 2 vehicles, 2 targets, and 3 tasks engagement. The trajectories on the left are based on the feasible solution found before 100 nodes were processed. The trajectories on the right represent the optimal assignment. In this figure, initial vehicle positions are marked with green disks and target positions are marked with red squares. The numbers in the figure mark the position of way points and the color-coded lines represent the trajectories assigned to each vehicle. For the 100 node solution $J = 73960m$ and the optimal solution is $J = 62468m$, representing a factor of 1.2, or 11474m, decrease in total distance travelled. Tables II and III show the vehicle assignments for the respective cases. The differences between the two assignments are that the classify, attack, and verify assignments for each vehicle have switched targets. Switching the targets made it possible for the vehicles to fly shorter trajectories.

To analyze the quantitative performance of the SSBFS algorithm a per vehicle capacity was assigned to each team

red	C1	A1	V2
green	C2	A2	V1

TABLE II
ASSIGNMENTS FOR THE 100 NODE ASSIGNMENT OF FIGURE 6
(C-CLASSIFY, A-ATTACK, V-VERIFY).

red	C2	A2	V1
green	C1	A1	V2

TABLE III
ASSIGNMENTS FOR THE OPTIMAL ASSIGNMENT OF FIGURE 6.

of UAVs making possible to calculate a team capacity

$$J_R = \sum_{i=1}^{N_v} R_i \quad (7)$$

where R_i is the per vehicle capacity, e.g for this BSTAP it is the total distance a vehicle $i \in V$ can perform assignments. This variable is assumed to be known a priori based on the assumption of constant speed flight and fuel consumption. Using Eqs. 4,7 we define the average capacity used by the UAV group performing the assignment

$$C = E(J)/J_R \quad (8)$$

Note that in this study the average was taken over all the Monte Carlo simulation runs performed. Figure 7 shows a plot of the amount of mission capability used by the team (C) to perform the candidate assignment versus the number of nodes processed. All engagements are with $N_v = 4$, $N_t = 3$ and three tasks that have to be performed on each target. The dashed lines represent the standard deviation. This plot can be used to judge the amount of time it will take to find an acceptable use of the UAV group capability for a particular mission. That is, a plot of this sort can be used to limit the processing time of the algorithm based on the needs of the mission.

V. CONCLUSION

The representation of the UAV assignment problem as a tree that need be searched allows incorporating all the states of the problem. The proposed SSBFS algorithm, which is a deterministic search method, has desirable qualities such as providing fast initial feasible solutions that monotonically improve and, eventually, converge to the optimal solution. Since the nodes in the tree represent the physical and temporal system, adding new constraints is relatively simple. In this paper, the total distance travelled by the UAV team members was minimized to produce the optimal assignment. Other objectives such as minimum target prosecution time, maximum target value and minimum per vehicle fuel usage, could be implemented by changing the per node calculations.

The characteristics of the algorithm of providing a fast feasible solution is of prime importance for very large

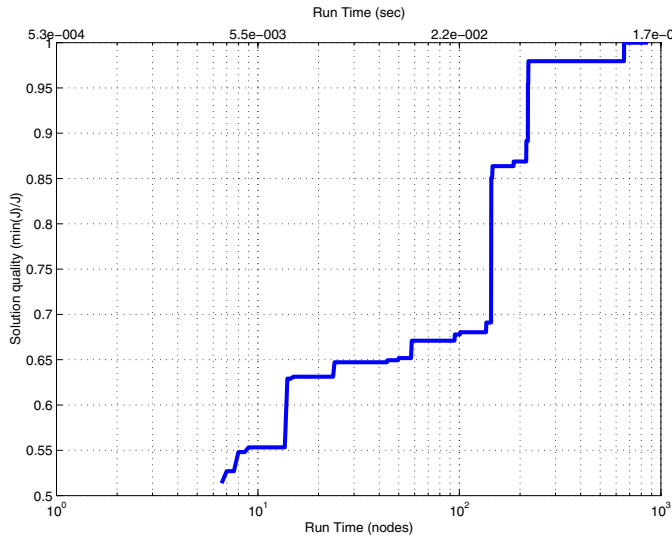


Fig. 4. Solution quality for a 4 vehicles, 2 targets engagement.

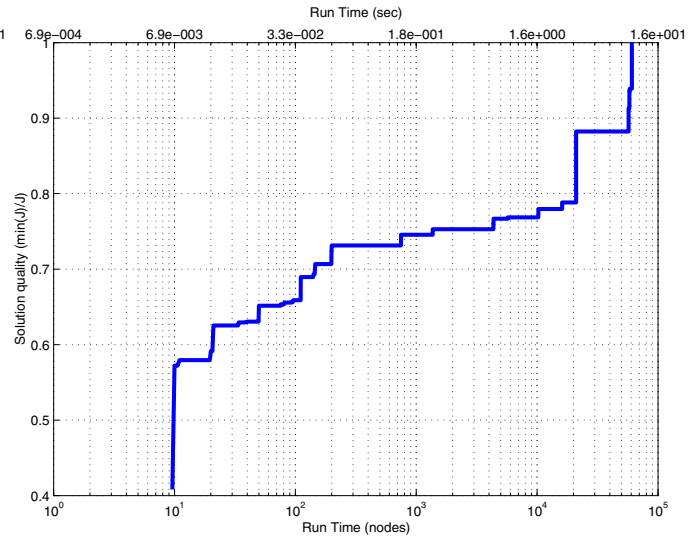


Fig. 5. Solution quality for a 4 vehicles, 3 targets engagement.

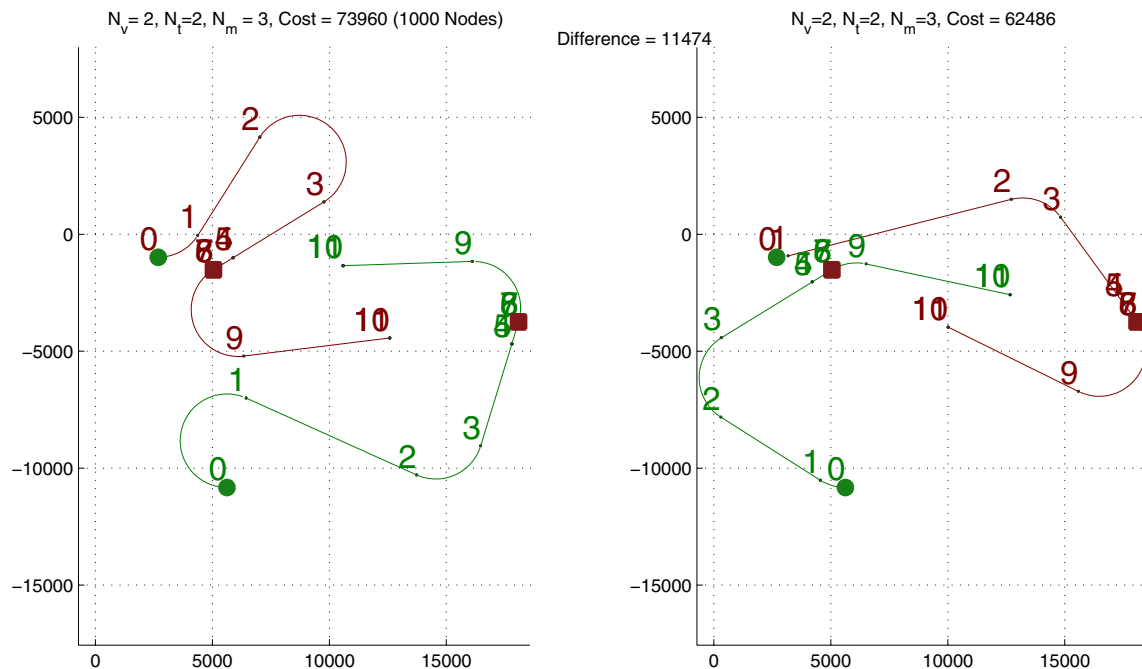


Fig. 6. Assignment trajectories for a 2 vehicles, 2 targets engagement. Left figure is an assignment based on 100 nodes, and right figure is optimal assignment.

dimensional problems. Another key attribute of the SSBFS algorithm to the BSTAP is the ability to improve the solution over time. This makes it possible to tailor the run-time of the algorithm to the situation. For instance, if the vehicles and targets are relatively close together then the initial solutions can be used, but if the targets are farther away, more nodes can be processed and a high quality assignment solution can be achieved.

A drawback of the SSBFS algorithm is that there are no guarantees on the rate of convergence to the optimal solution. Especially for large dimensional problems this process can take a significant amount of time. Thus, for

this algorithm to be implementable in a real system, improvements are required in the convergence process. In this direction two approaches can be taken: finding a faster search method and exploring receding horizon techniques. For example, more powerful stochastic search techniques, such as genetic algorithms, that implicitly use the gradients in the problem and do not converge to local minima may prove beneficial. Utilizing receding horizon techniques would make it possible to reduce the node space that needs to be searched, thus speeding up the convergence process.

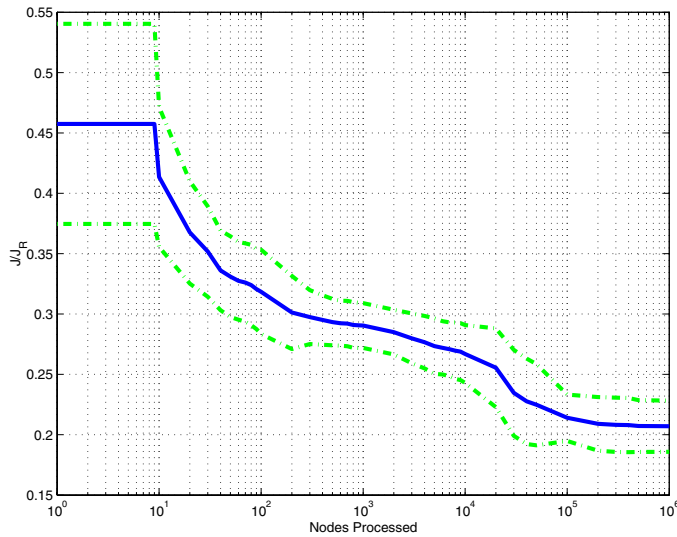


Fig. 7. Mean capacity used for all 4 vehicle, 3 target engagements.

REFERENCES

- [1] Chandler, P. R., Pachter, M., Swaroop, D., Fowler, J. M., Howlet, J. K., Rasmussen, S., Schumacher, C., and Nygard, K., "Complexity in UAV Cooperative Control," *Proceedings of the American Control Conference*, Anchorage, Alaska, 2002.
- [2] Guo, W. and Nygard, K., "Combinatorial Trading Mechanism for Task Allocation," *13th International Conference on Computer Applications in Industry and Engineering*, 2002.
- [3] Murphy, R. A., "An Approximate Algorithm For a Weapon Target Assignment Stochastic Program," *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*, Kluwer Academic Publishers, 1999.
- [4] Nygard, K. E., Chandler, P. R., and Pachter, M., "Dynamic Network Flow Optimization Models for Air Vehicle Resource Allocation," *Proceedings of the American Control Conference*, Arlington, Virginia, 2001.
- [5] Schumacher, C. J., Chandler, P. R., and Rasmussen, S. J., "Task Allocation for Wide Area Search Munitions Via Network Flow Optimization," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, 2001.
- [6] Alighanbari, M., Kuwata, Y., and How, J. P., "Coordination and Control of Multiple UAVs with Timing Constraints and Loitering," *Proceedings of the American Control Conference*, Denver, Colorado, 2003.
- [7] Rasmussen, S. J., Chandler, P. R., Mitchell, J. W., Schumacher, C. J., and Sparks, A. G., "Optimal vs. Heuristic Assignment of Cooperative Autonomous Unmanned Air Vehicles," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Austin, Texas, 2003.
- [8] Richards, A., Bellingham, J., Tillerson, M., and How, J. P., "Coordination and Control of Multiple UAVs," *Proceedings of the 2002 AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, 2002.
- [9] Schumacher, C., Chandler, P., Pachter, M., and Pachter, L., "Constrained Optimization for UAV Task Assignment," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Providence, RI, 2004.
- [10] Rasmussen, S. J., Mitchell, J. W., Schulz, C., Schumacher, C. J., and Chandler, P. R., "A Multiple UAV Simulation for Researchers," *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, Austin, TX, 2003.