2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Time-Dependent Cooperative Assignment

Derek B. Kingston
Electrical and Computer Engineering
Brigham Young University
Provo, Utah 84602
kingston@byu.edu

Corey J. Schumacher
Air Vehicles Directorate
Air Force Research Laboratory (AFRL/VACA)
Wright-Patterson AFB, Ohio 45433
Corey.Schumacher@wpafb.af.mil

Abstract— **The problem of assigning multiple agents to time-dependent cooperative tasks is addressed using a Mixed-Integer Linear Program. A time-dependent cooperative task is a task requiring multiple agents to perform separate subtasks simultaneously or within some predetermined margin where agent availability to perform a subtask is limited to specific intervals in time. By separating the underlying calculation of agent availability and cost from the mechanism of assignment, a method to solve complex cooperative assignment problems can be formulated. A cooperative UAV target tracking/target prosecution scenario is presented to illustrate the assignment method.**

I. INTRODUCTION

An important element of any autonomous team operation is the ability to assign members of the team to specific roles or tasks. A typical assignment problem involves assigning n tasks to n agents and can be solved efficiently using a linear program [1]. Extensions to this method involve iteratively building up sequences of these assignments to complete a mission [2], [3] or solving a dual maximal network flow problem [1]. These methods benefit from the power of a linear program to quickly solve large problems, but sacrifice the flexibility needed in some assignment scenarios. For example, strict precedence of tasks is difficult to enforce and tasks that require cooperation between agents can be hard to encode. In addition, some agents may have windows of availability in which particular tasks can/cannot be assigned.

In an effort to address some of these concerns, Schumacher et al [4], [5] developed a Mixed-Integer Linear Program (MILP) that enforces task timing precedence and generates a complete team assignment tour. However, in order to do this, the cost between consecutive tasks must be known *a priori*. Euclidean distance between targets was used as the cost between tasks, but this simplification leads to suboptimal results since the actual cost between tasks can be significantly greater due to the dynamic constraints of the agents.

Assignment problems with strict task precedence have also been addressed by [6], [7] and [8]. In [6], a complete representation of the search space for UAV task assignment is developed which can be searched directly or with a Genetic Algorithm. Turra et al [7] add the additional complication of moving targets to the strict task precedence assignment problem and present a solution with a pipeline process that performs the most computationally-intensive tasks off-line, creating an algorithm that is implementable in real time. Finally, Alighanbari et al [8] present a method to construct tours of sequentiality constrained tasks which can be solved optimally for small problems and with a Tabu search as the computation becomes intractable. Unfortunately, none of these methods provide explicit methods to account for agent availability windows or tasks that are constrained differently than a simple ordering in time. Additionally, highly coupled tasks resist attempts to be decentralized, making methods similar to [9] and [10] difficult to apply.

We present an assignment method that addresses task timing constraints, agent dynamic constraints (implicitly), cooperative tasks, and agent availability time windows in Section II. Extending this method to find a tour of assignments is discussed in Section IV in connection with the application scenario described in Section III. Simulation results are shown in Section V and Section VI gives conclusions and directions for future work.

II. TIME-DEPENDENT ASSIGNMENT METHOD

In many cooperative scenarios, the ability to assign cooperative tasks (i.e. two or more agents are assigned subtasks at the same target) is critical to mission performance. This assignment is often complicated by the fact that agents may have windows in time when they can complete specific tasks. In a UAV scenario, these windows are typically a product of the underlying dynamic constraints of the vehicle. For example, if a coordinated task requires one agent to track a target while the other attacks it, then one agent may need to adjust its path to wait for its partner. If the path cannot be extended smoothly (e.g. UAV attack paths cannot be extended without discontinuity in some cases [11]) then there are separate intervals in which the task can be done with a period of infeasibility in between. A MILP formulation of the problem allows these constraints to be addressed[1]. Once the problem nomenclature has been established, linear constraints are given that satisfy the requirements of time-dependent task assignment.

A. Nomenclature

Let K be the number of tasks to be performed on each target, V be the number of agents, and N be the number

[1]It has been brought to the authors' attention that the problem setup presented here bears striking resemblance to models of air traffic network flow. See, for example, Ref. [12].

of targets. Also, let $x_{v,k}^{n\,(w)}$ be a binary decision variable indicating that agent v is to perform task k at target n starting in time window (w). Note that the number of decision variables grows as the product of the numbers of agents, targets, tasks, and windows. To allow for the case when there are more agents than tasks to be done, let z_v be a binary decision variable indicating that agent v is to be assigned the null task (assumed to have zero cost). The case when there are more tasks than agents must be addressed in an iterative manner and will be discussed in Section IV. To enforce the timing between tasks and to optimize the total time of the assignment, let t_k^n be the time that task k is started on target n.

To represent the cost associated with each of the decision variables, the windows of time when agent v can accomplish task k at target n must be calculated. For each (v, k, n), a set $W_{v,k}^n$ is formed where each element of $W_{v,k}^n$ contains a start time ($T_{v,k}^{n\,\lfloor w}$) of window w and a stop time ($T_{v,k}^{n\,w\rfloor}$). This can be done in *any* manner suitable to the problem definition. It is this flexibility that gives this assignment method its appeal – as long as the underlying path planning can be represented by windows of task availability, this assignment algorithm can be used. Note that a worst-case target prosecution time exists for each target and the maximum of those times is the upper-bound on the final task completion time. Practically, an upper-bound on target completion time will always exist due to fuel constraints. This maximum target completion time, T, allows the formation of timing inequalities that support task precedence.

B. Constraints

A Mixed-Integer Linear Program is defined by linear constraints and the cost function that describe the problem. Following [4], a set of constraints and associated cost function to accomplish the assignment is presented below.

1) Non-Timing Constraints:

1. Each agent gets exactly one task or goes to the sink.

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{w} x_{v,k}^{n\,(w)} + z_v = 1 \tag{1}$$
$$\text{for } v = 1\ldots V$$

2. Any target that receives one task, receives all tasks.

$$\sum_{v=1}^{V}\sum_{w} x_{v,1}^{n\,(w)} = \sum_{v=1}^{V}\sum_{w} x_{v,k}^{n\,(w)} \tag{2}$$
$$\text{for } n = 1\ldots N,\ k = 2\ldots K$$

3. Each target is serviced at most once (in combination with the above constraint, this also ensures that each target receives each task at most once).

$$\sum_{v=1}^{V}\sum_{w} x_{v,1}^{n\,(w)} \le 1 \tag{3}$$
$$\text{for } n = 1\ldots N$$

2) Timing Constraints:

1. Time to start task k at target n must be in window w of agent v if the corresponding decision has been made. Note that these inequalities are trivially satisfied when $x_{v,k}^{n\,(w)}$ is zero, but become tight restrictions on t_k^n when $x_{v,k}^{n\,(w)}$ is one.

$$t_k^n \ge T_{v,k}^{n\,\lfloor w} - (1 - x_{v,k}^{n\,(w)})2NT \tag{4}$$
$$t_k^n \le T_{v,k}^{n\,w\rfloor} + (1 - x_{v,k}^{n\,(w)})2NT \tag{5}$$
$$\text{for } k = 1\ldots K,\ v = 1\ldots V,\ n = 1\ldots N,\ w \subset W_{v,k}^n$$

2. If precedence of tasks is required, then constraints similar to the following will be needed. Here we constrain the tasks to occur in order $1\ldots K$.

$$t_k^n \le t_{k+1}^n \tag{6}$$
$$\text{for } n = 1\ldots N,\ k = 1\ldots K{-}1$$

To have agents cooperate in servicing a target simply requires defining the relative timing of tasks. If, for example, two UAVs were to start two cooperative tasks (say task 1 and 2) simultaneously, then the constraint $t_1^n = t_2^n$ could be added. Similarly, if a task must occur within some interval after a previous task is performed, a constraint pair like ($t_2^n \le t_1^n + \alpha$, $t_2^n \ge t_1^n$) is applied.

3. To ensure that as many targets as possible are serviced, we impose a missed target penalty. Note that all t_k^n that are associated with targets that are missed are equal to MT where M is the number of missed targets. This constraint also eliminates the degenerate solution of assigning all agents to the null task.

$$t_k^n \ge \left(\sum_{m=1}^{N}\left\{1 - \sum_{v=1}^{V}\sum_{w} x_{v,k}^{m\,(w)}\right\}\right)T - \left(\sum_{v=1}^{V}\sum_{w} x_{v,k}^{n\,(w)}\right)2NT \tag{7}$$
$$t_k^n \le \left(\sum_{m=1}^{N}\left\{1 - \sum_{v=1}^{V}\sum_{w} x_{v,k}^{m\,(w)}\right\}\right)T + \left(\sum_{v=1}^{V}\sum_{w} x_{v,k}^{n\,(w)}\right)2NT \tag{8}$$
$$\text{for } k = 1\ldots K,\ n = 1\ldots N$$

C. Cost Function

Reasonable cost functions for many assignment scenarios include minimizing the final task completion time for all targets

$$J = \sum_{n=1}^{N} t_K^n \tag{9}$$

or minimizing all task completion times for all targets

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} t_k^n. \tag{10}$$

III. UAV ASSIGNMENT SCENARIO

One scenario which requires a high level of cooperation between team members and has the additional complexity of nonlinear agent dynamics is the Cooperative Moving Target Engagement (CMTE) scenario. CMTE requires that two or more UAVs track a moving (ground) target with doppler radar while an additional UAV launches a GPS-guided munition. The sensed target positions and associated error ellipses from each tracking UAV are fused to form a precise GPS location of the target for the munition to follow. In order to reduce the error in the location of the moving target, the UAVs tasked to perform the tracking must have different line-of-sight angles to the target, preferably near orthogonal views. In addition, a moving target can only be detected and tracked if the UAV has a line-of-sight view to the target within some offset angle, γ, from the heading of the moving target, ψ. Figure 1 shows the heading of the target and the associated regions in which UAVs can be located to detect its motion.

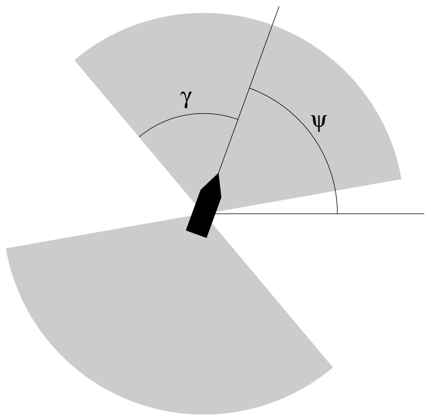


Fig. 1. Region of detectability based on target heading.

Complicating matters further, each UAV has a sensor footprint in which targets must be located to be tracked. The footprint has minimum and maximum ranges and bearings and, due to the configuration of the radar antenna array, is pointed out the wing of the UAV. Figure 2 shows a UAV tracking a target and the associated sensor footprint relative to the orientation of the UAV. The sensor can scan on either side of the UAV, but not both at the same time.

A team of UAVs designated to track and prosecute an area of targets can also be supported by an additional "off-board" team member who is located outside of the area and has a powerful sensor with an assumed 360-degree sensor footprint able to view the entire field. This assumption can be relaxed, but requires additional timing constraints. For purposes of this assignment algorithm, the off-board vehicle is assumed to travel circularly so that its line-of-sight to targets is approximately fixed. UAVs inside the area can then cooperatively track a target with the off-board vehicle or with an inside team member. Because the error in the position of the moving target can be reduced by multiple separated line-of-sight angles to the target, we restrict the difference in bearing angles of the UAVs to the target to be greater than 45 degrees. This restriction partitions the detectability region of the target further into regions that satisfy both the target detectability requirement and the angle offset requirement. For fixed target heading and position and fixed off-board vehicle position, regions where a target can be cooperatively tracked can be identified and used to develop path-planning routines of the UAVs to complete the mission.

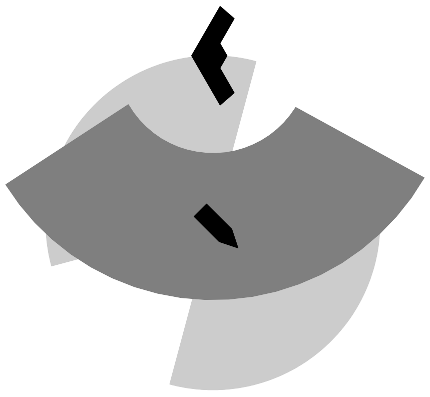


Fig. 2. UAV sensor footprint (dark gray region).

While the bulk of the complexity in the CMTE scenario comes from the cooperative tracking of targets, the attack on the target must also be considered. All UAVs inside the area of interest can track targets and drop weapons. To be in position to attack, a UAV must be headed toward the target and be within a maximum and minimum range. Once the weapon is launched, the attacking UAV is free to be reassigned to other tasks, but the UAVs tracking the target must track the target for the duration of the weapon flight.

The CMTE scenario requires strict timing constraints between tasks and cooperation between team members. UAV dynamics impose constraints on path planning techniques to get into tracking and attacking positions. This complexity makes the CMTE scenario an ideal problem for studying time-dependent cooperative assignment methods.

IV. CMTE APPLICATION

To reduce the CMTE scenario to a more reasonable level, the following assumptions and restrictions will be added.

1. Targets have constant heading. Admittedly, this is a poor assumption, but for targets traveling along known roads, it may be justified. Allowing dramatic target heading changes requires many more UAVs to be assigned to track the target to ensure coverage over all possible headings.
2. Tracking of targets occurs along arcs of a circle centered at the target with radius so as to place the target in the center of the sensor footprint (see Fig. 2). This allows the path planning to be performed as if the target were stationary since the sensor footprint is

much larger than the distance traveled by the target during a typical scenario.

3. Weapons are launched at a fixed distance from the target and an upper bound on the flight time of the weapon is known so as to fix the amount of time after an attack has occurred that the target must be tracked. This simply translates to planning more time to track than will be actually be needed for the weapon launch and travel.

These restrictions and assumptions simplify the level of path planning needed to accomplish a CMTE mission. Additional complexity could be added without changing the method of assignment as long as the interface between the nonlinear path planning and the assignment algorithm remains abstracted to the specification of windows of availability of team agents.

Because the CMTE scenario hinges on the ability of UAVs to cooperatively track a moving target, much of the assignment complexity is involved with determining which team members are assigned to track which target and with whom. To this end, the basic time-dependent assignment algorithm developed in Section II is augmented with additional decision variables to allow pairwise decisions. Let $y^{n\,(w)}_{u,v}$ be a binary decision variable indicating that UAVs u and v are assigned to cooperatively track target n in time window w. This allows the path planning routines to calculate windows of time when the pair of vehicles (u, v) can cooperatively track a target.

Following the nomenclature established above and in Section II, let $k = 1$ be designated the *attack* task and $k = 2$ be the *track* task, then the following constraints are used to assign a team of UAVs in the CMTE scenario.

A. Non-Timing Constraints

1. Each agent gets exactly one task or goes to the sink. An agent can be assigned to cooperatively track a target with the off-board vehicle ($x^{n\,(w)}_{v,2}$) or with another inside team member ($y^{n\,(w)}_{u,v}$), but not both. At a higher level, an agent could be assigned to attack ($x^{n\,(w)}_{v,1}$) or track, but not both.

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{w} x^{n\,(w)}_{v,k} + \sum_{n=1}^{N}\sum_{u=1,u\neq v}^{V}\sum_{w} y^{n\,(w)}_{u,v} + z_v = 1 \quad \text{for } v = 1\ldots V \tag{11}$$

2. Any target that receives one task, receives all tasks. Since the *track* task occurs in two different decision variables, the sum of both must equal the decision to complete the *attack* task by another vehicle.

$$\sum_{v=1}^{V}\sum_{w} x^{n\,(w)}_{v,1} = \sum_{v=1}^{V}\sum_{w} x^{n\,(w)}_{v,2} + \sum_{u=1}^{V-1}\sum_{v=u+1}^{V}\sum_{w} y^{n\,(w)}_{u,v} \quad \text{for } n = 1\ldots N \tag{12}$$

3. Each target is serviced at most once (in combination with the above constraint, this also ensures that each target receives each task at most once).

$$\sum_{v=1}^{V}\sum_{w} x^{n\,(w)}_{v,1} \leq 1 \quad \text{for } n = 1\ldots N \tag{13}$$

B. Timing Constraints

The CMTE scenario requires that a target be continuously tracked for the duration of the weapon flight after the weapon has been launched. Since it is assumed that the weapon is launched from a specific distance from the target and the speed of the munition is known, the path planning routines can return windows of time when an agent (or pair of agents) can start tracking a target and continue for at least t_α where t_α is the time from munition launch to impact. This also allows the compaction of t^n_1 and t^n_2 to t^n since we can constrain the tracking to begin when the weapon is launched without changing the requirements of the scenario.

1. Time to prosecute target n must be in window w of agent v if the corresponding decision has been made.

$$t^n \geq T^{n\,\lfloor w}_{v,k} - (1 - x^{n\,(w)}_{v,k})2NT \tag{14}$$
$$t^n \leq T^{n\,w\rfloor}_{v,k} + (1 - x^{n\,(w)}_{v,k})2NT \tag{15}$$
$$\text{for } k = 1\ldots K,\ v = 1\ldots V,\ n = 1\ldots N,\ w \subset W^n_{v,k}$$
$$t^n \geq T^{n\,\lfloor w}_{u,v} - (1 - y^{n\,(w)}_{u,v})2NT \tag{16}$$
$$t^n \leq T^{n\,w\rfloor}_{u,v} + (1 - y^{n\,(w)}_{u,v})2NT \tag{17}$$
$$\text{for } u = 1\ldots V{-}1,\ v = u{+}1\ldots V,\ n = 1\ldots N,\ w \subset W^n_{u,v}$$

2. Due to the reduction of timing variables to the representative target prosecution time, no additional task timing constraints are needed.
3. Missed target penalty carries over from Section II to ensure that all UAVs are not assigned to the null task. Note that only $x^{n\,(w)}_{v,1}$ needs to be examined due to the coupling through constraint (12), which ensures that targets that are not attacked will not be tracked either.

C. Cost Function

The CMTE scenario simply requires that all targets are prosecuted as quickly as possible.

$$J = \sum_{n=1}^{N} t^n \tag{18}$$

The constraints given in Sections IV-A and IV-B in connection with the cost function in IV-C define a Mixed-Integer Linear Program suitable for assigning each UAV in a team to one task in a CMTE scenario. The completion of the mission is when all the tasks are completed, so the assignment algorithm must be iterated to produce a tour of assignments. A completely optimal solution would require the optimization of path planning and assignment to be coupled. By applying this assignment scheme, a tour of assignments will be suboptimal, but the freedom allowed by separating the path planning and the assignment is desirable. By augmenting the assignment method with an additional set of complete target ordering constraints, heuristics can be used to improve the iterative solution. Specifically, we use the solution to a Traveling Salesman Problem with the targets as cities to guide the iteration, since the distance between targets is a good indication of the spatial coupling of the scene.

The time windows calculated in the underlying path planning routines can easily be shifted by the amount of time a vehicle has already committed to, so an iteration can be used where the state (position and time committed) of the vehicles is updated after each assignment stage is run. Once the assignment algorithm has allocated a task to each vehicle in the team, the earliest target prosecution time is selected. The target corresponding to that time is removed from the target list and the vehicles assigned to the tasks pertaining to the prosecution of that target have their states updated to reflect the time it will take to complete their respective tasks. Vehicles associated with tasks related to the prosecution of other targets are not updated. New time windows are computed with the updated positions and shifted by the amount of time committed to earlier stages. The assignment algorithm is iterated until all targets have been prosecuted.

Obtaining a solution to any MILP formulation is *NP*-hard. This assignment algorithm is based on MILP and, so, is also *NP*-hard, resulting in extreme amounts of computation needed for large problems. A number of different random CMTE situations were simulated to estimate the computation required for various problem sizes. It was found that problems in which the sum of the number of vehicles and targets is less than or equal to 12 are solvable in less than a minute on a modern desktop computer (we used MATLAB and an open-source linear programming package, GLPK). For many CMTE missions, small problem sizes are typical, involving 5 UAVs and 3 to 4 targets. Larger problems will require a re-formulation of the assignment algorithm or a partitioning of the team and target space into problems small enough to be solved in a timely manner.Solving a CMTE mission in small stages has the additional advantage that many of the assumptions used to simplify the path planning (e.g. constant heading) will be invalidated for lengthy tours of target prosecution, and hence, only a small number of targets will be prosecuted (and assigned) at each stage.

V. SIMULATION EXAMPLE

To illustrate the capability of the time-dependent cooperative assignment algorithm presented in Section IV, a problem of size $V = 5$ and $N = 3$ was simulated. UAVs and targets were randomly distributed over an area 110 km wide and 170 km long with an additional off-board vehicle fixed directly north of the area of interest. Figure 3(a) shows the initial positions of the targets and in-area UAVs. Each target is shown with an associated detectability region (outlined in black) and cooperative tracking region (solid wedge). Recall that the cooperative tracking region is the intersection of the detectability region with the line-of-sight angles greater than 45 degrees different from the off-board vehicle line-of-sight. Task time availability windows are computed based on computing minimum time trajectories to these regions.

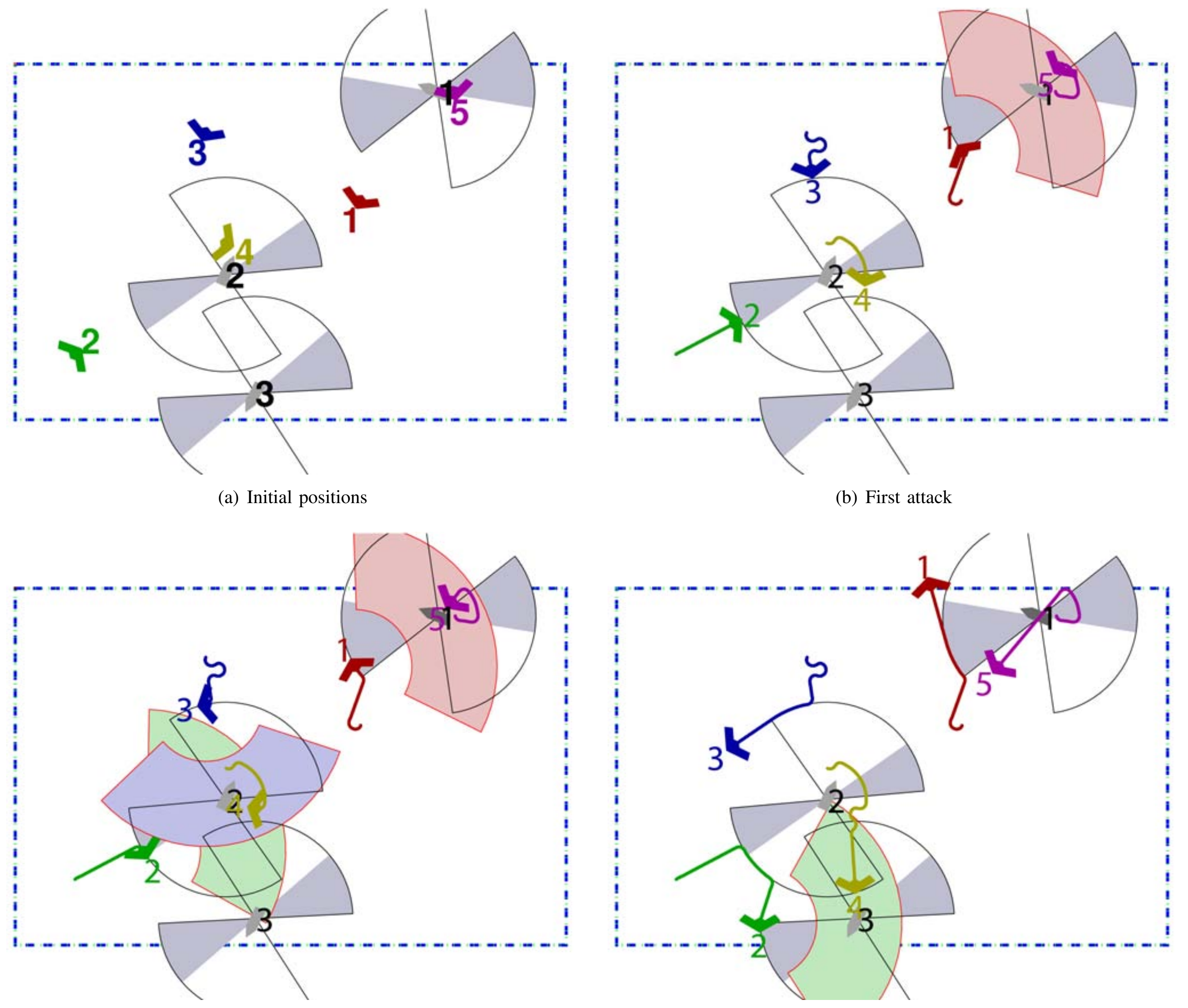
The CMTE scenario is rich in timing complexity making visualization difficult without animation. Figures 3(a)-3(d) show the progression of the simulated scenario at 4 distinct points in time. Figure 3(b) shows that UAV 1 is assigned to track target 1 in cooperation with the off-board vehicle while UAV 5 attacks. The sensor footprint of the tracking UAV is shown to validate that the UAV is in position to track the target. Because UAV 1 is in the cooperative tracking region of target 1, no other UAVs are needed to track this target. This represents one iteration of the assignment algorithm. During the next iteration, UAVs 2 and 3 are assigned to cooperatively track target 2. Figure 3(c) shows the instant in time when UAV 4 releases a weapon to attack target 2. Note that the assignment algorithm correctly assigned 2 UAVs to track this target due to the distance needed for UAV 2 or 3 to reach a cooperative tracking region with sufficient room to track the weapon for the entire weapon flight. Also note that UAV 3 extended its path to arrive at the correct position and time to track the target. Since the algorithm ensures that the target prosecution time falls in the availability time windows of each UAV, no vehicle will be given requirements that violate underlying dynamic constraints. The final target is attacked by UAV 4 with UAV 2 assigned to track in cooperation with the off-board vehicle (Fig. 3(d)).

This CMTE mission needed 70 variables (67 binary, 3 continuous) and 81 constraints to describe the optimization and required slightly less than 0.2 seconds to solve the MILP formulation. Current path planning routines are scripted in MATLAB and, for this scenario, required about 10 seconds of computation. It is anticipated that the path planning routines will require significantly less computation as code is moved from MATLAB to C; additionally, for large problems, the optimization will be the most time-intensive part.

VI. CONCLUSIONS AND FUTURE WORK

An assignment algorithm capable of dealing with agent availability time intervals and explicit task precedence was presented. The flexibility allowed by abstracting the agent path planning from the assignment algorithm allows for complex assignment scenarios to be considered. The Cooperative Moving Target Engagement (CMTE) scenario was presented as an example of a situation in which traditional assignment algorithms are not sufficient. An assignment formulation for the CMTE scenario was presented, along with a discussion of issues related to task tours and computation requirements.

Future work in this area involves finding heuristics to guide the selection of target ordering as well as partitioning strategies to break large problems into computationally



(a) Initial positions

(b) First attack

(c) Second attack

(d) Final attack

Fig. 3. Simulated CMTE scenario.

tractable ones. Tighter integration of the linear elements of path planning and the assignment algorithm are also being investigated in an attempt to gain back the optimality lost in the problem assumptions.

REFERENCES

[1] D. G. Luenberger, *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[2] C. Schumacher, P. Chandler, and S. Rasmussen, “Task allocation for wide area search munitions via iterative network flow,” in *Proc. AIAA Guidance, Navigation and Control Conference*, 2002.

[3] C. Schumacher, P. Chandler, S. Rasmussen, and D. Walker, “Task allocation for wide area search munitions with variable path length,” in *Proc. IEEE American Control Conference*, 2003.

[4] C. Schumacher, P. Chandler, M. Pachter, and L. Pachter, “UAV task assigment with timing constraints,” in *Proc. AIAA Guidance, Navigation and Control Conference*, 2003.

[5] ——, “UAV task assignment with timing constraints via mixed-integer linear programming,” in *Proc. AIAA 3rd Unmanned Unlimited Systems Conference*, 2004.

[6] S. J. Rasmussen, J. W. Mitchell, A. G. Sparks, T. Shima, and P. R. Chandler, “Use of state-space search to improve the performance of assignment algorithms for autonomous UAVs,” in *Proc. IEEE Conference on Decision and Control*, 2004.

[7] D. Turra, L. Pollini, and M. Innocenti, “Real-time UAVs task allocation with moving targets,” in *Proc. AIAA Guidance, Navigation and Control Conference*, 2004.

[8] M. Alighanbari, Y. Kuwata, and J. P. How, “Coordination and control of multiple UAVs with timing constraints and loitering,” in *Proc. IEEE American Control Conference*, 2003.

[9] J. Cortes, S. Martinez, and F. Bullo, “Coordinated deployment of mobile sensing networks with limited-range interactions,” in *Proc. IEEE Conference on Decision and Control*, 2004.

[10] E. Frazzoli and F. Bullo, “Decentralized algorithms for vehicle routing in a stochastic time-varying environment,” in *Proc. IEEE Conference on Decision and Control*, 2004.

[11] C. Schumacher, P. Chandler, S. Rasmussen, and D. Walker, “Path elongation for UAV task assignment,” in *Proc. AIAA Guidance, Navigation and Control Conference*, 2003.

[12] D. Bertsimas and S. S. Patterson, “The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach,” *Transportation Science*, vol. 34, pp. 239–255, 2000.