# Optimization of Air Vehicles Operations Using Mixed-Integer Linear Programming

Corey Schumacher
Air Force Research Laboratory (AFRL/VACA)
Wright-Patterson AFB, OH 45433-7531

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

Meir Pachter
Department of Electrical Engineering
Air Force Institute of Technology (AFIT/ENG)
Wright-Patterson AFB, OH 45433-7765

Lior S. Pachter
Department of Mathematics
University of California at Berkeley
Berkeley, CA 94720-3840

**Abstract**

A scenario where multiple air vehicles are required to prosecute geographically dispersed targets is considered. Furthermore, multiple tasks are to be successively performed on each target, i.e. the targets must be classified, attacked, and verified as destroyed. The optimal, e.g. minimum time, performance of these tasks requires cooperation amongst the vehicles such that critical timing constraints are satisfied, that is, a target must be classified before it can be attacked, and an air vehicle is sent to a target area to verify its destruction only after the target has been attacked. In this paper, the optimal task assignment/scheduling problem is posed as a mixed integer linear program (MILP). The solution of the MILP assigns all tasks to the vehicles and performs the scheduling in an optimal manner, including staged departure times. Coupled tasks involving timing and task order constraints are automatically addressed. When the air vehicles have sufficient endurance, the existence of a solution is guaranteed.

Keywords: Linear Programming, Military, Optimization, Planning, Scheduling

## Introduction

In this paper, the optimization of air-to-ground operations is undertaken (Chandler *et al*, 2001). A scenario where multiple Unmanned Air Vehicles (UAVs) are required to service geographically dispersed targets is considered. Moreover, multiple tasks must be successively performed on each target, viz., the targets must be classified, attacked, and the damage inflicted on the targets must be assessed. The floating timing constraints are critical: A target cannot be attacked before it is classified, and a UAV will be sent to the target area only after the target attack has been executed. Multi-role UAVs are considered s.t. each UAV can perform all of the tasks. A case in point: Autonomous Wide Area Search Munitions (WASM) are small UAV's, each with a turbojet engine and sufficient fuel to fly for thirty minutes. WASM are deployed in groups from aircraft flying at higher altitudes; they are typically deployed in groups of four, although larger teams are certainly possible. They are individually capable of autonomously searching for, recognizing, and attacking targets. The ability to network, that is, to communicate target information to one another, and consequently to cooperate, will greatly improve the weapon system's effectiveness of future UAV teams. The insertion of this technology into WASM systems under development is currently being investigated. Thus, the problem is posed of planning the performance of the UAVs' tasks such that critical timing constraints are satisfied. This entails optimal assignment and scheduling.

In (Schumacher *et al*, 2002, Schumacher *et al*, 2003[a], Schumacher *et al*, 2003[b]) a time-phased network optimization model was used to perform task allocation for a team of UAVs. The model is run simultaneously on all air vehicles at discrete points in time, and assigns each vehicle one or more tasks each time it is run. The network optimization model is run iteratively so that all of the known targets will be prosecuted by the resulting allocation. 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, thus achieving feedback action. Classification, attack, and battle damage assessment tasks can all be assigned to different vehicles when a target is found, resulting in the target being more quickly serviced. A single vehicle can also be given multiple task assignments to be performed in succession, if that is more efficient than having multiple vehicles perform the tasks individually. In (Schumacher *et al*, 2003[a]), variable path lengths are included to guarantee that feasible trajectories will be calculated for all tasks. This method is computationally efficient and scales well, however the iterative procedure is heuristic and suboptimal. Tabu search can be used to solve difficult combinatorial optimization problems, e.g., the vehicle routing problem with fixed time windows (Toth *et al*, 2002), and (Gendreau *et al*, 1994, Glover and Laguna, 1997, O'Rourke *et al*, 2000). In this paper a solution method for air vehicle routing combinatorial optimization problems with floating time windows, as described above, is developed. Moreover, we feel that there is an urgent need to develop an exact

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| | | |
|---|---|---|
| 1. REPORT DATE<br>**2006** | 2. REPORT TYPE | 3. DATES COVERED<br>**00-00-2006 to 00-00-2006** |

| | |
|---|---|
| 4. TITLE AND SUBTITLE<br>**Optimizaiton of Air Vehicle Operations Using Mixed-Integer Linear Programming** | 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)<br>**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

14. ABSTRACT

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT<br>**unclassified** | b. ABSTRACT<br>**unclassified** | c. THIS PAGE<br>**unclassified** | | **20** | |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std Z39-18

solution method against which heuristic methods, which are applicable to larger problems, can be benchmarked - see, e.g., (Hooker, 1995), where this is strongly advocated.

This paper addresses the optimal formulation for solving the coupled multiple-assignment and scheduling problem. Continuous timing variables are introduced. This, in turn, leads to the formulation of the optimization problem as a Mixed Integer Linear Program (MILP) (Nemhauser and Wolsey, 1988) and allows the optimal solution to be found while satisfying all timing constraints. A preliminary version of the MILP formulation for task assignment and scheduling given here was first presented in (Schumacher et al. 2004[a]). In this paper, time is treated as a continuous variable and a rigorous optimal task assignment/scheduling algorithm is developed. This requires the solution of a mixed integer linear program (ILOG 1999).

The formulation presented here can be solved optimally for some realistic problem sizes, e.g., a team of five UAVs servicing three targets, without requiring approximate solution. Loiter, if it is needed, is not handled at target waypoints. Rather, the method presented here allows staged departure times, before vehicles begin their tour of targets and tasks. This obviates the need for loiter in potentially dangerous target areas. Also, our MILP formulation is flexible enough to allow the consideration of many interesting cost functions, e.g., mission completion in minimum time, shortest total paths lengths traveled by the vehicles, or maximization of the number of air vehicles which survive the mission.

The method presented here can also accommodate fixed time windows, as in the Vehicle Routing Problem (VRP) (Toth *et al*, 2002), although arbitrary fixed time windows can make the problem, independent of solution methodology, infeasible. Moreover, the MILP presented here can also accommodate dynamic and logical constraints on task performance, as is the case in scheduling problems (Pinedo, 2002). In this work, without fixed time windows, feasibility is guaranteed, as long as the number of air vehicles exceeds the number of targets, even with three or more tasks per target, provided the air vehicles have sufficient endurance.

## Nomenclature

$i$ = Start node index

$j$ = Arrival node index

$J$ = Cost function

$k$ = Task index

$n$ = Number of targets

$t_{ij}^{(v,k)}$ = Time required for air vehicle $v$ to fly from node $i$ to node $j$ to perform task $k$ at node $j$

$t_j^{(k)}$ = Time of completion of task $k$ on target $j$

$T$ = Maximum endurance of any UAV

$T_{i,j}$ = Flight time between nodes $i$ and $j$

$T_v$ = Endurance of UAV $v$

$v$ = Air vehicle index

$w$ = Number of UAVs

$x_{i,j}^{(v,k)}$ = Binary task assignment variable for the task of air vehicle $v$ flying from node $i$ to node $j$ to perform task $k$ on node $j$

$x_{i,n+w+1}^{(v)}$ = Binary assignment variable for the task of air vehicle $v$ flying from node $i$ to the sink node

## Scenario

Consider $n$ geographically dispersed targets with known position and $w$ Air Vehicles (AV). We assume $w \geq n+1$. We then have $n+w+1$ nodes: $n$ target nodes, $w$ source nodes (the points of origin of the AVs), and one sink node. Nodes $1,\ldots,n$ are located at the $n$ target positions. Nodes $n+1,\ldots, n+w$ are located at the vehicle initial positions. Node $n+w+1$ is the "sink". An air vehicle with no future target assignments is relegated to the sink, i.e. will continue to search. A vehicle located at the sink cannot be reassigned.

- Spatial outlay: The flight time of AV $v$ from node $i$ to node $j$ to perform task $k$ at node $j$ is $t_{ij}^{(v,k)} \geq 0$. The indices $i=1,\ldots,n+w$, $j=1,\ldots n$, and $v = 1,\ldots,w$. The index $k$ designates the task to be performed at node $j$. Thus, it is acknowledged that the time to travel from node $i$ to node $j$ depends on the particular AV $v$'s airspeed and the assigned task $k$.

- The tasks: Three tasks must be performed on each target.

  1. $k=1$ – Classification
  2. $k=2$ – Attack
  3. $k=3$ – Target Damage Assessment (Verification)

Furthermore, once an AV attacks a target, it is destroyed and can no longer perform additional tasks. This is certainly the case for powered munitions and the WASM mission, but if the AV was a reusable aircraft, one would have to modify the problem formulation and account for the AVs' depletion of its store of ammunition following each attack.

The three tasks must be performed on each target in the order listed. This results in critical timing constraints, which set this problem apart from the classical Vehicle Routing Problem (VRP) (Toth *et al*, 2002). In the latter, rigid time windows for the arrival of the vehicles can be specified, however, the coupling brought about by the need to sequence the various tasks is absent. Evidently, our problem features some aspects of job shop scheduling (Pinedo, 2002).

In the operational scenario considered, the number of pre-specified problem parameters $t_{ij}^{(v,k)}$ is $3wn+3n(n-1)w = 3n^2w$. When Euclidean distances are used, the dimension of the parameter space is reduced to $0.5n(n-1)+wn= 0.5n(n+2w-1)$. Finally, the endurance of AV $v$ is $T_v, v=1,\ldots,w$.

Figure 1 illustrates a scenario where one stationary ground targets is engaged by three AVs. The potential target's position is known at the beginning of the optimization, but not the classification. First, the target will be overflown and imaged, then it will be attacked, and finally an AV will overfly the target for verification purposes.

## Mixed Integer Linear Programming

The Mixed Integer Linear Programming (MILP) model uses a discrete approximation of the real world based on nodes that represent discrete start and end positions for segments of a UAVs path. Nodes representing target positions range from $1\ldots n$ and nodes for initial UAV positions range from $n+1\ldots n+w$. There is also an additional logical node for the sink $n+w+1$. The sink node is used when no further assignment for the UAV is in the offing; it goes to the sink when it is done with all of its tasks, or when it is not assigned another task. In practice, when a UAV enters the sink it is then used for performing search of the battlespace. The MILP model requires the information on the costs (or times) for a UAV to fly from one node to another node. These known flight times are constants represented by $t_{ij}^{(v,k)}$, the time it takes UAV $v$ to fly from node $i$ to node $j$ to perform task $k$. The flight times are positive real numbers, $t_{ij}^{(v,k)} \geq 0$.

**Decision Variables**

The binary decision variable $x_{ij}^{(v,k)} = 1$ if AV *v* is assigned to fly from node *i* to node *j* and perform task *k* at node *j*, and 0 otherwise; *i = 1,…n+w, j = 1,…,n, v = 1,…,w,* and *k = 1,2,3.* For *i=n+1…n+w*, only $x_{n+v,j}^{(v,k)}$ exist. These variables correspond with the first task assignment each vehicle receives, starting from its unique source node. Only vehicle *v* can do a task starting from node *n+v*. For task assignments *k=1,3*, $i \neq j$ and for task assignment *k=2* we allow *i = j*; the latter allows for an AV to perform the target classification task, and immediately thereafter attack the target. Thus far, we have *wn(3n+1)* binary decision variables.

We also have the following additional binary decision variables. The decision variable $x_{i,n+w+1}^{(v)} = 1$ if AV *v* is assigned to fly from node *i* to the sink node *n+w+1*, and is 0 otherwise; *v=1,…,w* and *i = 1,…,n+w*. This adds *(n+1)w* binary decision variables. Entering the sink can also be thought of as being re-assigned to the search task.

Continuous decision variables:

The time of performance of task *k* on target *j* is $t_j^{(k)} > 0$; *k = 1,2,3* and *j = 1,…,n.* Thus, we have *3n* continuous decision variables. We also have *w* additional continous decision variables: the time AV *v* leaves *node j = n + v* is $t_v$; *v = 1,…,w*. In total we have *w[n(3n+2)+1]* binary decision variables and *3n+w* continuous non-negative decision variables.

**Cost Functions**

Possible cost functions include:

1. Minimize the total flight time of the AVs
$$J = \sum_{k=1}^{3} \sum_{v=1}^{w} \sum_{i=1}^{n+w} \sum_{j=1}^{n} t_{i,j}^{(v,k)} x_{i,j}^{(v,k)} \tag{1}$$
2. Alternatively, minimize the total engagement time. The target j is visited for the last time at time $t_j^{(3)}$. Let $t_f$ be the time at which all targets have been through Verification. Introduce an additional continuous decision variable $t_f \in \Re_+^1$. The cost function is then $J = t_f$ and we minimize *J* subject to the constraints
$$t_j^{(3)} \leq t_f,\ j = 1,\ldots,n \tag{2}$$
We also add a small weight to the time of performance of each individual task, to encourage each individual task to be completed as quickly as possible. Then
$$J = t_f + c_j^{(k)} t_j^{(k)},\ j = 1,\ldots,n, k = 1,2,3, \tag{3}$$
where $c_j^{(k)} > 0$ is a small weight on the completion time of each individual task. To weight the time of performance of individual tasks more heavily, one could use $J = c_j^{(k)} t_j^{(k)},\ j = 1,\ldots,n, k = 1,2,3.$
3. An additional cost function: The number $n_s$ of surviving UAVs, which end up in the sink. Thus, the cost
$$J = n_S = \sum_{v=1}^{w} \sum_{i=1}^{n+w} x_{i,n+w+1}^{(v)}$$
and the optimization problem considered is $\max J = n_S$.

**Constraints**

Inclusion of all of the required constraints is critical to automatically enforcing the desired vehicle behavior.

1. Mission completion requires that all three tasks are performed on each target exactly one time. Similar to linear assignment problems (Nemhauser and Wolsey, 1988) the following must hold:
$$\sum_{v=1}^{w} \sum_{i=1, i \neq j}^{n+w} x_{ij}^{(v,k)} = 1, \quad \begin{array}{c} k = 1,3 \\ j = 1,\ldots,n \end{array}, \tag{4}$$

and

$$\sum_{v=1}^{w}\sum_{i=1}^{n+w} x_{ij}^{(v,2)} = 1, \quad j = 1,\dots,n \tag{5}$$

This yields *3n* constraints.

2. Not more than one AV is assigned to perform a specific task *k* on a specified target *j*:

$$\sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,k)} \leq 1, \quad v = 1,\dots,w; \tag{6}$$

where $j=1,\dots,n$ and $k=1,3$, and

$$\sum_{i=1}^{n+w} x_{i,j}^{(v,2)} \leq 1, \quad v = 1,\dots,w;\ j = 1,\dots,n \tag{7}$$

This yields *3n* constraints. This constraint is redundant with Constraint 1, and will not be included in the examples. However, this constraint could be important with modifications to the cost function and Constraint 1. For example, if more targets were available than could be attacked by the available number of vehicles, it would be physically impossible to complete all tasks on all targets. In that case, we could make the cost function the value of targets killed, and we would not include Constraint 1, but would then need Constraint 3, to prevent unnecessary duplicate attacks on a single target.

3. An AV *v*, coming from the outside, can visit target *j* at most once:

$$\sum_{k=1}^{3}\sum_{i=1,i\neq j}^{n+w} x_{ij}^{(v,k)} \leq 1, \quad \begin{array}{l} v = 1,\dots,w \\ j = 1,\dots,n \end{array} \tag{8}$$

This, and Condition 4 below, eliminate the possibility of loops. In addition, each AV *v* can only enter the sink once:

$$\sum_{i=1}^{n+w} x_{i,n+w+1}^{(v)} \leq 1, \quad v = 1,\dots,w \tag{9}$$

This yields *(n+1)w* constraints.

4. AV *v* leaves node *j* at most once:

$$\sum_{k=1}^{3}\sum_{i=1,i\neq j}^{n} x_{ij}^{(v,k)} + x_{j.n+w+1}^{(v)} \leq 1, \quad v = 1,\dots,w; \tag{10}$$

where $j=1,\dots,n$. This yields *nw* constraints.

5. A munition is perishable. Thus, an AV *v* can be assigned to attack at most one target.

$$\sum_{j=1}^{n}\sum_{i=1}^{n+w} x_{ij}^{(v,2)} \leq 1, \quad \forall v = 1,\dots,w. \tag{11}$$

This yields *w* constraint equations.

6. If AV *v* is assigned to fly to target *j* for Verification, it cannot possibly be assigned to attack target *j*:

$$\sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,2)} \leq 1 - \sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,3)}, \quad v = 1,\dots,w; \tag{12}$$

where $j = 1,\dots,n$. Condition 3 renders Condition 6 redundant; we do however include Condition 6, because it holds in its own right, but also in the case where one would choose not to have recourse to the assumption which yields Condition 3.

7. Continuity Constraints. These Constraints ensure that proper flow balance is maintained at each node.
   7.1 If AV *v* enters target (node) j for the purpose of performing task 3, it must also exit target *j*:

$$\sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,3)} \leq \sum_{k=1}^{3}\sum_{i=1,i\neq j}^{n} x_{j,i}^{(v,k)} + x_{j,n+w+1}^{(v)}, \tag{13}$$

where $j=1,\ldots,n;\ v=1,\ldots,w$.

7.2 If AV $v$ enters target (node) $j$ for the purpose of performing task 1, it must also "exit" target (node) $j$:

$$\sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,1)} \leq \sum_{k=1}^{3}\sum_{i=1,i\neq j}^{n} x_{j,i}^{(v,k)} + x_{j,j}^{(v,2)} + x_{j,n+w+1}^{(v)}, \qquad \text{where } j=1,\ldots,n;\ v=1,\ldots,w. \tag{14}$$

7.3. A munition is perishable. Thus, if AV $v$ is assigned to fly to target (node) $j$ to perform task $k=2$, then, at any other point in time, AV $v$ cannot also be assigned to fly from target $j$ to a target $i$, $i \neq j$, to perform any other tastk at target $i$; recall that according to our Simplifying Assumption, AV $v$ can enter target $j$ not more than once. Thus

$$\sum_{k=1}^{3}\sum_{i=1,i\neq j}^{n} x_{j,i}^{(v,k)} + x_{j.n+w+1}^{(v)} \leq 1 - \sum_{i=1}^{n+w} x_{i,j}^{(v,2)}, \tag{15}$$

where $j=1,\ldots,n;\ v=1,\ldots,w$.

7.4. If AV $v$ is not assigned to visit node $j$, then it cannot possibly be assigned to fly out of node $j$. Thus

$$\sum_{k=1}^{3}\sum_{i=1,i\neq j}^{n} x_{j,i}^{(v,k)} + x_{j.n+w+1}^{(v)} \leq \sum_{k=1}^{3}\sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,k)}, \qquad \text{where } j=1,\ldots,n;\ v=1,\ldots,w. \tag{16}$$

7.5. All AVs leave the source nodes. An AV leaves the source node even if this entails a direct assignment to the sink.

$$\sum_{k=1}^{3}\sum_{j=1}^{n} x_{n+v,j}^{(v,k)} + x_{n+v.n+w+1}^{(v)} = 1, \quad \forall v = 1,\ldots,w. \tag{17}$$

7.6. An AV cannot attack target (node) $i$, coming from target (node) $i$, unless it entered target (node) $i$ to perform a classification. Thus

$$x_{i,i}^{(v,2)} \leq \sum_{j=1}^{n+w} x_{j,i}^{(v,1)}, \forall \begin{array}{l} i=1,\ldots,n \\ v=1,\ldots,w \end{array}. \tag{18}$$

8. Timing Constraints

Nonlinear equations which enforce the timing constraints are easily derived, and are given in (Schumacher *et al* 2003[c]). We are however interested in an alternative, elegant formulation which uses linear inequalities, so that a MILP formulation is achieved.

Thus, let

$$T \equiv \max_v \{T_v\}_{v=1}^{w}. \tag{19}$$

Then the linear timing constraints become:

$$t_j^{(k)} \leq t_i^{(1)} + t_{i,j}^{(v,k)} + \left(2 - x_{i,j}^{(v,k)} - \sum_{l=1,l\neq i}^{n+w} x_{l,i}^{(v,1)}\right) wT \tag{20}$$

$$t_j^{(k)} \geq t_i^{(1)} + t_{i,j}^{(v,k)} - \left(2 - x_{i,j}^{(v,k)} - \sum_{l=1,l\neq i}^{n+w} x_{l,i}^{(v,1)}\right) wT \tag{21}$$

$$t_j^{(k)} \leq t_i^{(3)} + t_{i,j}^{(v,k)} + \left(2 - x_{i,j}^{(v,k)} - \sum_{l=1,l\neq i}^{n+w} x_{l,i}^{(v,3)}\right) wT \tag{22}$$

$$t_j^{(k)} \geq t_i^{(3)} + t_{i,j}^{(v,k)} - \left(2 - x_{i,j}^{(v,k)} - \sum_{l=1,l\neq i}^{n+w} x_{l,i}^{(v,3)}\right) wT \tag{23}$$

for $i=1,\ldots,n;\ j=1,\ldots,n;\ i \neq j;\ v=1,\ldots w;\ k=1,3$. In addition,

$$t_j^{(2)} \le t_i^{(1)} + t_{i,j}^{(v,2)} + \left(2 - x_{i,j}^{(v,2)} - \sum_{l=1,l\ne i}^{n+w} x_{l,i}^{(v,1)}\right)wT \tag{24}$$

$$t_j^{(2)} \ge t_i^{1} + t_{i,j}^{(v,2)} - \left(2 - x_{i,j}^{(v,2)} - \sum_{l=1,l\ne i}^{n+w} x_{l,i}^{(v,1)}\right)wT \tag{25}$$

$$t_j^{(2)} \le t_i^{(3)} + t_{i,j}^{(v,2)} + \left(2 - x_{i,j}^{(v,2)} - \sum_{l=1,l\ne i}^{n+w} x_{l,i}^{(v,3)}\right)wT \tag{26}$$

$$t_j^{(2)} \ge t_i^{(3)} + t_{i,j}^{(v,2)} - \left(2 - x_{i,j}^{(v,2)} - \sum_{l=1,l\ne i}^{n+w} x_{l,i}^{(v,3)}\right)wT \tag{27}$$

for $i=1,\ldots,n;\ j=1,\ldots,n;\ i \ne j;\ v=1,\ldots w$.

Also,

$$t_j^{(k)} \le t_v + t_{n+v,j}^{(v,k)} + \left(1 - x_{n+v,j}^{(v,k)}\right)wT \tag{28}$$

$$t_j^{(k)} \ge t_v + t_{n+v,j}^{(v,k)} - \left(1 - x_{n+v,j}^{(v,k)}\right)wT \tag{29}$$

for all $j=1,\ldots,n;\ k = 1,2,3;\ v = 1,\ldots,w$.

These timing constraints operate in pairs. They are loose inequalities which do not come into play for assignments $x_{i,j}^{(v,k)}$ which do not occur, but effectively become hard equality constraints for assignments which do occur. Thus the time that a task *k* is performed on target *j* by AV *v* will be equal to the time that the preceding task was performed by AV *v* at node *i*, plus the time it will take AV *v* to fly from node *i* to node *j*. A similar constraint applies if AV *v* left its source node *n+v* to fly to node *j*.

Furthermore,

$$t_j^{(1)} \le t_j^{(2)},\ j = 1,\ldots,\ n \tag{30}$$

$$t_j^{(2)} < t_j^{(3)},\ j = 1,\ldots,n \tag{31}$$

The timing constraints thus add $2n[(6n-1)w+1]$ linear inequality constraints. The timing constraints (20)-(31) are critical for the MILP formulation of the optimization problem at hand.

**Extensions**

Additional constraints can be included.

9. A vehicle's assigned path duration cannot be longer than its endurance $T_v$:

$$\sum_{k=1}^{3}\sum_{i=1}^{n+w}\sum_{j=1,j\ne i}^{n} t_{i,j}^{(v,k)} x_{i,j}^{(v,k)} \le T_v, \qquad v = 1,\ldots,w \tag{32}$$

This yields *w* constraints.

10. It is fairly easy to specify additional rigid time window constraints akin to the VRP, e.g., for time critical targets one could demand that the attack on target *j* take place after time $\underline{t}_j^{(2)}$, and not before time $\bar{t}_j^{(2)}$, i.e.

$$\underline{t}_j^{(2)} \le t_j^{(2)} \le \bar{t}_j^{(2)}, \qquad j = 1,\ldots,n \tag{33}$$

11. Numerous other constraints can also be included, such as: specific vehicles performing certain tasks, minimum time delays between tasks, simultaneous completion of attack tasks, and requiring the vehicle that classifies a target to also attack it. Logical constraints are easily included. With some constraints included, such as vehicle endurance (Constraint 9), the existence of a solution is no longer guaranteed.
12. Heterogeneous vehicles: For some applications, a set of heterogeneous vehicles would be used, with different capabilities. Some might be sensor platforms with no attack capability. Or some vehicles might simply have used all their ordinance, or not be carrying the proper ordinance to attack certain targets. In

such cases, we add the constraint $x_{i,j}^{(v,k)} = 0$ for any combination where vehicle *v* cannot perform task *k* on target *j*.

13. Partially prosecuted targets: If this algorithm was used for task assignment by a group of UAV's, additional targets and tasks could be added to the overall task list while some previously-known targets were already partly prosecuted. In this case, less than three tasks would be required for some targets, when the assignments were recalculated. For already completed tasks, we modify Constraint 1 such that

$$\sum_{v=1}^{w} \sum_{i=1, i \neq j}^{n+w} x_{ij}^{(v,k)} = 0, \quad \begin{matrix} k = 1,3 \\ j = 1,\dots,n \end{matrix}, \tag{34}$$

and

$$\sum_{v=1}^{w} \sum_{i=1}^{n+w} x_{ij}^{(v,2)} = 0, \quad j = 1,\dots,n \tag{35}$$

for any target *j* and task *k* that have already been completed.

## Examples

### One Target and Three UAVs

We first consider the case of one target and three AVs, i.e. *n=1* and *w=3*, as the problem is small enough to be described in detail and allow the reader to follow the mechanics of MILP.

We have 18 binary decision variables and 6 continuous decision variables. Minimizing the time the final task occurs will add an additional continuous decision variable *t*, for a total of 25 decision variables. In this single-target case, we could exclude the additional variable and simply minimize $t_1^3$, but the additional variable will be included to demonstrate the additional variable which would be required for $n \geq 2$. The State Transition Diagram is given in Figure 1.

There are 18 binary decision variables:

$$\begin{aligned}
(x_1,\dots,x_5) &= \left( x_{1,1}^{(1,2)}, x_{1,1}^{(2,2)}, x_{1,1}^{(3,2)}, x_{2,1}^{(1,1)}, x_{2,1}^{(1,2)} \right) \\
(x_6,\dots,x_{10}) &= \left( x_{2,1}^{(1,3)}, x_{3,1}^{(2,1)}, x_{3,1}^{(2,2)}, x_{3,1}^{(2,3)}, x_{4,1}^{(3,1)}, \right) \\
(x_{11},\dots,x_{15}) &= \left( x_{4,1}^{(3,2)}, x_{4,1}^{(3,3)}, x_{1,5}^{(1)}, x_{1,5}^{(2)}, x_{1,5}^{(3)} \right) \\
(x_{16}, x_{17}, x_{18}) &= \left( x_{2,5}^{(1)}, x_{3,5}^{(2)}, x_{4,6}^{(3)} \right)
\end{aligned} \tag{36}$$

There are 7 continuous decision variables:

$$(x_{19},\dots,x_{25}) = \left(t_1, t_2, t_3, t_1^{(1)}, t_1^{(2)}, t_1^{(3)}, t\right), \tag{37}$$

We wish to minimize

$$J = x_{25} + 0.1(x_{22} + x_{23} + x_{24}) \tag{38}$$

subject to the following constraints:

From Constraint 1:

$$\begin{aligned}
&x_4 + x_7 + x_{10} = 1 \\
&x_6 + x_9 + x_{12} = 1 \\
&x_5 + x_8 + x_{11} + x_1 + x_2 + x_3 = 1
\end{aligned} \tag{39}$$

From Constraint 7.5:

$$x_4 + x_5 + x_6 + x_{16} = 1$$
$$x_7 + x_8 + x_9 + x_{17} = 1 \tag{40}$$
$$x_{10} + x_{11} + x_{12} + x_{18} = 1$$

Thus we have 6 equality constraints, plus the following inequality constraints.
From Constraint 2:

$$x_4 + x_7 + x_{10} \le 1$$
$$x_6 + x_9 + x_{12} \le 1 \tag{41}$$
$$x_5 + x_8 + x_{11} + x_1 + x_2 + x_3 \le 1$$

From Constraint 3:

$$x_4 + x_5 + x_6 \le 1$$
$$x_7 + x_8 + x_9 \le 1 \tag{42}$$
$$x_{10} + x_{11} + x_{12} \le 1$$

Constraints 4-6 drop out in the 1-target case.
The Continuity Constraints give:
From 7.1:

$$x_6 \le x_{13}$$
$$x_9 \le x_{14} \tag{43}$$
$$x_{12} \le x_{15}$$

From 7.2:

$$x_4 \le x_1 + x_{13}$$
$$x_7 \le x_2 + x_{14} \tag{44}$$
$$x_{10} \le x_3 + x_{15}$$

From 7.3:

$$x_{13} + x_1 + x_5 \le 1$$
$$x_{14} + x_2 + x_8 \le 1 \tag{45}$$
$$x_{15} + x_3 + x_{11} \le 1$$

From 7.4:

$$x_{13} \le x_4 + x_6$$
$$x_{14} \le x_7 + x_9 \tag{46}$$
$$x_{15} \le x_{10} + x_{12}$$

From 7.5:
The equality constraints given by Eq. 36.

From 7.6:

$$x_1 \le x_4$$
$$x_2 \le x_7 \tag{47}$$
$$x_3 \le x_{10}$$

With only 1 target node, the Constraints associated with Eq (20-23) and (26,27) are not meaningful. So we are left with the following timing constraints:

From Eq (24,25):

$$
\begin{aligned}
&x_{23} \le x_{22} + t_{1,1}^{(1,2)} + (2 - x_1 - x_4)wT \\
&x_{23} \ge x_{22} + t_{1,1}^{(1,2)} - (2 - x_1 - x_4)wT \\
&x_{23} \le x_{22} + t_{1,1}^{(2,2)} + (2 - x_2 - x_7)wT \\
&x_{23} \ge x_{22} + t_{1,1}^{(2,2)} - (2 - x_2 - x_7)wT \\
&x_{23} \le x_{22} + t_{1,1}^{(3,2)} + (2 - x_3 - x_{10})wT \\
&x_{23} \ge x_{22} + t_{1,1}^{(3,2)} - (2 - x_3 - x_{10})wT
\end{aligned} \tag{48}
$$

From Eq (28,29):

$$
\begin{aligned}
&x_{22} \le x_{19} + t_{2,1}^{(1,1)} + (1 - x_4)wT \\
&x_{22} \ge x_{19} + t_{2,1}^{(1,1)} - (1 - x_4)wT \\
&x_{22} \le x_{20} + t_{3,1}^{(2,1)} + (1 - x_7)wT \\
&x_{22} \ge x_{20} + t_{3,1}^{(2,1)} - (1 - x_7)wT \\
&x_{22} \le x_{21} + t_{4,1}^{(3,1)} + (1 - x_{10})wT \\
&x_{22} \ge x_{21} + t_{4,1}^{(3,1)} - (1 - x_{10})wT
\end{aligned} \tag{49}
$$

and

$$
\begin{aligned}
&x_{23} \le x_{19} + t_{2,1}^{(1,2)} + (1 - x_5)wT \\
&x_{23} \ge x_{19} + t_{2,1}^{(1,2)} - (1 - x_5)wT \\
&x_{23} \le x_{20} + t_{3,1}^{(2,2)} + (1 - x_8)wT \\
&x_{23} \ge x_{20} + t_{3,1}^{(2,2)} - (1 - x_8)wT \\
&x_{23} \le x_{21} + t_{4,1}^{(3,2)} + (1 - x_{11})wT \\
&x_{22} \ge x_{21} + t_{4,1}^{(3,2)} - (1 - x_{11})wT
\end{aligned} \tag{50}
$$

and finally,

$$
\begin{aligned}
&x_{24} \le x_{19} + t_{2,1}^{(1,3)} + (1 - x_6)wT \\
&x_{24} \ge x_{19} + t_{2,1}^{(1,3)} - (1 - x_6)wT \\
&x_{24} \le x_{20} + t_{3,1}^{(2,3)} + (1 - x_9)wT \\
&x_{24} \ge x_{20} + t_{3,1}^{(2,3)} - (1 - x_9)wT
\end{aligned} \tag{51}
$$

$$x_{24} \le x_{21} + t_{4,1}^{(3,3)} + (1 - x_{12})wT$$
$$x_{24} \ge x_{21} + t_{4,1}^{(3,3)} - (1 - x_{12})wT$$

Also, from Eq (30,31), we have:

$$x_{22} \le x_{23} - \varepsilon \qquad (52)$$

and

$$x_{23} \le x_{24} - \varepsilon \qquad (53)$$

where $\varepsilon > 0$ is a small positive constant. We will set $\varepsilon = 0.1$. This enforces a small delay between each task being performed on a target.

Finally, from Eq (2), we have

$$x_{24} \le x_{25}. \qquad (54)$$

Thus the full set of constraints contains 6 equality constraints and 51 inequality constraints, for 57 total constraints. A few of them are redundant for this case, but might not be for a more complex problem.

Let us make the simplifying assumption that the time to travel from node *i* to node *j* to perform task *k* is independent of which task is required, and which vehicle is performing the task. Then $t_{i,j}^{(v,k)}$ simply becomes $t_{i,j}$. For this example, let

$t_{1,1} = 0.1$
$t_{2,1} = 3.61$
$t_{3,1} = 4.24$
$t_{4,1} = 5.39$

We will set *T =Tv= 100* as the endurance of all of the AVs, so that endurance is not a constraint, and feasibility is guaranteed. Then the optimal assignment is:

$x_i = 1,\ i=1,4,9,14,18$
$x_i = 0,\ i=2,3,5,\ldots,8,10,\ldots,13,15,16,17$
$x_i = 0,\ i=19,20,21.$
$x_{22} = 3.61$
$x_{23} = 3.71$
$x_{24} = 4.24$
$x_{25} = 4.24$

This corresponds with all 3 vehicles immediately leaving their source nodes ($x_{19}$-$x_{21}=0$), and vehicle 1 performing classify and attack on the target at $t=3.61$ and $3.71$ respectively, with vehicle 2 performing verification at $T=4.24$. Vehicle 3 flies direction to the sink (it is not assigned to this target, but continues to search).

Suppose that it takes longer for a vehicle that has just classified a target to complete an attack on that target. Then we might have the initial conditions

$t_{1,1} = 1$
$t_{2,1} = 3.61$
$t_{3,1} = 4.24$
$t_{4,1} = 5.39$

In this case, the assignment is identical, except that the attack occurs at $t=4.61$and the verification at $t=4.71$. This is an example where the verification had to be delayed so that it occurred after the attack.

Finally, suppose that Vehicle 3 is closer to the target initially, and we have the initial conditions

$t_{1,1} = 1$

$t_{2,1} = 3.61$

$t_{3,1} = 4.24$

$t_{4,1} = 4.50$

Then the optimal assignment is:
$x_i = 1,\ i=4,8,12,13,15$
$x_i = 0,\ i=1,2,3,56,7,9,10,11,14,16,17,18$
$x_i = 0,\ i=19,20,21.$
$x_{22} = 3.61$
$x_{23} = 4.24$
$x_{24} = 4.50$
$x_{25} = 4.50$

This assignment requires all 3 vehicles to immediately leave their source nodes and proceed to the target . Vehicle 1 performed the classification, vehicle 2 the attack, and vehicle 3 the verification. WASMs 1 and 3 then proceed to the sink, in other words, continue to search for other targets.

**MILP Size**

For *n* targets, *w* vehicles, and *m=3* tasks per target, the problem size scales as follows: There are *n(n-1)wm+nwm+2nw+mn+2w+1* decision variables. Of these, *3+nm+1* are continuous timing variables, and the rest are binary decision variables. The number of constraints likewise grows rapidly. There are *12*(n-1)*n*w+9*n*w+2*n*w*m+2*n*m+3*w* constraints. Of these, *m*n+w* are equality constraints. The rest are inequality constraints, including *7*n*w+2*w* inequality non-timing constraints, and *12*(n-1)*n*w+2*n*w+2*n*w*m+m*n* inequality timing constraints.

The size of the MILP expands rapidly as problem size increases. However, some practically-sized problems are amenable to optimal solution with this mixed-integer linear program formulation. For n=2, w=3, there are 51 binary decision variables, 10 continuous decision variables, 9 linear equality constraints, and 174 linear inequality constraints. For n=2, w=4, there are 68 binary decision variables, 11 continuous decision variables, 10 linear equality constraints, and 230 linear inequality constraints. For n=2, w=5, there are 85 binary decision variables, 12 continuous decision variables, 11 linear equality constraints, and 286 linear inequality constraints. Problem size and complexity grow much more rapidly with an increased number of targets. For n=3, w=4, there are 136 binary decision variables, 14 continuous decision variables, 13 linear equality constraints, and 485 linear inequality constraints. The growth of constraints and variables is linear in the number of vehicles, but quadratic in the number of targets. For some operational scenarios involving larger problem sizes, optimal solutions may not be found within the available computation time. However, useful feasible solutions satisfying all constraints can still be found (ILOG 1999).

**Two Tasks per Target**

A similar MILP formulation with the classification and attack tasks combined as a single task, resulting in two tasks per target, was presented in [14]. The data used for an interesting example of an assignment and scheduling computation with two tasks per target is given below. The data for the MILP problem is specified by the distances, or flight times $T_{ij}$, between the nodes. It is here tacitly assumed that the listed times are not vehicle and/or task dependent. For the following examples, let $T_{ij}$ be specified as:

$$T_{i,j} = \begin{bmatrix} 0 & 5.8310 & 7.0711 & 7.2801 \\ 5.8310 & 0 & 7.2111 & 5.0000 \\ 7.0711 & 7.2111 & 0 & 3.0000 \\ 7.2801 & 5.0000 & 3.0000 & 0 \\ 9.2195 & 3.6056 & 8.0623 & 5.0990 \\ 3.1623 & 8.4853 & 10.0000 & 10.4403 \\ 10.0000 & 8.6023 & 3.1623 & 3.6056 \\ 4.4721 & 5.0990 & 10.2956 & 9.2195 \\ 8.2462 & 5.8310 & 12.7279 & 10.8167 \end{bmatrix}$$

where the start node *i* is indexed down the rows, and the end node *j* is indexed over the columns. So the time for a vehicle to fly from node 4 to node 3, $T_{43} = 3.0$, and so on. The sink position does not matter, as it only exists conceptually - the time to reach it is not meaningful in the optimization problem under consideration. In practice, a vehicle that is assigned to the sink continues to search for potential targets along a predefined search path.

In this case, an assignment $x_{ii}^{(v,k)} = 1$ is not possible for any *v* or *k*, as that would require a vehicle to both attack and verify the same target, which is not possible in the WASM scenario, where the vehicle is used up when performing an attack. The solution for this example scenario is given in Figure 3. This solution illustrates interesting behavior, with Vehicle 5 (orange) performing three verification tasks, and Vehicle 4 performing a verification, followed by a classify/attack task. Larger problems will be more computationally difficult to solve. However, any problem physically feasibly for a typical WASM team of four vehicles (i.e. with three targets or less) can be solved quickly enough for on-line implementation. A more detailed discussion of problem size and computation time as a function of scenario parameters can be found in (Schumacher et al. 2004[b]).

**Path Planning and Task Assignment Coupling**

So far we assumed that the path length from node *i* to node *j* is known ahead of time and we focused on the task assignment problem. Artificially decoupling the path planning and task assignment aspects of the UAV cooperative control problem can result in a lack of feasible solutions. Including coupling between these aspects of the problem is thus essential. We accommodate this critical coupling in our formulation by allowing the variables $t_v$ to represent a delay before a vehicle leaves its source node. This can also be thought of as adding loiter to a path, or otherwise extending the path length taken to perform the vehicle's first task.

In the example presented earlier, we assumed that all tasks required the same amount of time for a vehicle to accomplish, given identical starting and ending points. To illustrate an example that demonstrates infeasibility, we will need to discuss more precisely differences in path planning based on task, for which our formulation allows. We will examine the path planning and task assignment for the WASM scenario studied in (Schumacher *et al*, 2002, Schumacher *et al*, 2003[a], Schumacher *et al*, 2003[b]). For this scenario, classify, attack, and verification paths all have different constraints. Classify paths must approach a target from one of the preferred Automatic Target Recognition (ATR) aspect angles (Chandler and Pachter, 2001), and the task is completed at the proper sensor standoff distance, not when the vehicle is directly over the target. Attack paths can approach the target from any angle, and are completed when the vehicle is directly over the target. Verification paths can approach from any angle, and are completed when the vehicle is at the sensor standoff distance from the target. If path planning is decoupled from task assignment, and minimum length paths are calculated for all of the tasks, then, for some initial conditions, there will be no feasible assignment that completes all of the required tasks, using the calculated paths.

This is illustrated in Figure 4. Attack and Verification paths are illustrated as straight, dashed lines connecting the vehicles and the targets. Due to the desired approach angles, classification paths (presented as solid lines), form two sides of a triangle. Turning radius is neglected in the illustration. Although turning radius is important for calculating the true paths, neglecting it does not change the fundamental nature of the problem. As Figure 4 illustrates, for any combination of target node *j* and vehicle start node *i*, the minimum length path for classification is longer than the minimum length path for verification. Clearly, if the vehicles are collocated at a point $(x_1,y_1)$, and the targets are collocated at another point $(x_2,y_2)$, then all paths for a particular task *k* will be the identical, irrespective of target node *j* or vehicle start node *i*, and all of the "classify" paths will be longer than all of the "verify" paths. If the targets and vehicles are then allowed to spread out from ($x_1$,$y_1$) and ($x_2$,$y_2$), then, as long as the vehicles and targets each remain within some small region of each other, all of the "classify" paths will still be longer than all of the "verify" paths. For some values of the turn radius and sensor standoff distance, all minimum-distance "classify" paths will also be longer than all minimum-distance "attack" paths. For nearly-collocated vehicles, a similar difficulty arises when comparing minimum-distance "attack" and "verify" paths. The paths are identical, but the "verifies" would occur first, because of the sensor standoff distance. The results of the attacks on the targets would be tested before the attacks had occurred, which is obviously unacceptable.

In Figure 4, if the vehicles can only choose among the illustrated paths, plus additional paths connecting the two target locations at the ends of the initial paths, there is no assignment possible that will classify, attack, and verify both targets in the appropriate order. No matter which two vehicles are assigned the classify and attack tasks, the third vehicle's minimum-length verify paths will arrive before the vehicle is classified or attacked. Clearly, the inclusion of loiter to delay the verification task is required.

Suppose that the nominal vehicle and target data are:

$$T_{i,j} = \begin{bmatrix} 0 & 2.0 \\ 2.0 & 0 \\ 5.0 & 5.4 \\ 5.1 & 5.1 \\ 5.4 & 5.0 \end{bmatrix}$$

for "verify" tasks, but with the additional task-dependent variations that classifies take 2.0 longer due to flying smoothed triangular paths, and attacks take 0.4 longer due to covering the additional sensor standoff distance. In this case, all of the minimum-distance verification paths are shorter than the minimum-distance attack paths, which are in turn shorter than all of the minimum-distance classification paths. Using only the minimum-distance paths calculated in a decoupled manner, there is no feasible assignment. Using our MILP algorithm to perform the assignment allowing loiter, we now find the optimal assignment schedule:

Vehicle 1 leaves its start node immediately, classifying Target 1 at $t=7.0$, and attacking Target 1 at $t=7.4$. Vehicle 3 leaves its start node immediately, classifying Target 2 at $t=7.0$, and attacking Target 2 at $t=7.4$. Vehicle 2 does not leave its start node immediately, but loiters for $t=2.4$, before then proceeding to verify Target 1 at $t=7.5$, and then verifying Target 2 at $t=9.5$. The MILP algorithm presented here has been implemented into the MultiUAV high fidelity UAV operation simulation and visualization package, thus endowing the UAV teams with higher "intelligence" [http://www.isr.us/research_sim_muav.asp accessed 6 December 2005].

## Conclusions

This paper presents a Mixed Integer Linear Program (MILP) formulation for finding the optimal solution to a difficult multiple-task assignment and scheduling problem where the tasks are coupled by timing and task precedence constraints. Preserving linearity and casting this scheduling aspects - dominated combinatorial optimization problem into a MILP formulation, without having recourse to direct enumeration or time discretization, is a major contribution of this paper. This formulation allows staged vehicle departure times to guarantee that timing constraints are satisfied, and directly incorporates the varying task completion times into the optimization. Optimally staged departures bring about a guarantee of feasibility, without a need to loiter. Indeed, feasibility is guaranteed, provided that the number of UAVs employed exceeds the number of targets, and the UAV endurance is sufficiently high. Moreover, the formulation is flexible enough to allow for various alternative performance functionals, e.g., minimum time to mission completion, minimum path length traveled by the air vehicles, or minimum number of air vehicles required to accomplish the mission. This is a rigorous formulation, which allows a true optimal solution for a very challenging assignment and scheduling problem. Solution results were presented for practical problem sizes, with real-time implementable optimal solutions obtainable for many realistic problems. Our exact solution could serve as a test bed for heuristic methods currently employed for air vehicles operations optimization.

## References


Chandler PR and Pachter M (2001). Hierarchical control of autonomous teams. In: *Proceedings of the AIAA Guidance, Navigation, and Control Conference 2001*. American Institute of Aeronautics and Astronautics, Reston, VA, USA. AIAA Paper No 2001-4149.

Gendreau M, Hertz A and Laporte G (1994). A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science* **40**: 1276-1289.

Glover F and Laguna M (1997). *Tabu Search*. Kluwer Academic Publishers: Boston, MA, USA.

Hooker TN (1995). Testing heuristics – we have it all wrong. *Journal of Heuristics* **1**: 33-44.

ILOG CPLEX User's Guide, ILOG 1999.

Nemhauser GL and Wolsey LA (1988). *Integer and Combinatorial Optimization*. Wiley: New York, NY, USA.

O'Rourke KP, Bailey TG, Hill R and Carlton WB (2000). Dynamic routing of unmanned aerial vehicles using reactive Tabu Search. *Military Operations Research Journal* **6**: 33-42.

Pinedo M (2002). *Scheduling.* Prentice Hall: Upper Saddle, NJ, USA.

Schumacher C, Chandler PR and Rasmussen SJ (2002). Task allocation for wide area search munitions via iterative network flow. In: *Proceedings of the Guidance, Navigation, and Control Conference 2002*. American Institute of Aeronautics and Astronautics: Reston, VA, USA. AIAA Paper No 2002-4586.

Schumacher C, Chandler PR and Rasmussen SJ, (2003). Task allocation for wide area search munitions with variable path length. In: *Proceedings of the American Control Conference 2003*. Institute of Electrical and Electronic Engineers: New York, NY, USA.

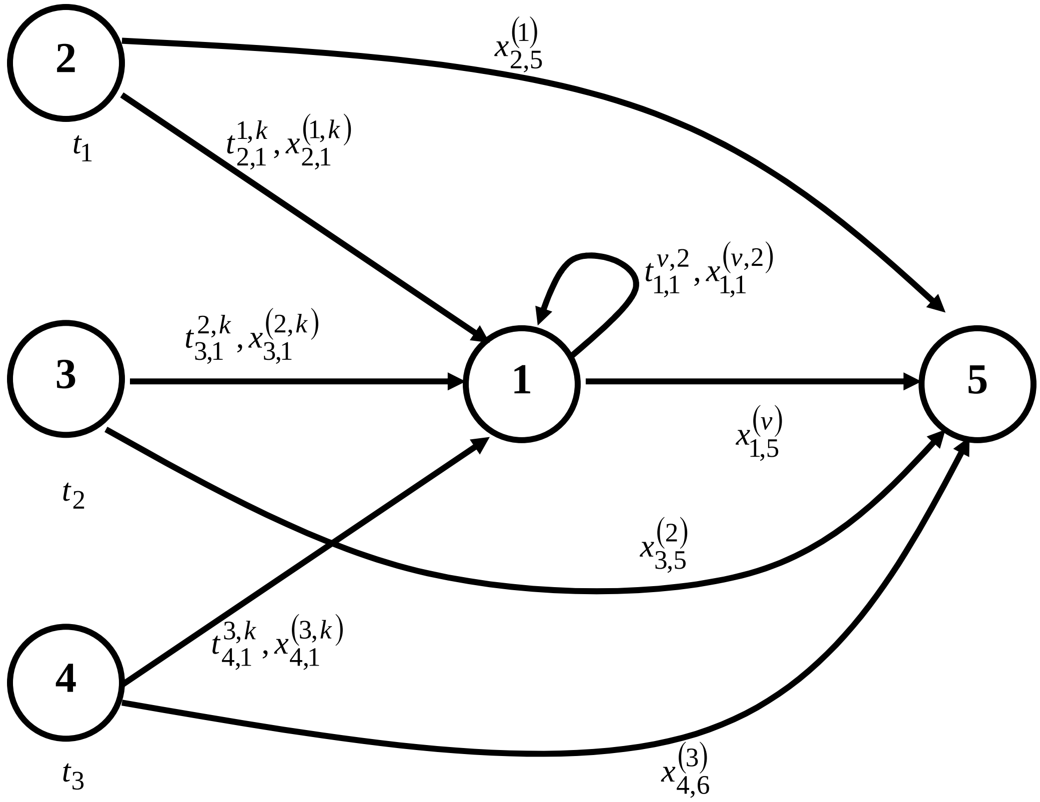
Schumacher C, Chandler PR, Rasmussen SJ, (2003). Path elongation for task assignment. In: *Proceedings of the AIAA Guidance, Navigation, and Control Conference 2003*. American Institute of Aeronautics and Astronautics: Reston, VA, USA. AIAA Paper No 2003-5585.

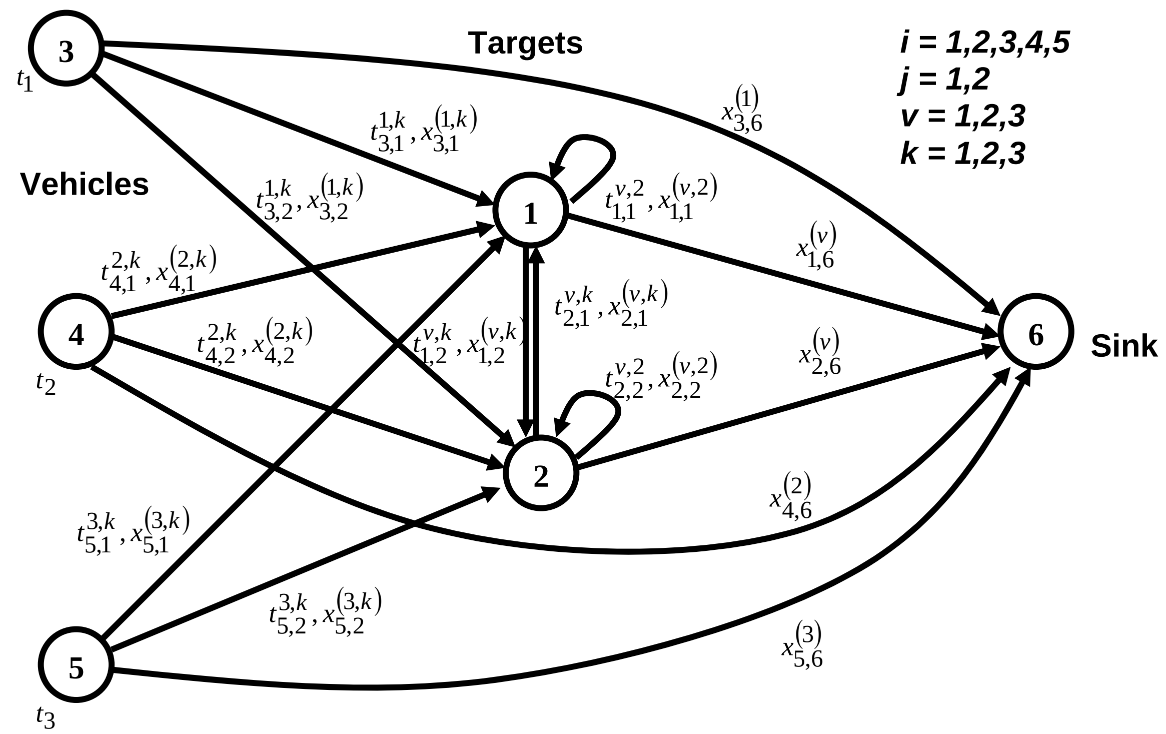
Schumacher C, Pachter M, Pachter LS and Chandler P (2003). UAV task assignment with timing constraints. In: *Proceedings of the AIAA Guidance, Navigation, and Control Conference 2003*. American Institute of Aeronautics and Astronautics: Reston, VA, USA. AIAA Paper No 2003-5664.

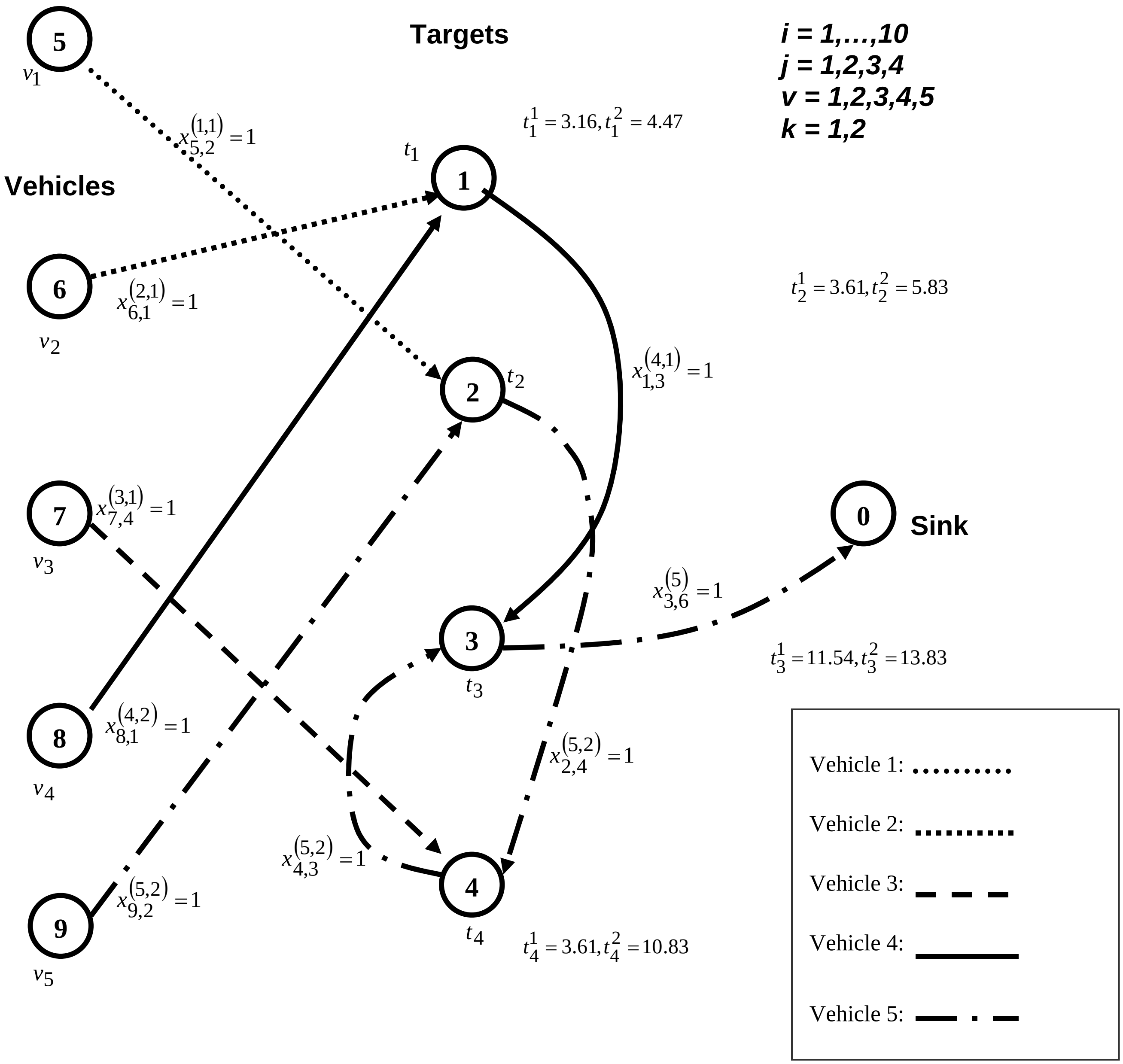
Schumacher C, Pachter M, Pachter LS and Chandler P (2004). Constrained optimization for UAV task assignment. In: *Proceedings of the AIAA Guidance, Navigation, and Control Conference 2004.* American Institute of Aeronautics and Astronautics: Reston, VA, USA. AIAA Paper No 2004-5352.

Schumacher C, Chandler P, Pachter M and Pachter LS (2004). UAV task assignment with timing constraints via Mixed-Integer Linear Programming. In: *Proceedings of the 3rd Unmanned Unlimited Technical Conference 2004*. American Institute of Aeronautics and Astronautics: Reston, VA, USA. AIAA Paper No 2004-6410.

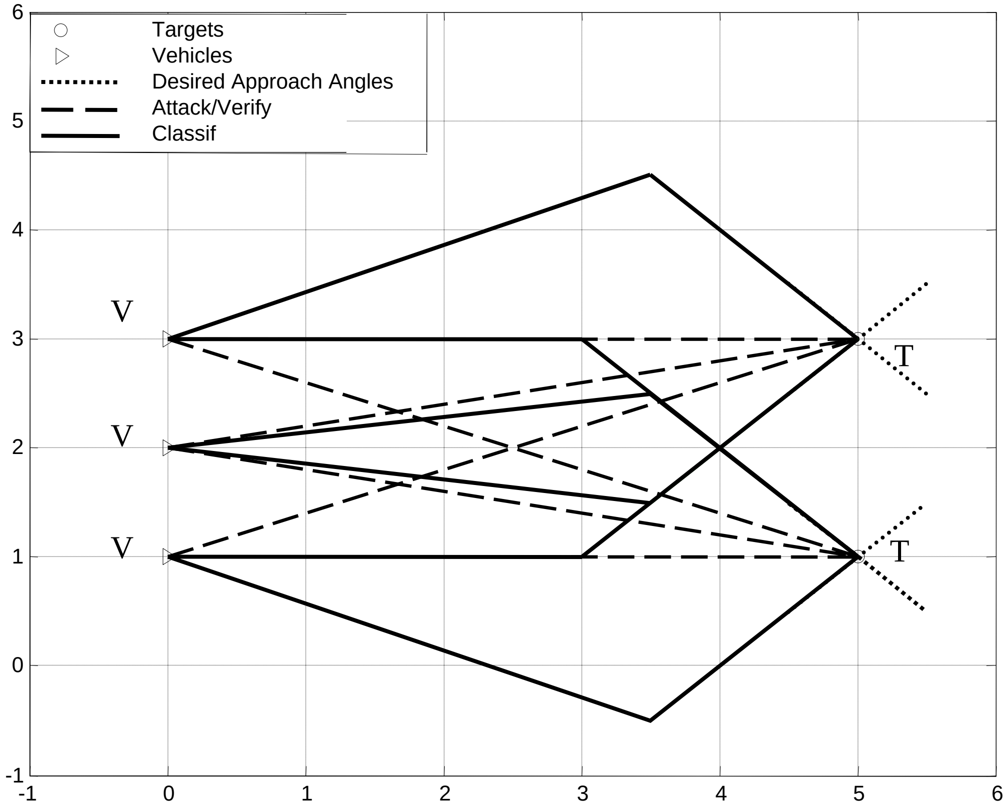
Toth P and Vigo D (2002). *The Vehicle Routing Problem*. SIAM: Philadelphia, PA, USA.


2
$t_1$
$x_{2,5}^{(1)}$
$t_{2,1}^{1,k}, x_{2,1}^{(1,k)}$
$t_{1,1}^{v,2}, x_{1,1}^{(v,2)}$
3
$t_{3,1}^{2,k}, x_{3,1}^{(2,k)}$
1
5
$x_{1,5}^{(v)}$
$t_2$
$x_{3,5}^{(2)}$
4
$t_{4,1}^{3,k}, x_{4,1}^{(3,k)}$
$t_3$
$x_{4,6}^{(3)}$


Targets
i = 1,2,3,4,5
j = 1,2
v = 1,2,3
k = 1,2,3
Vehicles
Sink
3
4
5
1
2
6
$t_1$
$t_2$
$t_3$
$t_{3,1}^{1,k}, x_{3,1}^{(1,k)}$
$t_{3,2}^{1,k}, x_{3,2}^{(1,k)}$
$t_{4,1}^{2,k}, x_{4,1}^{(2,k)}$
$t_{4,2}^{2,k}, x_{4,2}^{(2,k)}$
$t_{5,1}^{3,k}, x_{5,1}^{(3,k)}$
$t_{5,2}^{3,k}, x_{5,2}^{(3,k)}$
$t_{1,1}^{v,2}, x_{1,1}^{(v,2)}$
$t_{2,1}^{v,k}, x_{2,1}^{(v,k)}$
$t_{1,2}^{v,k}, x_{1,2}^{(v,k)}$
$t_{2,2}^{v,2}, x_{2,2}^{(v,2)}$
$x_{3,6}^{(1)}$
$x_{1,6}^{(v)}$
$x_{2,6}^{(v)}$
$x_{4,6}^{(2)}$
$x_{5,6}^{(3)}$



5
$v_1$
Targets
$i = 1,\ldots,10$
$j = 1,2,3,4$
$v = 1,2,3,4,5$
$k = 1,2$
$x_{5,2}^{(1,1)} = 1$
$t_1^1 = 3.16, t_1^2 = 4.47$
$t_1$
1
Vehicles
6
$x_{6,1}^{(2,1)} = 1$
$v_2$
$t_2^1 = 3.61, t_2^2 = 5.83$
$x_{1,3}^{(4,1)} = 1$
2
$t_2$
7
$x_{7,4}^{(3,1)} = 1$
$v_3$
0
Sink
$x_{3,6}^{(5)} = 1$
3
$t_3$
$t_3^1 = 11.54, t_3^2 = 13.83$
8
$x_{8,1}^{(4,2)} = 1$
$v_4$
$x_{2,4}^{(5,2)} = 1$
Vehicle 1:
Vehicle 2:
Vehicle 3:
Vehicle 4:
Vehicle 5:
$x_{4,3}^{(5,2)} = 1$
4
$t_4$
9
$x_{9,2}^{(5,2)} = 1$
$v_5$
$t_4^1 = 3.61, t_4^2 = 10.83$



Targets
Vehicles
Desired Approach Angles
Attack/Verify
Classif
V
V
V
T
T
6
5
4
3
2
1
0
-1
-1
0
1
2
3
4
5
6

Figure 1 – State Transition Diagram for 1 Target, 3 Tasks per Target, 3 Vehicles

Figure 2 – State Transition Diagram for 2 Targets, 3 Tasks per Target, 3 Vehicles

Figure 3 – Task Assignments for $n=4$ Targets, 2 Tasks per Target, $w=5$ Vehicles

Figure 4 – Minimum-Time Paths with 2 Targets, 3 Tasks per Target, 3 Vehicles