AFRL-VA-WP-TP-2007-306

UAV SCHEDULING VIA THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS (PREPRINT)

Amanda L. Weinstein and Corey Schumacher

JANUARY 2007

Approved for public release; distribution unlimited.

STINFO COPY

AIR VEHICLES DIRECTORATE
AIR FORCE MATERIEL COMMAND
AIR FORCE RESEARCH LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

NOTICE AND SIGNATURE PAGE

AFRL-VA-WP-TP-2007-306 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//

AMANDA L. WEINSTEIN
UAV Operational Employment Analyst
Control Design and Analysis Branch
Air Force Research Laboratory
Air Vehicles Directorate

//Signature//

DEBORAH S. GRISMER
Chief
Control Design and Analysis Branch
Air Force Research Laboratory
Air Vehicles Directorate

//Signature//

Dr. JEFFREY C. TROMP
Senior Technical Advisor
Control Sciences Division
Air Vehicles Directorate

*Disseminated copies will show "//Signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*: January 2007

2. REPORT TYPE: Conference Paper Preprint

3. DATES COVERED *(From - To)*: 04/01/2006 – 08/01/2006

4. TITLE AND SUBTITLE: UAV SCHEDULING VIA THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS (PREPRINT)

5a. CONTRACT NUMBER: In-house

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER: 62201F

6. AUTHOR(S): Amanda L. Weinstein and Corey Schumacher

5d. PROJECT NUMBER: A03D

5e. TASK NUMBER

5f. WORK UNIT NUMBER: 0B

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Control Design and Analysis Branch (AFRL/VACA)
Control Sciences Division
Air Vehicles Directorate
Air Force Materiel Command, Air Force Research Laboratory
Wright-Patterson Air Force Base, OH 45433-7542

8. PERFORMING ORGANIZATION REPORT NUMBER: AFRL-VA-WP-TP-2007-306

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

10. SPONSORING/MONITORING AGENCY ACRONYM(S): AFRL-VA-WP

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S): AFRL-VA-WP-TP-2007-306

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

Conference paper submitted to the Proceedings of the 2007 AIAA Infotech Aerospace Conference and Exhibit. This is a work of the U.S. Government and is not subject to copyright protection in the United States. PAO Case Number: AFRL/WS 07-0121 (cleared January 18, 2007).

14. ABSTRACT

In an urban environment, multiple small unmanned aerial vehicles (UAVs) may be utilized to locate, surveil, or attack various targets. Whatever the task, the air vehicles must cooperate by efficiently communicating with each other and optimally assigning each UAV to the appropriate task at the appropriate target and at the appropriate time. In this paper, a vehicle assignment algorithm is developed using a mixed integer linear program (MILP) to find the global optimal scheduling solution. The MILP can accommodate both binary and continuous decision variables as well as a variety of constraints and objective functions; however, the NP-hard nature of the problem implies a dramatic increase in the computing complexity as the number of variables and constraints increase. This formulation accounts for an assortment of scenarios focused on the military necessity for precise intelligence, surveillance, and reconnaissance (ISR) by modifying the vehicle routing problem with time windows (VRPTW) formulation. The VRPTW is a type of capacitated vehicle routing problem which optimally assigns a designated number of delivery vehicles originating at a single depot to a known number of customers. Specifically, the VRPTW and network flow techniques account for various scenarios as well as operator imposed timing constraints such as precedence constraints. For example, certain targets may take precedence or require simultaneous arrival times where the targets are first hierarchically clustered according to their proximity to each other. Thus, this paper also focuses on methods of clustering targets and implements this information into the MILP to optimally assign UAVs to targets. Clustering targets that are near enough to alert each other of an attack will allow UAVs to recognize this potential and hence surveil these targets simultaneously to avoid early detection. This technique will also prevent targets from being further camouflaged or moved once alerted to a nearby attack. Finally, this paper will directly compare the computation times and solutions for the min makespan objective, the minimum total time objective, and the total distance minimization.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
Unclassified	Unclassified	Unclassified	SAR	20	Amanda L. Weinstein

19b. TELEPHONE NUMBER *(Include Area Code)*: N/A

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

UAV Scheduling via the Vehicle Routing Problem with Time Windows

Amanda L. Weinstein[1] and Corey Schumacher[2]
Air Force Research Laboratory (AFRL/VACA), Wright-Patterson AFB, OH, 45433-7531

In an urban environment, multiple small unmanned aerial vehicles (UAVs) may be utilized to locate, surveil, or attack various targets. Whatever the task, the air vehicles must cooperate by efficiently communicating with each other and optimally assigning each UAV to the appropriate task at the appropriate target and at the appropriate time. In this paper, a vehicle assignment algorithm is developed using a mixed integer linear program (MILP) to find the global optimal scheduling solution. The MILP can accommodate both binary and continuous decision variables as well as a variety of constraints and objective functions; however, the NP-hard nature of the problem implies a dramatic increase in the computing complexity as the number of variables and constraints increase. This formulation accounts for an assortment of scenarios focused on the military necessity for precise intelligence, surveillance, and reconnaissance (ISR) by modifying the vehicle routing problem with time windows (VRPTW) formulation. The VRPTW is a type of capacitated vehicle routing problem which optimally assigns a designated number of delivery vehicles originating at a single depot to a known number of customers. Specifically, the VRPTW and network flow techniques account for various scenarios as well as operator imposed timing constraints such as precedence constraints. For example, certain targets may take precedence or require simultaneous arrival times where the targets are first hierarchically clustered according to their proximity to each other. Thus, this paper also focuses on methods of clustering targets and implements this information into the MILP to optimally assign UAVs to targets. Clustering targets that are near enough to alert each other of an attack will allow UAVs to recognize this potential and hence surveil these targets simultaneously to avoid early detection. This technique will also prevent targets from being further camouflaged or moved once alerted to a nearby attack. Finally, this paper will directly compare the computation times and solutions for the min makespan objective, the minimum total time objective, and the total distance minimization.

Nomenclature

N	=	total number of targets (customers)
K	=	total number of UAVs (vehicles)
L	=	total number of launch sites
C	=	total number of landing sites
i	=	index for traveling from node i
j	=	index for traveling to node j
k	=	index for the vehicle
x_{ijk}	=	binary variable indicating whether UAV k traveled from node i to j
c_{ij}	=	distance from node i to j
t_{ijk}	=	travel time from node i to node j for vehicle k
t_i	=	arrival time of UAV at target i
t_{jk}	=	arrival time of UAV k at landing site j
s_{ik}	=	service time for node i for UAV k
r_k	=	max route time allowed for UAV k

[1] UAV Operational Employment Analyst, AFRL/VACA, 2210 8th Street Bldg 146 Room 300 WPAFB, OH 45433
[2] Senior Research Aerospace Engineer, AFRL/VACA, 2210 8th Street Bldg 146 Room 300 WPAFB, OH 45433, Associate Fellow.

I. Introduction

Small autonomous unmanned aerial vehicles (UAVs) vary in size and capability. They can be large enough to be powered by an engine or small enough to be battery-powered and may be mistaken for a small bird. These small UAVs are therefore capable of a variety of tasks ranging from reconnaissance to strategic attack. This paper applies a vehicle routing algorithm to an ISR scenario using a small team of UAVs with specific emphasis on an urban environment. Regardless of the task, it is important for a small group of aerial vehicles to become cooperative by efficiently communicating with each other and assigning each UAV to a set of targets in an efficient manner.

Before assigning targets to this team of UAVs, a hierarchical clustering method is implemented to provide additional information into the mixed integer linear program (MILP). Clustering targets in such a manner is not computationally intensive, but can in fact, reduce the computational difficulty of the MILP. In a scenario with many teams of UAVs where each UAV from each team must be assigned a target, the target clusters can be used to initially dispense each team to a group of targets. With smaller target clusters, each UAV within a team can be assigned to a target cluster. Although this assignment method is suboptimal if the operator truly desires to minimize the total engagement time, it may be adequate given the computational efficiency afforded by such a method or at least provide a good initial solution. Conversely, the operator can take advantage of this information and choose to enforce simultaneous arrival times for each target within a cluster. Clustering targets that are near enough to alert each other of an attack will allow UAVs to recognize this potential and hence surveil these targets simultaneously to avoid early detection. This technique will also prevent targets from being further camouflaged or moved once alerted to a nearby attack.

The assignment of UAVs to targets is similar to the Vehicle Routing Problem (VRP). The VRP optimizes the routes a vehicle or several vehicles should follow when delivering goods to a network of customers from a single place of origin, a depot. When assigning UAVs, the customers in this case are targets and the depot is the launch and landing site. Figure 1 depicts a network diagram of a vehicle routing problem as applied to a scenario employing two UAVs which must surveil five targets, x_1 through x_5. The distance between each node is also represented along each route.

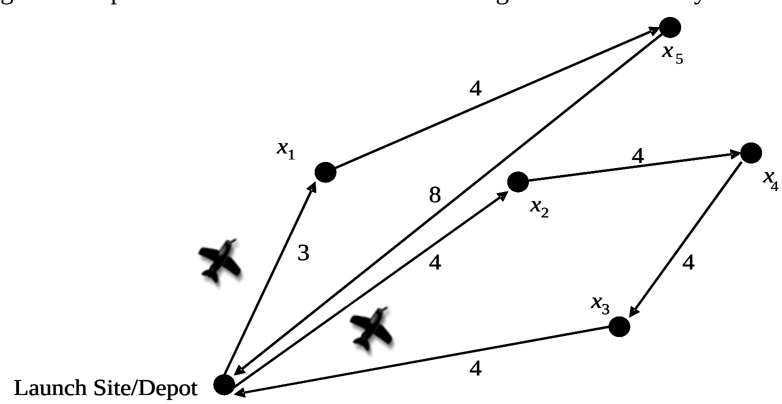

Figure 1. Example vehicle routing problem network.

Previous works have addressed various aspects of these issues. In Ref. 4 and 7, a Vehicle Routing Problem with Time Windows (VRPTW) is developed to minimize both the total distance of the routes and the number of vehicles by minimizing the summation of all chosen routes between customers in a supply delivery scenario. Whereas this method is sufficient in a pre-mission planning scenario, once the vehicles are launched the vehicle minimization objective is not appropriate.

Vehicles with the capability to perform multiple tasks are not specifically addressed in Ref. 4 and 7, but the MILP nature of the formulation enables this flexibility by adding constraints. In Ref. 4 and 7, there is only one depot since the vehicles will be routinely traveling along their assigned routes. The combat environment of the military in which teams of UAVs will be operating will not necessarily be limited to just one depot or launch site; the scenario may require multiple launch and landing sites. The time windows of the VRPTW will also not necessarily be imposed, but an operator may choose to enforce these upper and lower bounds on the target arrival times.

Various other works have discussed the optimal assignment of UAVs required to accomplish multiple tasks for each target.[3,5,6] Joint and overlapping tasks and limited time windows were imposed in Ref. 3. In Ref. 5 and 6, a MILP formulation for a wide area search munitions team of UAVs was implemented to accommodate a scenario in which targets must first be found by the UAVs before they are destroyed. Each target required three distinct tasks to be executed in a specific order. Varying cost objectives were also considered including minimizing the total flight time and minimizing the engagement time, also known as the min makespan objective.[5,6]

In Ref. 4 and 7, a VRPTW is developed which incorporates a multi-objective cost function. The cost function places equal weight on minimizing both the number of vehicles and the total distance of all chosen routes. By incorporating decision analysis techniques here such as multi-objective decision making or any type of value-focused method into the vehicle routing problem, the operator may instill a set of "values" in both the assignment process and the vehicles themselves. These "values" are integrated anytime various weighting schemes are incorporated into the objective function.

The VRP incorporated in all of these previous works is an NP-hard problem with many extensions such as the VRPTW and the Multiple Depot Vehicle Routing Problem (MDVRP). These extensions easily lend themselves to UAV task assignment problems; however, because they are NP-hard, the size of the problem and thus the computational effort increases exponentially. This issue is specifically investigated in Ref. 6. The computation times from a variety of scenarios each producing a different number of decision variables and constraints produced by the MILP are directly compared in the analysis.[6] With five UAVs, four targets, and three tasks per target, the computation was terminated because of the excessive computation time.[6] A problem formulation, therefore, must be created that is robust enough to examine a wide variety of scenarios and accommodate additional operator constraints, but must also be computationally efficient.

II. Scenario

Assume there are a finite number of targets, and the location of each target is known with certainty. These targets may be located in either an urban environment in a more rigid grid pattern or in a less constrained environment where Euclidean distances are used as opposed to rectilinear. Because the target locations are known, the operator can specify target precedence constraints based on their perceived importance. As previously discussed, we can choose to first cluster the targets to allow the targets within a cluster to be attacked or surveilled simultaneously.

In this scenario, assume each target requires exactly one task, for example, either surveillance or attack, and thus only one UAV should be assigned to each target. Suppose there are exactly N targets and a total of K aerial vehicles. Assume $K \leq N$ since the number of targets is known with certainty and each target requires only one UAV. The time to perform each task for each target or the amount of time the UAV will be at each target is called a service time. The service times may be dependent upon the UAV, the target, or both. Any length of time a UAV spends waiting for another UAV is referred to as a wait time.

Although vehicle routing problems typically require a vehicle to begin and end its route at the same depot location, the MILP created here allows vehicles to begin and end at separate launch and landing sites. This allows these small UAVs to be expendable by creating dummy nodes or dummy landing sites. Should the vehicle be detected, this flexibility becomes invaluable because an expendable UAV or any UAV traveling to a landing site that is not its launch site will not lead enemy personnel to the person who launched it or the place from which it was launched. A person may also launch the UAV from a location outside of the air base and require the UAV to land at the air base once its mission is complete. Assume there are L launch sites and a total of C landing sites.

III. Hierarchical Clustering of Targets

Cluster analysis organizes a finite set of objects or data into subsets that have meaning specific to a certain scenario. There are various methods and computing techniques that cluster data in different ways. Choosing a preferred method may depend on the scenario or it may be based solely on the computational efficiency of the algorithm. Figure 2 depicts the locations of 5 targets. Although rather simplistic, this example is sufficient to understand hierarchical clustering methods used to develop timing constraints in the MILP.

Before choosing a clustering method, a valid measurement of distance between objects must first be established. Given an urban environment, the rectilinear distance is sufficient to measure distances when the vehicle must travel along a type of grid pattern typical of streets and tall buildings. When the buildings are shorter and there is no need to follow such a rigid path, the Euclidean distance is more accurate. Although many more distance measures

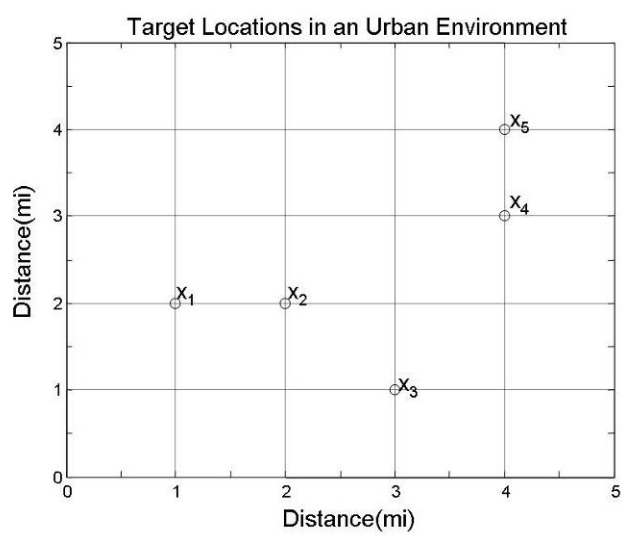

Figure 2. Grid locations of targets.

Table 1. Proximity matrix.

	x_1	x_2	x_3	x_4	x_5
x_1	0	1	3	4	5
x_2		0	2	3	4
x_3			0	3	4
x_4				0	1
x_5					0

exist, the Euclidean and rectilinear distances are the most appropriate in this scenario. Once the desired measure of distance is determined, a proximity matrix is created which stores the relationships between all objects.[2] Table 1 lists the proximity of each of the targets given in Figure 2 where the distance is measured rectilinearly. The distances as well as the time it takes to travel these distances are assumed to be the same regardless of the direction of travel. For example, the time it takes to travel from target x_4 to x_5 is equal to the time it takes to travel from x_5 to x_4.

Clustering methods are classified into different categories. Exclusive clustering methods ensure each object belongs to only one cluster whereas non-exclusive methods allow clusters to overlap and each target can belong to more than one cluster. In this method, each object is measured with a degree of belongingness. Intrinsic methods organize objects based solely on the proximity matrix. Extrinsic methods use both the proximity matrix and category labels. In this paper, exclusive, intrinsic techniques will be applied to accurately cluster the targets. Two types of exclusive, intrinsic techniques include partitional and hierarchical clustering. Partitional clustering methods partition the objects into two clusters, but hierarchical methods create a nested sequence of partitions. Since the number of UAVs may not be sufficient to cover only two clusters of targets, only hierarchical methods are utilized. Throughout the process of the hierarchical clustering method, given some set of N targets denoted $X = \{x_1,\ldots,x_N\}$ and M clusters denoted $\{C_1,\ldots,C_M\}$, $C_i \cap C_j = \Phi$ where $i,j \in [1,M]$, $i \neq j$ and $C_1 \cup C_2 \cup \ldots \cup C_M = X$.[0]

Hierarchical clustering methods include both agglomerative and divisive methods. Agglomerative methods begin the process of organizing the data by assigning each object to its own cluster and continue by merging clusters according to the proximity matrix. Divisive methods are similar to agglomerative but work in reverse, beginning by assigning all objects to the same cluster and breaking apart the clusters until each object is its own unique cluster.[2] Both methods will produce the same dendrogram, a graphic representation of the clustering method shown in Figure 3. The left hand side of the dendrogram gives the proximity at which the targets were clustered.

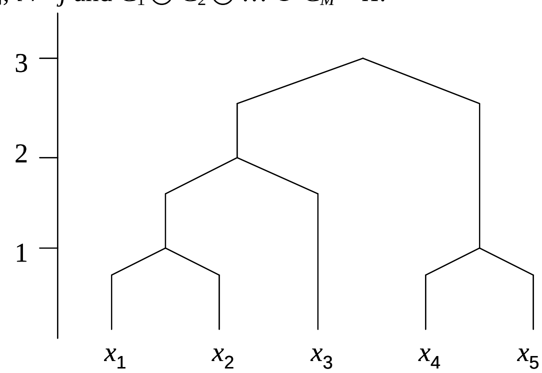

Figure 3. Dendrogram of the target clusters.

To produce this dendrogram, the proximity matrix is iteratively altered until all of the targets are combined into one cluster (using an agglomerative method). The clustering algorithm first finds the two closest targets. In this case, targets x_1 and x_2 are one mile apart as well as targets x_4 and x_5. Following the algorithm, x_1 and x_2 are first combined as it doesn't necessarily matter which two targets (x_1 and x_2 or x_4 and x_5) are clustered first in this example. There are various methods of updating the proximity matrix once a cluster is formed or altered. Among these methods are single link, completed link, group average, and minimum variance. Each method can produce a different dendrogram. Ties, as seen with the distance between x_1 and x_2 and the distance between x_4 and x_5 can significantly affect the dendrogram produced by the completed link method. The dendrogram given in Figure 3 is created using the single link method. The single link method alters the proximity matrix once a cluster is created or altered by combining the columns and rows for x_1 and x_2 into a single column and row for the newly created cluster. The distances are updated for the cluster by using the smallest distance between x_1 and x_2 to the remaining targets as shown in Figure 4. The remaining

Single Link

	x_1,x_2	x_3	x_4	x_5
x_1,x_2	0	2	3	4
x_3		0	3	4
x_4			0	1
x_5				0

	x_1,x_2	x_3	x_4,x_5
x_1,x_2	0	2	3
x_3		0	3
x_4,x_5			0

	x_1,x_2,x_3	x_4,x_5
x_1,x_2,x_3	0	3
x_4,x_5		0

Completed Link

	x_1,x_2	x_3	x_4	x_5
x_1,x_2	0	3	4	5
x_3		0	3	4
x_4			0	1
x_5				0

	x_1,x_2	x_3	x_4,x_5
x_1,x_2	0	3	5
x_3		0	4
x_4,x_5			0

	x_1,x_2,x_3	x_4,x_5
x_1,x_2,x_3	0	5
x_4,x_5		0

Group Average

	x_1,x_2	x_3	x_4	x_5
x_1,x_2	0	2.5	3.5	4.5
x_3		0	3	4
x_4			0	1
x_5				0

	x_1,x_2	x_3	x_4,x_5
x_1,x_2	0	2.5	4
x_3		0	3.5
x_4,x_5			0

	x_1,x_2,x_3	x_4,x_5
x_1,x_2,x_3	0	3.83
x_4,x_5		0

Figure 4. Updated Proximity Matrix Comparison

iterations using the single link to produce the dendrogram in Figure 3 are also shown in Figure 4. Figure 4 compares the single link, completed link, and the group average methods for updating the proximity matrix. The completed link method uses the largest distance to update the proximity matrix whereas the group average method uses the average distance of all of the targets within the cluster.[0] Although there are no notable differences in the clusters themselves, dramatic differences in the clusters are possible when using different methods of updating the proximity matrix. For our purposes as well as ease of computation, the single link method is used to create the target clusters for the MILP.

The targets that should be observed simultaneously are dependent upon the clusters created by the single link hierarchical clustering method. The number of targets which can be simultaneously surveilled is limited by the number of UAVs. Thus, the cluster size is limited to the number of UAVs available. It is also possible to specify what distance is realistic to require simultaneous arrival times within the clustering program. In this case, no targets outside of the specified proximity will be clustered and no simultaneous arrival time constraints will be enforced for these targets.

Whereas the clustering algorithm here is used to send multiple air vehicles for simultaneous surveillance, the method can also be used to group targets and assign different UAVs or groups of UAVs to each cluster

IV. The Vehicle Routing Problem with Windows

The vehicle routing problem with time widows (VRPTW) is an extension of the capacitated vehicle routing problem (CVRP). The CVRP schedules K vehicles originating at a singe depot to deliver goods to N customers. The CVRP creates a route for each vehicle where a route is a path beginning at the depot, visiting at least one customer, and returning to the same depot.[7] The capacity or supply of each vehicle is the only additional constraint that restricts the vehicles in this type of problem.[7] For our purposes, the capacity of the vehicle is unlimited and can therefore visit an unlimited number of targets, but the vehicle is constrained by its endurance or maximum flight time. It is possible to compartmentalize the vehicle allowing each vehicle to have various different capacities depending on the compartment where compartments can be viewed as different tasks. This type of strategy can be implemented to allow each vehicle to perform more than one task such as classification and reconnaissance. Although compartmentalization is not implemented, constraints can be added to the MILP to model the scenario where multiple tasks must be accomplished for each target and can be performed by the same or different UAVs. It is also possible to have vehicles with different classifications, for example, certain vehicles can perform one type of task where other vehicles can perform another. These multiple task formulations are shown in Ref. 5 and 6.

The objective function and all other constraints are typical of the VRP. Various different objectives or cost functions can be applied to all VRPs including the CVRP and VRPTW depending on the scenario. The assignment of vehicles can be optimized by minimizing the path or total distance traveled by all vehicles or by similarly minimizing the total travel time of all vehicles. The situation may demand a slightly different objective including, minimizing the total number of vehicles necessary to service all customers within a certain amount of time or to ensure all routes are as balanced as possible.[7]

The vehicle flow model defines the constraints necessary to characterize the network itself. One of these constraints guarantees that each vehicle is assigned to exactly one route. The remaining constraints describe the flow of the network ensuring each node has exactly one path entering it and one path exiting; however, the depot must have K paths exiting and entering it. Additional constraints must ensure the connectivity of the route and eliminate the possibility of any subtours.[7]

The VRPTW formulation encompasses all of these constraints as well as additional time window constraints. A typical VRPTW assigns an earliest arrival and latest arrival time for each customer. The time windows are treated as either soft time windows or hard time windows. Soft time windows are formulated such that a penalty is assigned when the customer is not serviced within its time window. Hard time windows are treated as requirements and the solution becomes infeasible if all time windows constraints are not met.[7] Although this scenario does not require pre-specified time windows, the constraints used to develop the arrival time for each target and ensure simultaneous arrival times for targets residing in the same cluster are modeled after the VRPTW time window constraints. These constraints are also necessary to develop precedence constraints for the set of customers or targets.

An operational scenario may require only one launch site which is also the landing site; however, it may also require multiple launch sites and multiple landing sites strategically located separate from the launch sites. The constraints associated with the multiple depot VRP (MDVRP) are applied to the UAV assignment MILP discussed in the next section. The MDVRP incorporates a network with multiple depots where the customers are dispersed throughout the network, i.e. they are not each clustered around a specific depot. Each route must start and end at the same depot in the MDVRP because the VRP assumes a scenario of a set of customers with recurring service needs.

In a combat scenario, the tasks associated with a set of targets are usually not recurring tasks; however, reconnaissance or surveillance needs may or may not be recurring. Therefore, the MILP does not require UAVs begin and end at the same site, but this requirement can easily be specified.

The CVRP, VRPTW, and MDVRP are just a few variations of the vehicle routing problem among many. Aspects of all of these VRP variants can be combined to create a unique VRP formulation that realistically models a given scenario.

V. MILP for UAV Task Assignment

A. Decision Variables

The decision variable $x_{ijk} = 1$ if UAV k is assigned to travel from node i to node j, and 0 otherwise; $i = 1,\ldots N+L$, $j = 1,\ldots,N,N+L+1,\ldots,N+L+C$, and $k = 1,\ldots,K$. Thus, the variable i is either a target or launch site and j is either a target or landing site. This eliminates the possibility of an air vehicle traveling directly from the launch site to landing site and ensures all UAVs will be utilized.

The arrival times at each target and each landing site by each UAV account for the remaining decision variables. The variable t_i is a continuous variable which indicates the arrival time. The variable t_{jk} is also a continuous variable, but indicates when each UAV will land at each landing site; $t_{jk} = 0$ when UAV k is not assigned to land at landing site i. Landing site arrival times are distinguished by the vehicle k whereas the target arrival times are not because more than one vehicle can land at a single landing site. This ensures that the specific landing time of each vehicle is constrained by the vehicle's endurance or the maximum flying time.

The total number of binary decision variables for the MILP is equal to $KN(L+C+N-1)$. The total number of continuous decision variables is equal to $N+KC$.

B. Cost Functions

Various cost functions can be applied to the UAV assignment problem, each with its own advantages and disadvantages. This paper examines three different cost functions.

1) Route minimization/shortest path (total cost of all routes taken).

$$\text{Min Total Distance} = \sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=1}^{K} c_{ij}\, x_{ijk} \tag{1}$$

where $x_{ijk} = \begin{cases} 1 & \text{if UAV } k \text{ strikes node } j \text{ after node } i \\ 0 & \text{otherwise} \end{cases}$

and c_{ij} = distance (or cost incurred) when traveling from node i to j

Assume all vehicles must be utilized because, for example, vehicles may be assigned to targets after they are airborne. This ensures one UAV is not assigned to all targets to limit the total distance which would greatly increase the time it takes to strike all targets.

2) Minimize the time it takes to cover all targets, also known as min makespan.

$$\text{Min} \quad \max\{ t_{jk} \} = \text{makespan} \tag{2}$$

The makespan is implemented into the MILP by adding another decision variable, makespan such that:

$$\text{makespan} \le t_{jk} \quad \forall k \in [1, K], i \in [N+1+L, N+L+C] \tag{3}$$

Targets are denoted here as the nodes numbered from 0 to N, launch sites are the nodes numbered from N+1 to N+L, and landing sites are the nodes numbered from $N+L+1$ to $N+L+C$.

3) Minimize the total time it takes for all UAVs to cover all targets.

$$\text{Min} \quad \text{Total Time} = \sum_{j=N+L+1}^{N+L+C}\sum_{k=1}^{K} t_{jk} \tag{4}$$

At first glance, it seems these equations are merely different methods of obtaining the same measurement, but they are not, in fact, equivalent. Equation (1), the shortest path type measurement, can easily be converted to time by using the speed of each air vehicle; however, Eq. (1) doesn't account for service times and thus, doesn't account for the total time and energy required. Adding service times into the cost function does not increase the computational difficulty, but it only ensures the objective function is sufficient when no wait times exist. Because wait times can only be calculated once all routes are chosen, this increases the computing times required to execute the MILP. Minimizing the routes traveled by all of the UAVs is computationally efficient, but is lacking in precision and not very robust.

By minimizing the makespan or the maximum arrival time of all of the air vehicles in Eq. (2), the total time required to strike all targets is minimized including travel times, wait times, and service times. By including the wait times for all of the UAVs, the objective functions for Eq. (2) and (4) become computationally more difficult.

C. Constraints

The constraints of the MILP define the problem itself. They ensure that each vehicle is properly tasked and they define the set of feasible solutions.

1) Each vehicle that enters a target must also exit.

$$\sum_{i=1, i\neq h}^{N} x_{ihk} - \sum_{j=1, j\neq h}^{N} x_{hjk} = 0 \quad \forall h \in [1, N], k \in [1, K] \tag{5}$$

Equation (5) ensures the same UAV will both enter and exit a target area eliminating the possibility of an air vehicle striking a target without exiting. The total number of constraints this adds to the program is *KN*.

2) Each target must be visited exactly once.

$$\sum_{k=1}^{K} \sum_{j=1, j\neq i}^{N} x_{i\,jk} = 1 \quad \forall\, i \in [1, N] \tag{6}$$

This constraint ensures that each target is covered by a UAV. It adds exactly *N* constraints to the MILP. Equations (5) and (6) are formulations described in "Heuristic Methods for Vehicle Routing Problem with Time Windows".[7]

3) Each UAV must launch from a launch site.

$$\sum_{i=N+1, i\neq j}^{N+L} \sum_{j=1}^{N} x_{ijk} = 1 \quad \forall\, k \in [1, K] \tag{7}$$

Equation (7) ensures that each UAV begins its route by exiting a launch site.[8] This constraint adds *K* constraints to the problem formulation.

4) Each UAV must land at a landing site.

$$\sum_{i=1}^{N} \sum_{\mathrm{j=N+1+L,\, i\neq j}}^{\mathrm{N}+L+\mathrm{C}} x_{ijk} = 1 \quad \forall k \in [1, K] \tag{8}$$

Equation (8) adds *K* constraints to the MILP and ensures that each UAV will complete its route by entering a landing site.[8] Both constraints 3 and 4 ensure that each vehicle is assigned to exactly one route and that each vehicle is therefore utilized. When these constraints are inequality constraints (less than or equal to one) instead of equality constraints, the MILP will allow the possibility of not using all the UAVs available. This will then minimize the number of vehicles utilized as well as the total time or total distance. This may be useful for purposes such as pre-mission planning; however, modifying these constraints also implies that the time it takes for the UAVs to cover all targets (the makespan) may increase when the number of vehicles is also minimized.

5) Each target and landing site must have an arrival time.

$$t_i + t_{ijk} + s_{ik} - M(1 - x_{ijk}) \le t_j \qquad \forall i \in [1, N+L], j \in [1, N], k \in [1, K], i \ne j \tag{9}$$

$$\text{if } i \in [N+1, N+L], \text{ then } t_i = 0$$

$$t_i + t_{ijk} + s_{ik} - M(1 - x_{ijk}) \le t_{jk} \;\; \forall i \in [1, N], j \in [N+1+L, N+L+C], k \in [1, K] \tag{10}$$

Equation (9) represents the linear time constraints for all the targets and Eq. (10) corresponds to the landing sites. These timing constraints were modified from the timing constraints used by Ombuki, Ross, and Hanshar in their paper, "Multi-objective Genetic Algorithms for Vehicle Routing Problem with Time Windows".[4] The timing constraints also guarantee that no feasible solution contains subtours. In this formulation, the variable *M* merely needs to be sufficiently large. The maximum endurance of each UAV (r_k) is guaranteed to be sufficiently large and thus, M = r_k. Equation (9) adds *NK*(*N*-1+*L*) constraints whereas Eq. (10) only adds *NCK* constraints.

6) Each UAV must arrive at its landing site within the limits of its maximum endurance.

$$t_{jk} \le r_k \qquad \forall j \in [N+1, N+L], k \in [1, K] \tag{11}$$

The max arrival time constraint adds *KL* constraints to the MILP.

7) Targets *a* and *b* residing in the same cluster or as specified by the operator should have simultaneous arrival times.

$$t_a = t_b = \mathrm{t_c} \;\; \text{such that } a, b, c \in [1, N] \tag{12}$$

The number of constraints for Eq. (12) depends on the number of targets requiring simultaneous arrival times.

8) Defines specified target precedence for arrival times.

$$t_a \le t_b \le t_c \;\; \text{such that } a, b, c \in [1, N] \tag{13}$$

$$t_a + s_{ak} \le t_b + s_{ak} \le t_c + \mathrm{s_{ak}} \;\; \text{such that } \; a, b, c \in [1, N], \forall k \in [1, K] \tag{14}$$

Similar to Eq. (12), the number of constraints specified by Eq. (13) is dependant upon the specific scenario and the precedence order set forth by the operator. Equation (14) ensures the preceding target's surveillance is complete before proceeding where Eq. (13) does not.

The total number of decision variables for the MILP is *KN*(*L*+*C*+*N*-1)+*N*+*KC*. The total number of constraints can be calculated as *NK*(*N*+*L*+*C*)+*N*+2*K*, but does not include any specified timing constraints from Eq. (12), Eq. (13), and Eq. (14).

D. Extensions

Various different situations can be represented using this formulation. Different constraints can be relaxed or added to signify different aspects of a specific mission. We have specified two different timing constraints, simultaneous and precedence, in this scenario. Various other timing constraints may be added to the formulation to model a variety of mission requirements. For example, some targets may require overlapping tasks or different types of precedence constraints. The nature of the targets may require if/then constraints. For example, if UAV *k* is assigned to target *i* then it must also be assigned to target *j*. This type of constraint is represented in Eq. (15).

$$\sum_{h=1}^{N+L} x_{hi} \le \sum_{h=1}^{N+L} x_{hjk} \tag{15}$$

Consider the situation where a solution is infeasible because the number of vehicles is limited or their endurance is not sufficient to surveil all targets. Thus, either the simultaneous timing constraints need to be relaxed or the number of targets a UAV team is assigned to needs to be decreased. The first solution involves using soft time windows or timing constraints as opposed to the hard timing constraints previously implemented. As previously discussed, soft time windows merely add a penalty for the degree to which a time window is not observed. Similarly, this type of penalty can be added to all timing constraints. This will allow the program to generate a feasible solution, but slightly penalized for violating a timing constraint that is preferred by the operator. Equation (16) shows how arrival penalties may be added to the total time cost function. The new objective incorporates the total time that each vehicle was late to a target with a set arrival time (t_{iw}).

$$\text{Total time with penalties} = \sum_{\mathrm{i=N+}L+1}^{\mathrm{N+L+C}} \sum_{k=1}^{K} t_{ik} + \sum_{\mathrm{i=1}}^{\mathrm{N}} (t_i - t_{iw}) \tag{16}$$

Equation (17) shows an example cost function with penalties incurred when simultaneous arrival times are not met.

$$\text{Total time with penalties} = \sum_{\mathrm{i=N+}L+1}^{\mathrm{N+L+C}} \sum_{k=1}^{K} t_{ik} + \sum_{i,j \text{ simultaneous}} abs\left(t_i - t_j\right) \tag{17}$$

For the case with multiple tasks, the operator can also place similar penalties or preferences on which tasks are more important or if it is most important for all tasks to be completed on one target rather than completing only one task on many targets. These types of values or weighting techniques, derived from the operator, can be implemented into the optimization.

If there are simply too many targets for the UAVs to surveil, a value function can be created to instill these objectives and constraints into the formulation. A value function is a type of weighted multi-objective cost function.

Different aspects from vehicle routing problems other than the CVRP, VRPTW, and MDVRP can be incorporated to account for uncertainty. The MILP formulation can be modified using aspects from the dynamic vehicle routing problem (DVRP) and the stochastic vehicle routing problem (SVRP). The DVRP is a method of assignment that can handle previous tasks as well as new tasks that develop after the vehicles have begun their initial routes. The SVRP handles levels of uncertainty within various aspects of the problem such as uncertainty in a target's required tasks or the location of a target.

VI. Problem Solution

Because the number of decision variables and constraints dramatically increases as the number of targets and vehicles increases, a simple example is explored to show the formulation and solution. Consider the case with two vehicles and three targets, two of which are close enough (one mile apart) to be clustered and require simultaneous arrival times (x_1 and x_2) shown below in figure 5. Assume an urban environment where the rectilinear distance is the best measurement of distance.

There are 29 total decision variables of which 24 are binary: UAV1: (x_{121}, x_{131}, x_{151}, x_{211}, x_{231}, x_{251}, x_{311}, x_{321}, x_{351}, x_{411}, x_{421}, x_{431}); UAV2: (x_{122}, x_{132}, x_{152}, x_{212}, x_{232}, x_{252}, x_{312}, x_{322}, x_{352}, x_{412}, x_{422}, x_{432})

There are 5 continuous variables, target x_1 arrival time (t_1), target x_2 arrival time (t_2), target x_3 arrival time (t_3), landing site x_5 arrival time by UAV 1 (t_{51}), and landing site x_5 arrival time by UAV 2 (t_{52}). The 40 constraints are broken out by the constraint numbers listed above.

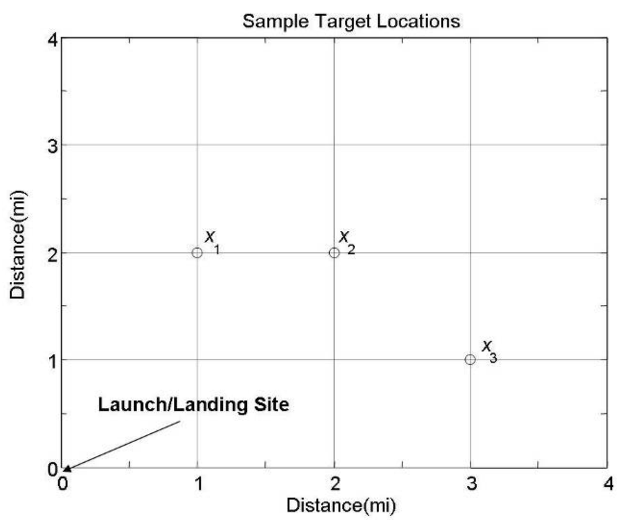

Figure 5. Sample target set.

Constraint 1:
$(x_{211} + x_{311}) - (x_{121} + x_{131}) = 0$
$(x_{121} + x_{321}) - (x_{211} + x_{231}) = 0$
$(x_{131} + x_{231}) - (x_{311} + x_{321}) = 0$
$(x_{212} + x_{312}) - (x_{122} + x_{132}) = 0$
$(x_{122} + x_{322}) - (x_{212} + x_{232}) = 0$
$(x_{132} + x_{232}) - (x_{312} + x_{322}) = 0$

Constraint 2:
$x_{121} + x_{131} + x_{122} + x_{132} = 1$
$x_{211} + x_{231} + x_{212} + x_{232} = 1$

$x_{311} + x_{321} + x_{312} + x_{322} = 1$

Constraint 3:
$x_{411} + x_{421} + x_{431} = 1$
$x_{412} + x_{422} + x_{432} = 1$

Constraint 4:
$x_{151} + x_{251} + x_{351} = 1$
$x_{152} + x_{252} + x_{352} = 1$

Constraint 5:
UAV 1:
$t_1 + t_{121} - r_1(1\text{-}x_{121}) \le t_2$
$t_1 + t_{131} - r_1(1\text{-}x_{131}) \le t_3$
$t_2 + t_{211} - r_1(1\text{-}x_{211}) \le t_1$
$t_2 + t_{231} - r_1(1\text{-}x_{231}) \le t_3$
$t_3 + t_{311} - r_1(1\text{-}x_{311}) \le t_1$
$t_3 + t_{321} - r_1(1\text{-}x_{321}) \le t_2$
$t_4 + t_{411} - r_1(1\text{-}x_{411}) \le t_1$
$t_4 + t_{421} - r_1(1\text{-}x_{421}) \le t_2$
$t_4 + t_{431} - r_1(1\text{-}x_{431}) \le t_3$

UAV 2:
$t_1 + t_{122} - r_2(1\text{-}x_{122}) \le t_2$
$t_1 + t_{132} - r_2(1\text{-}x_{132}) \le t_3$
$t_2 + t_{212} - r_2(1\text{-}x_{212}) \le t_1$
$t_2 + t_{232} - r_2(1\text{-}x_{232}) \le t_3$
$t_3 + t_{312} - r_1(1\text{-}x_{312}) \le t_1$
$t_3 + t_{322} - r_1(1\text{-}x_{322}) \le t_2$
$t_4 + t_{412} - r_2(1\text{-}x_{412}) \le t_1$
$t_4 + t_{422} - r_2(1\text{-}x_{422}) \le t_2$
$t_4 + t_{432} - r_1(1\text{-}x_{432}) \le t_3$

Constraint 5:
UAV 1:
$t_1 + t_{151} - r_1(1\text{-}x_{151}) \le t_{11}$
$t_2 + t_{251} - r_1(1\text{-}x_{251}) \le t_{21}$
$t_3 + t_{351} - r_1(1\text{-}x_{351}) \le t_{31}$

UAV 2:
$t_1 + t_{152} - r_2(1\text{-}x_{152}) \le t_{12}$
$t_2 + t_{252} - r_2(1\text{-}x_{252}) \le t_{22}$
$t_3 + t_{352} - r_2(1\text{-}x_{352}) \le t_{32}$

Constraint 6:
$t_{51} \le r_1$
$t_{52} \le r_2$

Constraint 7:
$t_1 = t_2$

There are three different approaches to this optimization in terms of the cost function. Using the shortest path minimization, we have the total route distance = $x_{121} + 3x_{131} + 3x_{151} + x_{211} + 2x_{231} + 4x_{251} + 3x_{311} + 2x_{321} + 4x_{351} + 3x_{411} + 4x_{421} + 4x_{431} + x_{122} + 3x_{132} + 3x_{152} + x_{212} + 2x_{232} + 4x_{252} + 3x_{312} + 2x_{322} + 4x_{352} + 3x_{412} + 4x_{422} + 4x_{432}.$

According to the min makespan objective, we wish to minimize the maximum value of the set, $\{t_{51}, t_{52}\}$. The total time minimization has the following objective function, $t_{51} + t_{52}$.

Assume the endurance of each vehicle is 1.5 hours and they both travel with a speed of 25 mi/hr. The service time for each target is 0.25 hours irrespective of the UAV. We also assume that any targets within one mile of each other must have simultaneous arrival times. We compute the solution by implementing the MILP we have created into the GNU Linear Programming Kit for the GLPK solver.[1]

The simplicity of the example results in no difference between the solutions of the three methods. In both cases, the total distance traveled between targets and the launch and landing site is 16 miles and the latest arrival time is .9 hours after launch. The optimization assigned one UAV to arrive at target x_1 in 0.16 hours and then complete its path by returning to the landing site within 0.53 hours of its launch. The second UAV is assigned to arrive at target x_2 in 0.16 hours which is the same arrival time of target x_1 as desired. The second UAV continues its path by arriving at target x_3 at 0.49 hours and landing at 0.9 hours from launch. Because targets x_1 and x_2 were one mile apart, their requirement for simultaneous arrival times was met. The difference in computation time for the three methods is also negligible with such small problems. The operator can easily apply additional timing constraints by specifying a precedence ordering. For example, when the operator specifies that surveillance on x_3 must be complete before surveillance begins on x_1, then the assignment of UAVs changes to accommodate these new demands. With these additional constraints, the first UAV is assigned to arrive at x_3 at 0.16 and x_2 at 0.49 then landing at 0.9. The second UAV is assigned to arrive at x_1 at 0.49 and land at 0.86.

As mentioned earlier, vehicle routing problems can become increasingly complex and require substantially more computing time as the complexity increases. Table 2 displays computing times given the number of targets, vehicles, launch and landing sites as well as additional timing constraints. The computation times are also separated by the three different cost functions, minimizing the total distance of all routes, minimizing the makespan, and the minimization of the total time.

Table 2. Computing time of various examples.

Targets N	Air Vehicles K	Launch Land	Launch	Land	Decision Variables	Constraints	Computation Time(s) Distance	Min	Max	Computation Time(s) Makespan	Min	Max	Computation Time(s) Total Time	Min	Max
3	2	1	0	0	29	37	0.034	0.028	0.133	0.036	0.033	0.115	0.036	0.022	0.143
4	2	1	2	2	82	88	0.044	0.036	0.161	0.523	0.226	0.888	0.824	0.359	1.295
4	3	0	1	1	67	82	0.051	0.043	0.172	0.093	0.074	0.178	0.093	0.088	0.203
5	2	0	2	1	77	89	0.057	0.036	0.151	0.510	0.339	0.695	0.650	0.419	0.964
5	3	1	1	2	149	161	0.132	0.054	0.519	13.859	3.886	30.898	46.323	18.756	87.424
5	4	0	2	2	173	193	0.200	0.078	1.049	13.692	3.282	34.300	36.140	14.483	75.876
6	3	0	1	2	156	174	0.277	0.060	1.321	16.603	8.037	30.961	51.411	19.019	79.705
6	4	1	2	1	254	278	1.289	0.150	13.919	466.14	101.00	1,397	7,401	3,409	17,506
7	4	1	0	1	267	295	9.499	0.180	111.334						
8	4	1	1	1	368	400	41.099	0.792	523.673						

In all cases except one shown in table 2, the shortest path type method is computationally faster than both the min makespan and total time methods; however these methods do not always produce the same solution. Using a seven target, three vehicle example, we can compare the different solutions each method produces. There is only one launch site which is the same as the landing site for this example. Figure 6 depicts the target locations and launch/landing site.

Additional simultaneous timing constraints were added to this routing problem. It is assumed that the operator desired only vehicles that were less than or equal to one mile apart to have simultaneous arrival times. Thus, the clustering algorithm assigns targets, x_4 and x_6 to the same cluster. The clustering also combines targets x_1 and x_2 to the same cluster. Tables 3 through 5 show the comparison of the solutions generated by each method.

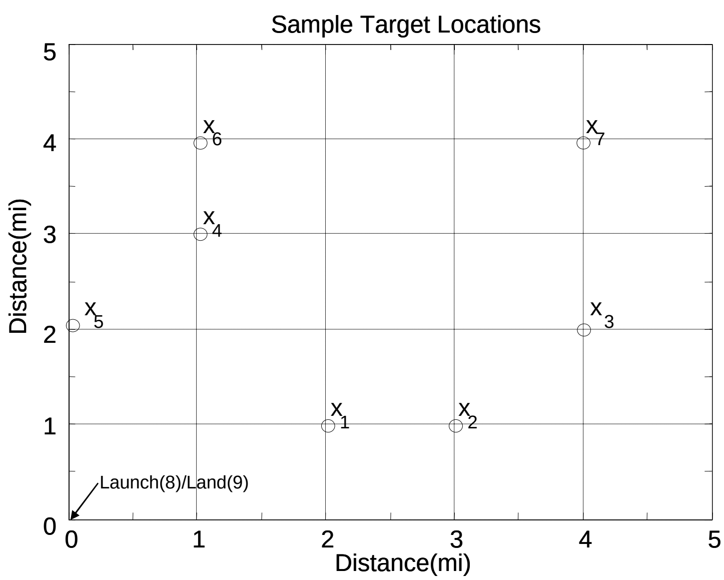

Figure 6. Target locations for method comparison.

The total distance of the routes traveled by the air vehicles is 30 miles for the shortest path method, but it is 34 for the min total time method and 38 miles for the min makespan method. Even though the route is longer for the min makespan, the latest arrival time at the landing site is less than that of the solution for the shortest path method. One reason this occurs is because of the service times required for each target. In this example, the service time for each target, regardless of the UAV, is 0.2 hours. Each air vehicle is assumed to travel at a speed of 25 mi/hr with a maximum endurance of 1.5 hours.

Similar to the total distance comparison, the min makespan method certainly produces a solution with the shortest makespan at 1.08 hours after launch. The last vehicle to arrive when employing the total distance method arrives at 1.5 hours after launch, whereas the last UAV to arrive for the total time method arrives at 1.24 hours. Because the total distance method does not account for service times or wait times, this method can get careless with its arrival times. For example, the true makespan of the total distance method is 1.44 not 1.5 as stated in table 3. Since the method does not minimize time, it only ensures the final arrival times are less than or equal to 1.5 and inadvertently adds unnecessary wait times as it did in this example.

Table 3. Solution for min total distance.

UAV 1			
i	j	ti/tik	Path Distance
8	5	0.08	8
5	4	1.1	
4	9	1.46	
UAV 2			
i	j	ti/tik	Path Distance
8	1	0.16	6
1	9	0.48	
UAV 3			
i	j	ti/tik	Path Distance
8	2	0.16	16
2	3	0.44	
3	7	0.78	
7	6	1.1	
6	9	1.5	
Total Time (hrs)	**Makespan (hrs)**	**Total Distance**	**Computation Time (s)**
3.44	1.5	30	0.475

Table 4. Solution for min makespan.

UAV 1			
i	j	ti/tik	Path Distance
8	6	0.2	16
6	7	0.56	
7	9	1.08	
UAV 2			
i	j	ti/tik	Path Distance
8	4	0.2	10
4	1	0.72	
1	9	1.04	
UAV 3			
i	j	ti/tik	Path Distance
8	5	0.08	12
5	3	0.44	
3	2	0.72	
2	9	1.08	
Total Time (hrs)	**Makespan (hrs)**	**Total Distance**	**Computation Time (s)**
3.2	1.08	38	57.336

Table 5. Solution for min total time.

UAV 1			
i	j	ti/tik	Path Distance
8	5	0.08	4
5	9	0.36	
UAV 2			
i	j	ti/tik	Path Distance
8	4	0.2	14
4	3	0.56	
3	1	0.88	
1	9	1.2	
UAV 3			
i	j	ti/tik	Path Distance
8	6	0.2	16
6	7	0.52	
7	2	0.88	
2	9	1.24	
Total Time (hrs)	**Makespan (hrs)**	**Total Distance**	**Computation Time (s)**
2.8	1.24	34	119.769

According to the MILP formulation constraints above, each vehicle must be utilized. The min makespan method will inherently use as many UAVs as possible; however, this is not true for the total distance method or the total time method as the total time and total distance will generally increase as the number of air vehicles increase. By modifying constraints 3 and 4 as previously discussed, we can also minimize the number of vehicles employed. Table 6 shows the new solution when these utilization constraints are eliminated.

Generally, the total distance method will attempt to assign all targets to one vehicle especially when there is only one launch and landing site. The number of UAVs required is dependent upon the size of the clusters and the endurance of the vehicles. The utilization constraint can, in most cases, limit the makespan without becoming computationally more difficult. Table 7 shows the new solution when the utilization constraints are eliminated and the total time objective is employed.

Table 6. Unconstrained total distance solution.

UAV 1			
i	j	ti/tik	Path Distance
8	1	0.16	12
1	5	0.48	
5	4	1.1	
4	9	1.5	
UAV 2			
i	j	ti/tik	Path Distance
8	2	0.16	16
2	3	0.44	
3	7	0.78	
7	6	1.1	
6	9	1.5	
Total Time (hrs)	**Makespan (hrs)**	**Total Distance**	**Computation Time (s)**
3	1.5	28	7.612

Table 7. Unconstrained total time solution.

UAV 1			
i	j	ti/tik	Path Distance
8	2	0.16	14
2	7	0.48	
7	6	0.84	
6	9	1.24	
UAV 3			
i	j	ti/tik	Path Distance
8	1	0.16	16
1	3	0.48	
3	4	0.84	
4	5	1.12	
5	9	1.4	
Total Time (hrs)	**Makespan (hrs)**	**Total Distance**	**Computation Time (s)**
2.64	1.24	30	182.310

Both tables 6 and 7 show the tendency to assign all targets to one vehicle which is limited only by the simultaneous constraints placed on the targets and endurance. The makespan for the total time method significantly increases when fewer vehicles are utilized.

Interestingly, even though the constraints are relaxed and at least one constraint eliminated, the computation time increases, due to the increase in feasible alternatives. This example shows the importance lower and upper bounds have on the computation time. Better bounds imply a smaller feasible space and less computation time. This example also implies that other factors than the number of constraints and decision variables affect the computational efficiency of the MILP.

VII. Conclusions

The method of assignment employed by a team of UAVs may be mission specific. Thus, certain methods may be preferred based on a given scenario. The MILP formulations presented here show how various constraints can easily be added or removed to represent a variety of scenarios especially related to the precise engagement of UAV performed ISR.

The min makespan method of minimizing the total engagement time is generally the most precise option when encountering a scenario as previously described. It balances all routes to minimize the makespan or the latest arrival time of all vehicles to a landing site. Although the min makespan optimization of the UAV assignment problem is more precise, its additional computation time may cause it to become impractical in many instances. The cost function which minimizes the total time is the most computationally difficult of all three methods. Without the utilization constraints, it would also tend to assign all targets to as few UAVs as is allowable by the endurance constraints. The minimization of the routes all UAVs travel is a practical alternative to obtain an optimal solution with less computational difficulty. When the utilization constraints are enforced, the total distance method becomes more precise in terms of ensuring all targets are covered in a minimal amount of time.

References

[1]GLPK (GNU Linear Programming Kit), Software Package, Version 3.2.4. Department for Applied Informatics, Moscow Aviation Institute, Moscow, Russia, 2003.

[2]Jain, Anil K., Dubes, Richard C. Algorithms for Clustering Data. New Jersey: Prentice-Hall Inc., 1988. Pp 55-90

[3]Kingston, D.B. and Schumacher, C. "Time-Dependent Cooperative Assignment", American Control Conference, 2005, pp. 4084-4089.

[4]Ombuki, B., Ross, B.J., Hanshar, F. "Muti-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows". Applied Intelligence, 2006. Volume 24, 17-30. http://www.springerlink.com/content/y417535551321426/fulltext.pdf.

[5]Schumacher, C., Chandler, P.R., Pachter, M., and Lior, S. "Optimization of Air Vehicles Operations Using Mixed-Integer Linear Programming", *to appear* in the Journal of the Operations Research Society.

[6]Schumacher, C., Chandler, P.R., Pachter, M., and Pachter, L. "UAV Task Assignment with Timing Constraints via Mixed-Integer Linear Programming", AIAA Unmanned Unlimited Conference, Chicago, IL, September, 2004. AIAA-2004-6410.

[7]Tan, K.C., Lee, L.H., Zhu, K.Q. "Heuristic Methods for Vehicle Routing Problem with Time Windows". Artificial Intelligent in Engineering, Elsevier (2001) 281-295 http://guava.comp.nus.edu.sg/~kzhu/docs/ai-eng2001.pdf.

[8]Toth, Paolo and Daniele Vigo. The Vehicle Routing Problem. Philadelphia: Society for Industrial and Applied Mathematics, 2002.