2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Towards Real Time Scheduling for Persistent UAV Service: A Rolling Horizon MILP Approach, RHTA and the STAH Heuristic

Byung Duk Song, Jonghoe Kim, and James R. Morrison*, *Member, IEEE*

Abstract— The automation of logistics tasks for fleets of UAVs is a key element of persistent operation. Such automation includes the provision of robotic service stations to replace consumables and orchestration algorithms enabling the UAVs to simultaneously pursue their objectives and manage the logistics process. Here we consider a system of UAVs and service stations distributed across a field of operations whose purpose is to provide continuous escort/surveillance to customers traversing known time-space trajectories. Our goal is to develop centralized real-time large scale-system orchestration methods for such a service. This goal is pursued in three directions.

- **We extend an existing mixed integer linear program (MILP) formulation to allow for arbitrary UAV initial locations and fuel levels. The MILP uses a more general service station recharge model. The new MILP is incorporated into a rolling horizon optimization for real time use.**
- **We extend an RHTA heuristic to allow for arbitrary fuel levels and UAV locations.**
- **Based on insight from the problem formulation, the STAH heuristic is developed.**

Numerical studies assess the effectiveness and numerical character of the proposed approaches. STAH was at least 30 times faster than RHTA with similar values. Both are much faster than the MILP solved via CPLEX. A real time scheduling example is considered.

I. INTRODUCTION

Persistent operation of systems of UAVs is enabled by the inclusion of automated service stations and logistics orchestration algorithms. The service stations provide replenishment services to UAVs in need of consumables such as fuel or batteries. The orchestration algorithms ensure that the logistics tasks are accomplished before the UAV consumables are next required and in such a fashion as to minimally effect the UAV system mission objectives. Such persistent systems of UAVs may find application roles in surveillance, border patrol, tracking, aerial bombardment, crop dusting and aerial photography. Here we develop methods for real time persistent operation of a system of UAVs looking toward large scale implementation.

A. Relevant Literature

There have been efforts to determine UAV system activities that incorporate an awareness of the UAV limitations, such as fuel/battery life; c.f., [1-6]. Our decision variables are inspired by classical MILP formulation for the VRP as used in [5]. Such work has not allowed UAVs to return to the field after a single trip. Research incorporating logistics tasks for persistent operations has been pursued since about 2007.

Centralized real time algorithms to enable persistent operation, directing UAVs to conduct multiple flights in the planning horizon and including visits to service stations, were conducted in [7-12]. Uncertainties such as UAV health and fuel levels can be incorporated. They used methods including Markov decision processes, approximate dynamic programming, receding horizon task assignment (RHTA), reinforcement learning and mixed integer linear programming (MILP). Indoor demonstration of persistent operations was discussed with service stations positioned at a single location.

Service stations for UAVs have been studied, developed or used for demonstrations in [7-9,12-17]. In [16], the location of stations on an n by n grid is determined by solving the p-median problem.

Scheduling methods for persistent operations for a system of UAVs with service stations distributed across a field of operations were conducted in [18-20]. There, the task was to provide uninterrupted target following (e.g., security escort) service to customers over a finite horizon. The customer paths were assumed deterministic and there were no disturbances. Their problem was thus a finite horizon deterministic optimization. MILP models were used and a genetic algorithm (GA) proposed for tractability. A small scale indoor demonstration was discussed in [19]. The UAVs used vision feedback for target tracking and landing ([21]).

The authors in [20] pursued a similar problem to that of [18-19]. However, in addition to determining which UAV should accomplish which task, they also conducted system design. That is, their MILP determines the type of UAVs, number of UAV, locations of station and number of stations. An RHTA and branch and bound algorithm (B&B) are developed and studied.

*The work reported here has been supported by National Research Foundation of Korea (NRF) N01130498.

Byung Duk Song, Jonghoe Kim and James R. Morrison are with the Department of Industrial and Systems Engineering, KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea.

*Corresponding author (e-mail: james.morrison@kaist.edu; homepage: http://xS3D.kaist.edu; phone: +82 – 42 – 350- 3127; fax: +82 – 42 – 350 – 3110).

B. Contribution

Here we focus on a system of UAVs and service stations geographically distributed across a field of operations. The UAVs may have different maximum travel speeds and fuel capacity. Several tasks may be conducted between visits to any desired service station. UAVs may return to service after replenishment at a station. The goal is to provide uninterrupted service to all customers. Multiple UAVs may be used to ensure uninterrupted service. Customers provide their planned time-space trajectory; it may change if they desire, but they will inform the system of their new plan. UAV locations and fuel levels may deviate from the plan.

The efforts of [7-12] focused on real time control. In their demonstrations, their service stations are collocated. As in their work, we develop a centralized control approach to the persistent UAV problem. However, our focus is on *large scale persistent operations* (distributed across a field). In this context the inherent computational complexity of prior formulations (when using centralized controllers) may limit their applicability.

Unlike prior real time efforts, our approach is based in a MILP formulation. The exact solutions to this formulation provide insight into the solution structure that we exploit to develop a heuristic that is about 1,000 times faster than solving the MILP via CPLEX.

We draw inspiration for our efforts from [18-20]. While the MILPs and heuristics in [18-20] allowed for geographically distributed service stations, their formulations cannot be used in a real time context. The MILP, GA, RHTA and B&B methods of [18-20] require all UAVs to be initially located at the service stations with a particular fuel/battery level. As such they cannot be used for a rolling horizon real time approach; the formulation does not allow for UAVs to be arbitrarily located and fueled initially.

Toward real time task assignment, the contributions of this paper are as follows. We

- Develop a MILP model allowing for arbitrary UAV initial location and fuel level over a finite horizon. An affine model for UAV replenishment time as a function of fuel consumed is used. (Section II)
- Develop a heuristic (STAH) inspired by the solutions obtain from solving the MILP. STAH helps to address the computational intractability of the MILP. (Section III.B)
- Extend the RHTA from [20] to allow for arbitrary initial conditions. (Section III.C)
- Insert the MILP, STAH or RHTA into a rolling horizon task assignment approach to provide the real time capability. (Section IV)
- Conduct numerical experiments to assess the proposed methods. RHTA and STAH are about 25 and 1,000 times faster than the MILP solved via CPLEX. They provide solutions within about 1-10% optimal. The rolling horizon approach is demonstrated via example. (Section V)

Concluding remarks are presented in Section VI.

II. Mathematical Formulation

A MILP model allowing arbitrary initial location and battery or fuel level is first developed. It extends the model of [18-20]. These arbitrary initial conditions will allow it to be used together with a rolling horizon approach for real time.

The time-space customer trajectories are divided into segments called split jobs that may each be served by a different UAV. Each split job must be served by one UAV. Each split job i is defined by a start location (x_{is},y_{is}), start time Ei, finish location (x_{ie},y_{ie}) and end time Li. (We assume the speed during that split job is constant.) Figure 1 shows a time-space trajectory divided into a set of split jobs. The travel distance D_{ij} from split job i's finish point or station i to split job j's start point or station j is calculated using the Euclidean distance. Note that D_{ij} is not equal to D_{ji} (the start and end points change). They are not decision variables and must be determined in advance. This approach serves to discretize each mission. Using more split jobs for a given customer trajectory may give a lower objection function value, but will increase computational complexity.

The split job data, station locations, current location of each UAV and current battery/fuel level of each UAV is assumed given and constant. We next develop a MILP model. Note that each station has two indices; they distinguish when a UAV starts a trip (flight) from that station and ends a trip (flight) at that station.

A. Notation

i, j	:	Indices for jobs
S	:	Index for stations
K	:	Index for UAVs
R	:	Index of a UAV's r^{th} flight
N_J	:	Number of split jobs
N_{UAV}	:	Number of UAVs in the system
N_{STA}	:	Number of recharge stations
N_R	:	Maximum number of flights per UAV during the time horizon
M	:	Large positive number
D_{ij}	:	Distance from the finish point of split job i to the start point of split job j

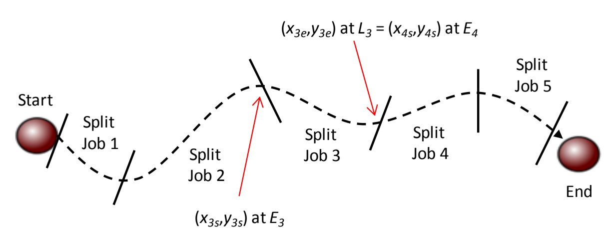

Figure 1. A time-space trajectory is divided into split jobs.

Ei	:	Start time of split job i
Li	:	End time of split job i
Pi	:	Processing time of split job i ($Li - Ei$)
H	:	Required time for fully recharge (refuel) the empty fuel tank (battery).
U	:	Setup time for recharge/refuel process
q_k	:	Maximum traveling time of UAV k
$q_{k,ini}$	:	Initial level of battery(fuel) of UAV k
S_{ok}	:	Initial location of UAV k
TS_k	:	Travel speed of UAV k
Ω_J	:	$= \{1, \ldots, N_J\}$, Set of split jobs
Ω_{SS}	:	$= \{N_J+1, N_J+3, \ldots, N_J+2\cdot N_{STA}-1\}$, set of UAV flight start stations
Ω_{SE}	:	$= \{N_J+2, N_J+4, \ldots, N_J+2\cdot N_{STA}\}$, set of UAV flight end stations
Ω_A	:	$= (\Omega_J \; U \; \Omega_{SS} \; U \; \Omega_{SE}) = \{1, \ldots, N_J+2\cdot N_{STA}\}$, set of all jobs and recharge stations
Ω_{INI}	:	$=\{1_{INI}, \ldots, K_{INI}\}$, set of initial UAV location
X_{ijkr}	:	Binary decision variable, 1 if UAV k processes split job j or recharges at station j after processing split job i or recharging at station i during the r^{th} flight; 0, otherwise.
C_{ikr}	:	Binary decision variable, split job i's start time by UAV k during its r^{th} flight or UAV k's recharge start time at station i; otherwise its value is 0.
q_{kr}	:	Real number decision variable, total battery (fuel) consumption for UAV k during its r^{th} flight

B. Recharge/refuel time function

We assume that UAVs fully replenish their energy when they visit a service station. The duration of time is affine in the remaining fuel level:

$$RT_f = \left(\frac{H}{q_k}\right)\cdot\left(q_{kr-1} + q_k - q_{k,ini}\right) + U \tag{1}$$

$$RT_r = \left(\frac{H}{q_k}\right)\cdot q_{kr-1} + U \tag{2}$$

where H and U are constants. Equation (1) is used for the first station visit for each UAV. After the first visit, equation (2) is applied (each UAV was fully charged in a previous station visit).

C. Mixed integer linear program

$$Minimize \sum_{i\in\Omega_A}\sum_{j\in\Omega_A}\sum_{k\in K}\sum_{r\in R} D_{ij}X_{ijkr} \tag{3}$$

Subject to

$$\sum_{j\in\Omega_J\cup\Omega_{SE}} X_{S_{ok},jk1} = 1 \qquad (k \in K) \tag{4}$$

$$\sum_{s\in\Omega_J\cup\Omega_{INI}}\sum_{j\in\Omega_J\cup\Omega_{SE}} X_{sjkr} = 1 \qquad (k \in K, r \in R) \tag{5}$$

$$\sum_{s\in\Omega_{SE}}\sum_{i\in\Omega_J\cup\Omega_{SS}} X_{iskr} = 1 \qquad (k \in K, r \in R) \tag{6}$$

$$\sum_{i\in\Omega_J\cup\Omega_{SS}} X_{iskr} = \sum_{i\in\Omega_J\cup\Omega_{SE}} X_{s-1,ikr+1} \quad (k \in K, r = 1, \ldots, N_R - 1, s \in \Omega_{SE}) \tag{7}$$

$$C_{skr} = C_{s-1,kr+1} \quad (k \in K, r = 1, \ldots, N_R - 1, s \in \Omega_{SE}) \tag{8}$$

$$\sum_{k\in K}\sum_{r\in R}\sum_{i\in\Omega_A} X_{ijkr} = 1 \qquad (j \in \Omega_J) \tag{9}$$

$$\sum_{j\in\Omega_A} X_{ijkr} - \sum_{j\in\Omega_A} X_{jikr} = 0 \quad (i \in \Omega_J, k \in K, r \in R) \tag{10}$$

$$\sum_{i\in\Omega_J\cup\Omega_{SS}} X_{iskr} = 0 \quad (k \in K, r \in R, s \in \Omega_{SS} \cup \Omega_{INI}) \tag{11}$$

$$C_{ikr} + P_i + D_{ij}/TS_k - C_{jkr} \le M\left(1 - X_{ijkr}\right) \quad (i \in \Omega_J \cup \Omega_{SS}, j \in \Omega_J \cup \Omega_{SE}, k \in K, r \in R) \tag{12}$$

$$M \cdot \sum_{j\in\Omega_J\cup\Omega_{SE}} X_{ijkr} \ge C_{ikr} \quad (i \in \Omega_J \cup \Omega_{SS}, k \in K, r \in R) \tag{13}$$

$$C_{ikr} + RT_f + D_{ij}/TS_k - C_{jkr} \le M\left(1 - X_{ijkr}\right) \quad (i \in \Omega_{SS}, j \in \Omega_J \cup \Omega_{SE}, k \in K, r = 2) \tag{14}$$

$$C_{ikr} + RT_r + D_{ij}/TS_k - C_{jkr} \le M\left(1 - X_{ijkr}\right) \quad (i \in \Omega_{SS}, j \in \Omega_J \cup \Omega_{SE}, k \in K, r > 2) \tag{15}$$

$$\sum_{k\in K}\sum_{r\in R} C_{ikr} = E_i \quad (i \in \Omega_J) \tag{16}$$

$$\sum_{i\in\Omega_A}\sum_{j\in\Omega_A}\frac{D_{ij}}{TS_k}\cdot X_{ijkr} + \sum_{i\in\Omega_J}\sum_{j\in\Omega_A} P_i \cdot X_{ijkr} \le q_{kr} \quad (k \in K, r \in R) \tag{17}$$

$$q_{kr} \le q_{k,ini} \qquad (k \in K, r = 1) \tag{18}$$

$$q_{kr} \le q_k \qquad (k \in K, r \ne 1) \tag{19}$$

$$C_{ikr} \ge 0 \qquad (k \in K, r \in R, i \in \Omega_A) \tag{20}$$

$$q_{kr} \ge 0 \qquad (k \in K, r \in R) \tag{21}$$

$$X_{ijkr} \in \{0,1\} \quad (i \in \Omega_A, j \in \Omega_A, k \in K, r \in R) \tag{22}$$

The objective function (3) minimizes the total travel distance. Constraint (4) ensures that every UAV start its first flight from its initial location. Constraint (5) guarantees that UAV k starts its flight from the initial location or split job and proceeds to a split job or ending service station. Constraint (6) indicates that UAV k must finish each flight at an ending service station. Constraint (7) requires that if UAV k finishes its r^{th} flight at service station s, then its r+1th flight starts at service station s. Constraint (8) states that the finish time of a UAV's r^{th} flight is equal to the start time of the UAV's r+1th flight. Constraint (9) forces every split job in Ω_J to be processed by a UAV. Constraint (10) is a flow balance constraint that ensures a UAV does not finish its flight at a job. Constraint (11) ensures that a UAV cannot finish its flight at the starting station (this is due to our assignment of two indices to each station).

Constraints (12) and (13) are used to link the C_{ikr} and X_{ijkr}

decision variables. The start time constraint (12) gives the relationship between the split job start time or service station visit start time (which is the time that the fuel replacement process begins) and that of its successor for the same UAV during its r^{th} flight. Constraint (13) implies that the value of C_{ikr} is set to zero if split job i is not assigned to UAV k's r^{th} flight.

Constraints (14) and (15) were applied to determine recharge (refuel) time of each UAV and job start time after replenishment. If $r = 2$, constraint (14) is applied with equation (1) to reflect the initial level of battery (fuel) of UAV k ($q_{k,ini}$) on the recharge (refuel) time. If $r > 2$, constraint (15) together with equation (2) is used instead of constraint (14).

Constraint (16) ensures that every split job in Ω_J starts its process at its pre-determined start time. The fuel constraint (17) states that the total flight time, including travel time between split jobs and process time for split jobs, cannot exceed UAV k's fuel capacity. Constraints (18) and (19) limit the range of decision variables q_{kr}. Finally, constraints (20), (21) and (22) describe decision variable of proposed mathematical formulation.

Note that proposed mathematical formulation requires $N_R > 1$ due to constraints (7) and (8). N_R is just the maximum number of flights allowed for each UAV; this is not in any way a restriction on the modeling capability.

III. Solution Approach

MILP formulations are computationally complex. We discuss how we solve our MILPs as well as develop a heuristic and extend an RHTA approach to significantly improve computation.

A. *CPLEX*

CPLEX is a commercial solver designed to solve large scale MILPs. We employ CPLEX 12.4 to obtain an optimal solution to our MILP when CPLEX can solve the problem.

B. *Sequential Task Assignment Heuristic*

We develop the Sequential Task Assignment Heuristic (STAH) to address the computational intractability of the MILP formulation. Refer to Appendix I for the pseudo code of STAH. We describe some of the key points next.

Customers are ordered by the start time of their service from earliest to latest. Let $P = \{1,2,\ldots,p\}$ be the set of customers and P_t be the set of split jobs for customer t. Elements of P_t are arranged in non-decreasing order of split job start time. We assume that the customer split jobs form a continuous path. If not, consider them as two customers. We also assume that every UAV has sufficient speed to serve the split jobs (that is, no customer moves faster than the slowest UAV and we are free to assign any UAV to their split jobs).

Starting with $t = 1$, all split jobs for a customer are assigned before proceeding to the next customer. For a given customer, its split jobs are assigned in chronological order, starting from the first. To assign a UAV to a particular split job l, two values are calculated for all UAVs.

- The first value $V_D(k)$ corresponds to the UAV k directly proceeding to split job l from the end of its most recent assigned task and sequentially serving as many of customer k's split jobs for which it has sufficient fuel (and can then make it to a service station). This is a *direct flight*.
- The second value $V_I(k)$ corresponds to the UAV k proceeding to the station nearest the end of its most recent assigned task, prior to proceeding to split job l and sequentially serving as many of customer k's split jobs for which it has fuel (and can then make it to a service station). We refer to this as an *indirect flight*.

Throughout, for simplicity of notation, we suppress the dependence of these values on anything other than the UAV index k. The other variables will be immediately obvious as a function of where in the pseudo code the calculation is located. The UAV achieving the maximum value is assigned to that split job. If two UAVs achieve the maximum, one is selected arbitrarily. In the case of ties, a UAV and direct/indirect flight are selected randomly. We require some notation. Let $Cl(k)$ and $Cq(k)$ be the location and battery/fuel level of UAV k , respectively(after completing its last scheduled task). Let $A(k)$ be the time at which UAV k completes its last scheduled task and is available to serve. Recall that l is our split job index. Use the notation $(.)'$ and $(.)''$ to denote the start and end locations of the split job $(.)$, respectively. E.g., l' and l''. Let $s(a,b) \in R^2 \cup (+\infty,+\infty)$ be the location of the service station that gives minimal distance when UAV k flies from point a to that station and then to b, if the fuel level $Cq(k)$ is sufficient to reach the station (and then reach point b). If no such station exists, the function returns the point at $x=+\infty$, $y=+\infty$. That is, $s(a,b) = \text{argmin}_{s \in \Omega SE}\{D_{a,s} + D_{s,b} \mid Cq(k) \geq D_{a,s}/TS(k),\ q(k) \geq D_{s,b}/TS(k)\}$, if feasible, $(+\infty,+\infty)$ otherwise. If there are several such stations, chose one arbitrarily. We will have particular interest in $s(Cl(k),l')$. Let $s(a) = \text{argmin}\ s \in \Omega_{SE}\{D_{a,s}\}$ denote the location of the station nearest to a (selected arbitrarily if more than one).

Let $N_D(k)$ be the maximum number of split jobs that UAV k can sequentially serve for customer t, starting with l via a direct flight. That is, $N_D(k) = max\{n \in Z_+ \mid D_{Cl(k),l'} / TS_k + \sum_{i=l}^{l+n-1} Pi + D_{(l+n-1)'',s((l+n-1)'')} / TS_k) \leq Cq(k)\}$, or 0 if no such value exists. Here Z_+ is the non-negative integers. The indicator for feasibility of the direct flight we define as $D(k) = 1 - [\ I\{N_D(k) \geq 1\} * I\{A(k) + D_{Cl(k),l'} / TS_k \leq E_l\}]$. Where the indicator function I{.} is 1 if the condition {.} is true, and 0 otherwise.

Similarly, let $N_I(k)$ be the maximum number of split jobs that UAV k can sequentially serve for customer t, starting with l via an indirect flight. That is, $N_I(k)=max\{n \in Z_+ \mid D_{Cl(k),s(Cl(k),l')} / TS_k \leq Cq(k),\ D_{s(Cl(k),l'),l'} / TS_k + \sum_{i=l}^{l+n-1} Pi + D_{(l+n-1)'',s((l+n-1)'')} / TS_k \leq q_k \}$, or 0 if no such value exists or in the case of $s(Cl(k),l')=(+\infty,+\infty)$. The indicator for feasibility of the direct flight we define as $Ind(k) = 1 - I\{N_I(k) \geq 1\} * I\{D_{Cl(k),s(Cl(k),l')} / TS_k + U + (H/q_k) \cdot [q_k - (Cq(k) - D_{Cl(k),s(Cl(k),l')} / TS_k)] + D_{s(Cl(k),l'),l'} / TS_k < E_l - A(k)\}$.

Together, these feasibility indicators will be used to ensure that a particular UAV assignment for split job l is feasible for constraints (16-17). They also enforce constraints (14-15).

The values assigned to UAV k when we seek to assign split job l are as follows.

$$V_D(\mathrm{k}) = \{\alpha \cdot \omega(N_D(k)) - (1-\alpha) \cdot D_{Cl(k),l'}\} - M \cdot D(k) \quad (23)$$

$$V_I(\mathrm{k}) = \left\{\alpha \cdot \omega(N_I(k)) - (1-\alpha)\left(D_{Cl(k),s(kl')} + D_{s(kl'),l'} - M \cdot Ind(k)\right.\right. \quad (24)$$

M is a large positive value, α and ω are parameters to balance the terms. The UAV achieving the greatest value for $V_D(k)$ or $V_I(k)$ is selected to prosecute those jobs via that kind of flight.

This procedure is repeated until all split jobs have been assigned for a customer. Then, we proceed to the next customer. After every split job is assigned, all UAVs not at a service station travel to the nearest station. STAH is complete.

C. Receding Horizon Task Assignment

We modify the $RHTA_d$ in [20] for our problem. We eliminate the resource selection components of $RHTA_d$; our system design is fixed. We modify the replenishment time for a UAV at a service station to conform to equations (1-2). The detailed pseudo code is provided in Appendix 2. We modify the IP model inside the $RHTA_d$ to ignore resource selection. We delete constraints (4-5) in [20] and replace the objective function (1) in [20]:

$$Min \sum_{k=1}^{N_{UAV}} \sum_{p=1}^{N_{kp}} S_{kp} X_{kp} \quad (25)$$

IV. Real Time Operation

To achieve real time task allocation, we employ the MILP, STAH or RHTA in a rolling horizon approach. The optimization is used either when specific events occur or at fixed time intervals.

A. Event based real time operation

During the operation of a system of UAVs, many events may happen: new customer service requests may arrive, customers may change their planned path, UAV fuel may be consumed faster than anticipated, UAVs may fail, etc. Tasks must be allocated to the UAVs to account for these disturbances and ensure persistent UAV service. Our event-based rolling horizon method follows.

- **STEP 1:** Set customer index $i=0$ and finished service index $f=0$. If a customer service request arrives, set $i \leftarrow i+1$ and solve for UAV assignments via MILP, STAH or RHTA. Go to STEP 2.
- **STEP 2:** UAVs execute their tasks. If a customer service is complete, go to STEP 3. If a new customer request arrives, go to STEP 4. If any other unexpected event occurs, go to STEP 5.
- **STEP 3:** Set $f \leftarrow f+1$. (Note that there are i-f customers in the system.) Go to STEP 2.
- **STEP 4:** Set $i \leftarrow i+1$. For all remaining jobs (for customers i-f-l to i-1) and the new customer, solve for UAV assignments using the MILP, STAH or RHTA. Go to STEP 2.
- **STEP 5:** Update the UAV information (location, fuel level, availability) and solve for UAV assignments using MILP, STAH or RHTA. Go to STEP 2.

B. Time based real time operation

The time based approach is similar. Every small interval of time, if there is a deviation from the previously determined schedule, the UAV assignments are reallocated using MILP, STAH or RHTA.

V. Numerical Examples

We conduct numerical analysis of the MILP via CPLEX, STAH, RHTA and their use for real time operations. These studies use a personal computer with Intel(R) Core(TM)2 Quad CPU Q8400, 2.66 GHz and4.00 GB RAM.

Table 1. Split job information of example 1

Customer	Split job	Start point		End point		Start time
		x	y	x	y	
1	1	596	167	532	161	5
	2	532	161	483	129	6
	3	438	129	432	94	7
	4	432	94	372	91	8
	5	372	91	315	87	9
	6	315	87	262	74	10
	7	262	74	218	99	11
	8	218	99	171	134	12
	9	171	134	142	186	13
	10	142	186	102	225	14
	11	102	225	54	251	15
	12	54	251	6	266	16
2	13	458	64	479	105	8
	14	479	105	514	63	9
	15	514	63	536	15	10

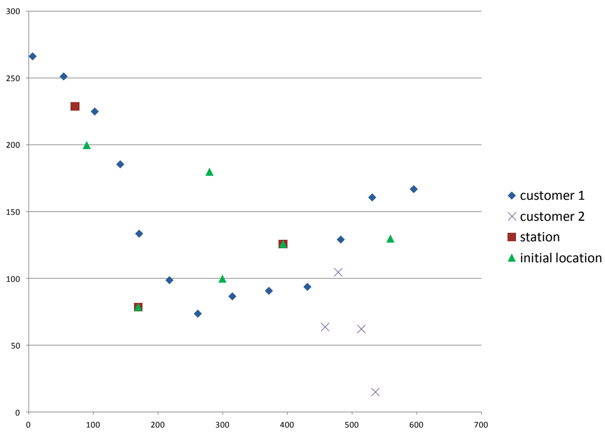

Figure 2. Geographical information of example 1.

Throughout we use $\alpha = 0.3$ and $\omega = 50$ in STAH.

A. Example 1: Scheduling two customers

Consider the layout illustrated in Figure 2. There, the location of service stations, initial UAV locations and customer paths are provided. The split job data is given in Table 1. Customer 1 requires 12 split jobs. Customer 2 requires 3 split jobs. Table 2 gives the initial location, maximum travel speed, initial battery (fuel) level and maximum traveling time of each UAV. The three stations are located at the x-y coordinates (394, 126), (170, 79) and (72, 229). The service station constants $H=3$ and $U=0.5$.

The MILP via CPLEX obtained an optimal solution with value 809.067 (meters) in 9.17 seconds. STAH obtained the same value (with an alternate optimal solution) in 0.006 seconds. RHTA obtained a solution with value 911.949 in 0.187 seconds; this is a 12.7 % gap.

B. Example 2: Real time operation

We now consider the system of Example 1 with a new customer arrival at time T=10. Customer 3 wishes to be escorted from T=13 to T=19. We assume a time-based rolling horizon and reallocate system tasks starting from T=11. All split jobs for customer 2 have been served by T=11. Table 3 provides the remaining split jobs for customer 1 (there are 6) and additional split jobs for customer 3 (there are also 6) from T=11.

Table 2. UAV information for Example 1

UAV	x	y	q_k	$q_{k,ini}$	TS_k
1	560	130	12	3	240
2	280	180	12	8	240
3	170	79	12	8	240
4	300	100	12	6	240
5	394	126	12	6	240
6	90	200	12	8	240

Table 3. Split job information at T=11

Customer	Split job	Start point		End point		Start time
		x	y	x	y	
1	1’	262	74	218	99	11
	2’	218	99	171	134	12
	3’	171	134	142	186	13
	4’	142	186	102	225	14
	5’	102	225	54	251	15
	6’	54	251	6	266	16
3	7’	413	210	430	184	13
	8’	430	184	454	162	14
	9’	454	162	482	137	15
	10’	482	137	451	10	16
	11’	451	10	442	82	17
	12’	442	82	428	65	18

Table 4 gives the location and fuel level of UAVs at T=11. UAV 1 and 5 are at station 1. UAV 2 is at the end point of split job 6 in Table 1 with 7.629 remaining fuel. UAV 3 is at station 2. UAV 4 is at the end point of split job 15 in Table 1 with 2.325 remaining fuel. UAV 6 is at station 3. Figure 3 depicts the layout.

The revised schedule assigns UAV 2 to the remaining split jobs of customer 1 (as in the original schedule). Instead of resting at station 1, UAV 5 now serves customer 3.

C. Various problem sizes

Consider Example 1. We vary the number of split jobs used for the same customer paths. Refer to Table 5. We set the initial fuel of each UAV ($q_{1,ini}$, …, $q_{10,ini}$) = (6,8,8,10,6,8,6,8,8,10). For the case of (N_J, N_{STA}, N_{UAV}) = (15, 3, 6), due to the higher level of initial fuel, the sum of

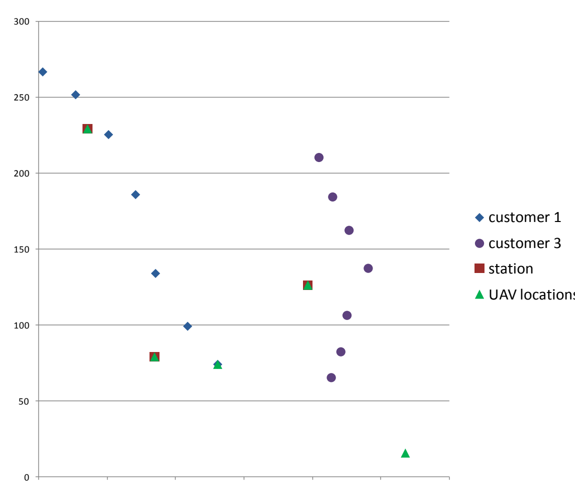

Figure 3. Geographical information at T=11

Table 4. UAV information at T=11

UAV	x	y	q_k	$q_{k,ini}$	TS_k
1	394	126	12	12	240
2	262	74	12	7.629	240
3	170	79	12	12	240
4	536	15	12	2.325	240
5	394	126	12	12	240
6	72	229	12	12	240

travelling distances for the UAVs is reduced. STAH and RHTA obtain very nearly optimal values. RHTA and STAH are about 25 times and 1300 times faster than the MILP via CPLEX.

For the cases with more than N_J = 15 split jobs, CPLEX issues an out of memory error; it cannot solve the problem. STAH and RHTA can derive feasible solutions. STAH is at least 24 times faster than RHTA in these examples. STAH generate superior solutions. STAH is a customized heuristic for a sequential set of split jobs while RHTA is a universal approach for common vehicle route problems.

VI. CONCLUDING REMARKS

One key component of persistent UAV systems is the automation of logistics operations. Logistics automation requires consumable replenishment service stations and algorithms to orchestrate the system activities. Here, we consider real time operations of a system of UAVs and service stations distributed over a field of operations. We consider missions whose customer missions provide uninterrupted UAV escort/surveillance.

We developed a MILP scheduling formulation of the problem that allows for arbitrary initial UAV locations and fuel levels. To address the computational complexity inherent in this formulation, we developed a heuristic – termed STAH – that is inspired by solutions observed in solutions of the MILP model. We extended an RHTA heuristic to allow for these arbitrary initial conditions. In our numerical studies, the RHTA was about 25 times faster than the MILP solved via CPLEX 12.4. STAH was about 24 times faster than the RHTA. For real time operation, the MILP, STAH or RHTA was incorporated in a rolling horizon task assignment approach. There, customer service requests may arrive continuously and there may be unexpected disturbances such as deviations from anticipated customer paths and fuel levels. The real time operation was studied via example.

There are several opportunities for future work. Capacitated service stations that can serve a limited number of UAVs at a time should be incorporated. A key caveat is that the MILP formulation presented here (which is carried over to the STAH and RHTA) requires that all split jobs be served. This should be generalized to allow infeasible job requests to be ignored.

REFERENCES

[1] T. Shima and C. Schumacher, “Assignment of cooperating UAVs to simultaneous tasks using genetic algorithm,” In Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, 2005, Paper no. AIAA-2005-5829.

[2] J. Zeng, X. Yang, L. Yang, and G. Shen, “Modeling for UAV resource scheduling under mission synchronization,” Journal of Systems Engineering and Electronics, vol. 21, no. 5, 2010, pp. 821-826.

[3] A. L. Weinstein and C. Schumacher, “UAV scheduling via the vehicle routing problem with time windows,” In Proc. AIAA Infotech@Aerospace 2007 Conference and Exhibit, Rohnert Park, California, 2007, Paper no. AIAA-2007-2839.

[4] Y.S. Kim, D.W. Gu, and I. Postlethwaite, “Real-time optimal mission scheduling and flight path selection,” IEEE Transactions on Automatic control, vol. 52, no. 6, 2007, pp. 1119-1123.

[5] B. Alidaee, H. Wang, and F. Landram, “A note on integer programming formulations of the real-time optimal scheduling and flight selection of UAVS,” IEEE Transactions on Control Systems Technology, vol.

Table 5. Numerical comparison on problem size

System parameters			CPLEX		STAH			RHTA		
N_J	N_{STA}	N_{UAV}	CPU time	Obj. value	CPU time	Obj. value	Gap	CPU time	Obj. value	Gap
15	3	6	5.42	662.570	0.004	665.261	0.406 %	0.203	664.857	0.345 %
30	3	6	N/A	N/A	0.008	1010.24	-	0.327	1274.991	-
45	3	6	N/A	N/A	0.015	1314.34	-	0.458	1901.816	-
45	5	10	N/A	N/A	0.015	1346.96	-	0.608	1656.667	-
60	3	6	N/A	N/A	0.016	1642.34	-	0.578	2677.189	-
60	5	10	N/A	N/A	0.031	1565.90	-	0.765	1997.992	-

17, no. 4, 2009, pp. 839-843.
[6] V. K. Shetty, M. Sudit, and R. Nagi, “Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles,” Computers and Operations Research, vol. 35, 2008, pp. 1813–1828.
[7] M. Valenti, D. Dale, J. How, and J. Vian, “Mission Health Management for 24/7 Persistent Surveillance Operations,” in AIAA Guidance, Control and Navigation Conference, Myrtle Beach, SC, August 2007.
[8] B. Bethke, M. Valenti, and J. How "UAV Task Assignment," Robotics & Automation Magazine, vol. 15, no. 1, 2008, pp. 39-44.
[9] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of Multiple UAVs for Persistent Surveillance: Algorithm and Flight Test Results," IEEE Transactions on Control Systems Technology, vol. 20, no. 5, 2012, pp. 1236-1251.
[10] A. Geramifard, J. Redding and J.P. How, “Intelligent Cooperative Control Architecture: A Framework for Performance Improvement Using Safe Learning,” Journal of Intelligent & Robotic Systems, vol. 73, issue. 1, 2013, pp.83-103.
[11] N. K. Ure, G. Chowdhary, J.P. How, M. A. Vavrina and J. Vian, “Health Aware Planning Under Uncertainty for UAV Missions with Heterogeneous Teams,” In Proc. European Control Conference, zurich, Switzerland, 2013, pp.3312-3319.
[12] N.K. Ure, G. Chowdhary, T. Toksoz, J.P. Hpw, M. A. Vavrina and J. Vian, “An Automated Battery Management System to Enable Persistent Missions With Multiple Aerial Vehicles,” IEEE/ASME Transactions on Mechatronics, vol. PP, issue. 99, pp.1-12.
[13] P. F. Kemper, K.A.O. Suzuki, J.R. Morrison, “UAV consumable replenishment: design concepts for automated service station,” Journal of Intelligent & Robotics system. vol. 61, issue. 1-4, 2011, pp.369-397.
[14] K.A.O. Suzuki, P.F. Kemper, J.R. Morrison, “Automated battery replacement system for UAVs: Analysis and design,” Journal of Intelligent & Robotics system. vol. 65, issue. 1-4, 2012, pp.563-586.
[15] K.A. Swieringa, C.B. Hanson, J.R. Richardson, J.D. White, Z. Hasan, E. Qian and A. Girard, “Autonomous battery swapping systems for small-scale helicopters,” In Proc. IEEE International Conference on Robotics and Automation(ICRA), Anchorage, Alaske, 2010, pp. 3335-3340.
[16] R. Godzdanker, M.J. Rutherford, K.P. Valavanis, “ISLAND: A Slef-Leveling landing platform for autonomous miniature UAVs,” In Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Budapest, Hungary, 2011, pp. 170-175.
[17] K. Fujii, K. Higuchi and J. Pekimoto, “Endless Flyer: A Continuous Flying Drone with Automatic Battery Replacement,” In Proc. IEEE International Conference on Ubiquitous Intelligence & Computing and IEEE International Conference on Autonomic & Trusted Computing, Sorrento Peninsula, Italy, 2013, pp. 216-223.
[18] J. Kim, B.D. Song, and J. R. Morrison, “On the scheduling of systems of heterogeneous UAVs and fuel service stations for long-term mission fulfillment,” Journal of Intelligent & Robotic Systems, vol. 70, no. 1 2013, pp. 347-359.
[19] B.D. Song, J. Kim, J. Kim, H. Park, and J. R. Morrison, “Persistent UAV service: An improved scheduling formulation and prototypes of system components,” Journal of Intelligent & Robotic Systems, vol. 74, issue. 1-2, 2014, pp.221-232.
[20] J. Kim and J. R. Morrison, “On the concerted design and scheduling of multiple resources for persistent UAV operations,” Journal of Intelligent & Robotic Systems, vol. 74, issue. 1-2, 2014, pp.479-498.
[21] J. Kim and D.H. Shim, “A vision-based target tracking control system of a quadrotor by using a tablet computer,” In Proc. International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, 2013, pp. 1165-1172.

APPENDIX

A1. Pseudo code of STAH

Let $MD_D(k)$ and $MD_I(k)$ be the moving distance by direct and indirect flight of UAV k. Let Obj.value be the total travelling distance of UAVs
Set Obj.value = 0, $MD_D(k)$ =0, $MD_I(k)$ =0 and t = 1.
For $P = t$, **do**
Let W be the number of performed split jobs of customer t and set W=0.
While $W < P_t$, **do**
For all UAV k, calculate $V_D(k)$ and $V_I(k)$. Select the biggest Value.
Let k' be the selected UAV.
IF $V_D(k')$ is selected, **do**
$\mathrm{MD_D(k)} \mathrel{+}= \left(\mathrm{D}_{Cl(k),l'}\right)$; \\\\ {Calculate moving distance of direct flight of UAV k'}
$\mathrm{Obj.value} \mathrel{+}= \mathrm{MD_D(k)}$;
$\mathrm{W} \mathrel{+}= \mathrm{N_D(k')}$;
$\mathrm{Cl(k')} = \left(l + \mathrm{N_D}(\mathrm{k'}) - 1\right)''$; \\\\ {updated UAV k' information}
$\mathrm{Cq(k')} = \mathrm{Cq(k')} - \mathrm{MD_D(k)} / \mathrm{TS(k')} - \mathrm{P} \cdot \mathrm{N_D(k')}$;
$\mathrm{A(k')} = \mathrm{L}_{l+\mathrm{N_D(k')}-1}$;
$l = l + \mathrm{N_D(k')}$; \\\\ {updated earliest start split job information}
$l' = (l + \mathrm{N_D(k')})'$;
$l = (l + \mathrm{N_D(k')})''$;
ELSE IF $V_I(k')$ is selected, **do**
$\mathrm{MD_I(k)} \mathrel{+}= \left(\mathrm{D}_{Cl(k),s(kl')} + \mathrm{D}_{s(kl'),l'}\right)$; \\\\ {Calculate moving distance of indirect flight of UAV k'}
$\mathrm{Obj.value} \mathrel{+}= \mathrm{MD_I(k)}$;
$\mathrm{W} \mathrel{+}= \mathrm{N_I(k')}$;
$\mathrm{Cl(k')} = \left(l + \mathrm{N_I}(\mathrm{k'}) - 1\right)''$; \\\\ {updated UAV k' information}
$\mathrm{Cq(k')} = \mathrm{q(k')} - \mathrm{D}_{s(kl'),l'} / \mathrm{TS(k')} - \mathrm{P} \cdot \mathrm{N_I(k')}$;
$\mathrm{A(k')} = \mathrm{L}_{l+\mathrm{N_I(k')}-1}$;
$l = l + \mathrm{N_I(k')}$; \\\\ {updated earliest start split job information}
$l' = (l + \mathrm{N_I(k')})'$;
$l = (l + \mathrm{N_I(k')})''$;
IF W=P_t, **break;**
t+=1;
End for
For all UAV k, **do** \\\\ {send UAVs to the stations which are not located in station}
IF Cl(k) $!\in \Omega_{SE}$
s(Cl(k)) = argmin $s \in \Omega_{SE}\{D\mathrm{Cl(k)}_{,s}\}$
$obj.value \mathrel{+}= \left(\mathrm{D}_{\mathrm{Cl(k)},\mathrm{S}_{\mathrm{cl(k)}}}\right)$;
End for

A2. Pseudo code of RHTA

Find the travel distance from split job i's finish point or station i to split job j's start point or station j using Euclidean distance (Set D_{ij})
Set input variables (# of UAV, # of job, # of station, maximum flight time, travel speed, process time of split jobs, recharge or replacement time, start time of split job, initial location of UAV, available time of UAV, remaining fuel time)
Set W = W_0 (the set of all job)
While W is not empty **do**
For all UAV k **do** p←1;
For all numbers n_c of jobs to visit n_c ← 1,…,P **do**
For all combinations C of nc jobs **do**
For all permutations i of jobs [w_1, …, w_{nc}] in C, with i ←1…n_c! **do**
For all station s **do**
if ($D(cL(k),w_1)+\sum_{i=1}^{i=n_c-1} D(w_i,w_{i+1})+D(W_{nc},s)+\sum_{i=1}^{i=n_c} P(w_i)\le rF(k)$ and $at(k)+D(cL(k),w_1)/TS(k)\le E(w_1)$ and $E(w_{i-1})+P(w_{i-1})+D(w_{i-1},w_i)/TS(k)\le E(w_i)$ for i=1,…,nc) **do**
$S_i \leftarrow D(cL(k),w_1)+\sum_{i=1}^{i=n_c} D(w_{i-1},w_i);$
$P_i \leftarrow [w_1,...,w_{n_c}];$
break;
i_{min} ← $argmin_i$ S_i; \\ {Choose the best feasible permutation} S_{vp}←S_{imin}; P_{vp}←P_{imin}; p ← p+1;
solve the optimization model to find minimum cost strategy
for all UAV k **do** \\ { assign selected job to UAV's job list}
if x_{vp} ==1 **do**
w_{opt}← P_{vp} (1); \\ {Pick the first job in the permutation}
M_k ← [$M_k w_{opt}$]; \\ {Adds the job to the mission list of UAV k}
rF(k) ← rF(k) – D(cL(k), w_{opt})/TS(k) – P(w_{opt}); \\ {update remaining fuel time of UAV k}
at(k) ← E(w_{opt})+P(w_{opt}); \\ {update available time}
cL(k) ← w_{opt}; \\ {update current location}
W ← W- w_{opt}\\ {remove the selected job from the list}
for all UAV k **do** \\ {send exhausted UAV to the nearest station}
for all w in W **do**
for all station s **do**
if (($D(cL(k),w)/TS(k)+P(w)+D(w,s)/TS\ge rF(k)$ or $D(cL(k),w)/TS(k)+at(k)\ge E(w)$) and $cL(k)\neq s$) **do**
Assign UAV to closest station (S_{min})
M_k←[M_k,s_{min}]; rF(k)←maxF(k); at(k)←at(k) + D(cl(k),s) + U + H·(maxF(k)-rF(k)) /maxF(k) ;cL(k)←smin ;
for all UAV k **do** // {send UAV at end of job to station}
if !(cL(k)==s) **do**
Assign UAV to closest station;