J Intell Robot Syst (2014) 74:479–498
DOI 10.1007/s10846-013-9958-8

On the Concerted Design and Scheduling of Multiple Resources for Persistent UAV Operations

Jonghoe Kim · James R. Morrison

Received: 10 September 2013 / Accepted: 14 September 2013 / Published online: 17 October 2013

Abstract A fleet of unmanned aerial vehicles (UAVs) supported by logistics infrastructure, such as automated service stations, may be capable of long-term persistent operations. Typically, two key stages in the deployment of such a system are resource selection and scheduling. Here, we endeavor to conduct both of these phases in concert for persistent UAV operations. We develop a mixed integer linear program (MILP) to formally describe this joint design and scheduling problem. The MILP allows UAVs to replenish their energy resources, and then return to service, using any of a number of candidate service station locations distributed throughout the field. The UAVs provide service to known deterministic customer space-time trajectories. There may be many of these customer missions occurring simultaneously in the time horizon. A customer mission may be served by several UAVs, each of which prosecutes a different segment of the customer mission. Multiple tasks may be conducted by each UAV between visits to the service stations. The MILP jointly determines the number and locations of resources (design) and their schedules to provide service to the customers. We address the computational complexity of the MILP formulation via two methods. We develop a branch and bound algorithm that guarantees an optimal solution and is faster than solving the MILP directly via CPLEX. This method exploits numerous properties of the problem to reduce the search space. We also develop a modified receding horizon task assignment heuristic that includes the design problem ($RHTA_d$). This method may not find an optimal solution, but can find feasible solutions to problems for which the other methods fail. Numerical experiments are conducted to assess the performance of the $RHTA_d$ and branch and bound methods relative to the MILP solved via CPLEX. For the experiments conducted, the branch and bound algorithm and $RHTA_d$ are about 500 and 25,000 times faster than the MILP solved via CPLEX, respectively. While the branch and bound algorithm obtains the same optimal value as CPLEX, $RHTA_d$ sacrifices about 5.5 % optimality on average.

Parts of this paper are reprinted from the paper of the same title in the 2013 Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS'13), May 2013.

The work reported here has been supported in part by KAIST HRHRP N10120008 and by the National Research Foundation of Korea (NRF) Basic Science Research Program grant 2013012292.

J. Kim · J. R. Morrison (✉)
Department of Industrial and Systems Engineering
KAIST, 373-1 Guseong-dong, Yuseong-gu,
Daejeon 305-701, Republic of Korea
e-mail: james.morrison@kaist.edu
URL: http://xS3D.kaist.edu

J. Kim
e-mail: windf@kaist.ac.kr

Keywords Persistent UAV service · Concerted design and scheduling · Replenishment stations · Branch and bound algorithms · UAV scheduling for persistence

1 Introduction

Systems of unmanned aerial vehicles (UAVs) have the potential to serve in many roles such as border patrol and security escort. However, long term persistent operations require both UAVs and logistics resources, such as automated service stations. Guided by orchestration algorithms, UAVs may return to the field after replenishing their consumables at automated service stations distributed throughout the field. We envision that future UAV systems will operate in this manner and provide indefinite service to their customers.

To deploy a complex system of UAVs, logistics resources and management software entails numerous stages of development. Two key stages are system design and resource scheduling. The system design stage determines the number and location of physical resources, such as UAVs and service stations. The resource scheduling stage orchestrates the operations of these resources to achieve the mission objectives. Traditionally, these stages are conducted sequentially. However, there is inherent loss of efficiency associated with this hierarchical approach. We select the resources and schedules in concert via a mixed integer linear program (MILP), branch and bound algorithm and receding horizon task assignment heuristic in the context of persistent UAV operations.

Figure 1 depicts an example system. There are three service station candidates, four UAV candidates and three tasks. The missions are to follow three objects along their deterministic known time-space trajectories without interruption. Objects 2 and 3 are stationary. The candidate service stations are geographically distributed; they can be used by any UAV. Both the stations and UAVs are resources that may or may not be selected for use. One feasible solution is for UAV 1 to start its first flight from station 1 to observe object 2 and return to station 2 for replenishment. After replenishment, it flies to object 3 and then returns to station 2. Object 1's path duration is longer than a UAV's maximum travel time. UAV 2 serves object 1 and then hands off the duty to UAV 3, who completes the mission. UAV 4 and station 3 are not required and need not be included as resources in the system.

In general, we will consider N UAV candidates (each with different capabilities), M refueling station candidates and J trajectories. There may be many customer missions prosecuted simultaneously. Each mission may be served by several

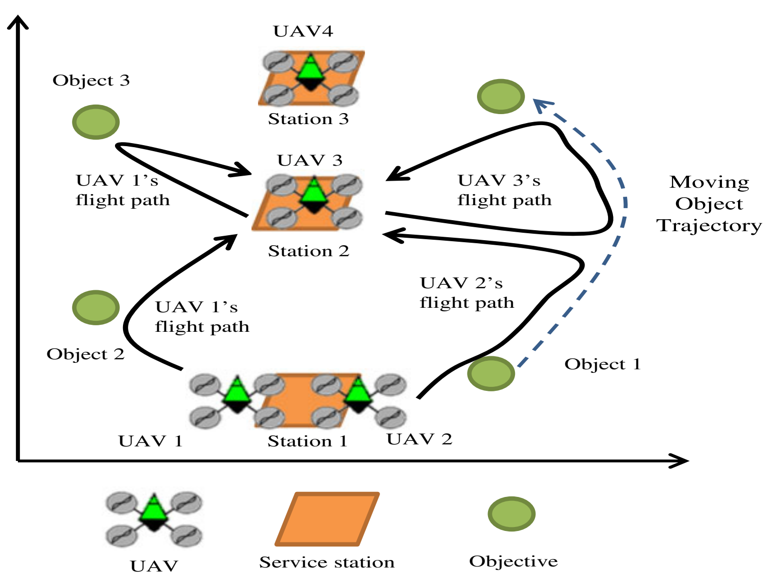

Fig. 1 A system of three service station candidates, four UAV candidates and three tasks

UAVs. The UAVs can conduct multiple tasks between visits to the stations. For such a complex problem as optimally selecting the resources to employ and obtaining a schedule for them, one requires a formal approach.

Note that UAV technologies, such as battery replacement systems (c.f., [1–3]), have been developed to support persistent operation.

1.1 Literature on System Design

Resource selection (system design) methods for UAVs have been developed to minimize cost and ensure a desired service level. The placement of recharge stations for capacitated UAVs was investigated in [4, 5]. In [4], the authors seek to locate a fixed number of service stations on an n by n grid by solving the p-median problem. There is no consideration of flight path. Determining station locations while considering the UAV flight path was studied in [5]. Their objective was to observe all discrete sections of a given region. First, algorithms were developed to efficiently observe the area without consideration of UAV flight duration. Then, service stations were placed every d units of distance along these paths. The area may also be segregated into disjoint areas and served separately. Time constraints on the missions are not considered and the UAVs are identical. A Petri net to find the number of UAVs and service stations to maintain a desired number of UAVs in flight was developed in [2]. In [6], an analytic approach to determine the number of identical automated guided vehicles (AGVs) was proposed. The AGVs deliver cargo between two points with a target service level.

1.2 Literature on UAV Scheduling

Many authors have addressed UAV scheduling. For example, the authors in [7] studied scheduling methods for UAVs without fuel limitations. A mathematical model was developed to direct multiple UAVs for cooperative engagement of moving ground targets. A genetic algorithm was used to obtain feasible solutions.

Research on capacitated UAVs with finite flight capability was investigated in [8–12]. When their fuel is exhausted, the UAVs must return to base and stay there; they do not return to the field. A nonlinear mathematical model to allocate capacitated UAV resources to the battlefield was developed in [8]. A MILP based on the vehicle routing problem for capacitated UAVs, including time windows for the jobs, was investigated in [9]. Kim et al. [10] developed a MILP to assign m identical UAVs, with a flight capacity q each, to n tasks. UAVs should return to where they departed from. They consider MILP models for two cases: no UAV return and UAV return. Simplified, suboptimal, MILP models are considered for computational tractability. A MILP with fewer variables and constraints based on [10] was introduced in [11]. Shetty et al. [12] used a MILP to assign UAVs to tasks with the goal of maximal target coverage. They consider payload, maximum range and service level. Alidaee et al. [13] improved the tractability of the MILP from [12].

Persistent operations using service stations were investigated in [3, 14, 15]. There, the service stations were not geographically distributed (there is a single multiport station). An indoor system prototype was implemented and tested in [3, 14, 15]. In [16], scheduling for persistent operations with geographically distributed service stations was considered. Their MILP assumed a fixed initial configuration of resources. Song et al. [17] reduced the number of variables and constraints for the MILP in [16].

1.3 Contribution and Organization

Scheduling methods for UAVs assume a fixed number of UAVs and fixed numbers/locations of stations. It is our purpose to jointly address the UAV scheduling and system design problems. A few efforts have considered UAV system design. However, these do not address the persistent UAV problem with generic missions possessing time constraints.

Here, we develop methods to simultaneously select the resources, their locations and schedules. We consider persistent operation with a fleet of UAVs supported by shared service stations which are geographically distributed. Multiple missions occur at the same time. Each one can be served by different UAVs. UAVs can be assigned several tasks between visits to the stations. To our knowledge, this is the first such effort to jointly optimize UAV system design and schedule with persistence.

The contributions are as follows. For what is to our knowledge the first time, we:

- Develop a MILP to optimally select resources (UAVs, service stations and their locations) and schedules in the persistent context. Capacitated UAVs are allowed to return to service after a visit to any of the shared service stations distributed across the field. For each of the given time-space trajectories, one UAV must be assigned at all times.
- Develop a modified receding horizon task assignment heuristic to address the computational complexity of the MILP.
- Develop a branch and bound algorithm that reduces the computational time relative to the MILP via CPLEX while guaranteeing an optimal solution.

Despite that we hope to improve the efficiency of the overall system by jointly determining the system resources and their schedules, there are concerns associated with this approach. First, the combined problem is significantly more computationally complex than either optimization when conducted independently. However, as will be demonstrated, effective heuristics can be developed to significantly reduce the computation required. Second, we assume known deterministic mission paths. As such, the proposed approach will be best for highly structured customer missions, e.g., border patrol. Our MILP formulation is appropriate for these deterministic assumptions. However, in the presence of real world uncertainties, a Markov Decision Process (MDP) formulation may be more appropriate. It would explicitly model randomness. Practically, a MILP could be used for real time control by taking a rolling horizon perspective and allowing arbitrary system state for the initial condition. Our MILP would have to be extended to allow such arbitrary initial condition. (Naturally, real time use would assume a fixed design.) The $RHTA_d$ with fixed design can be used for real time task allocation. However, it would be very interesting to investigate explicit methods to address this problem with probabilistic customer demands.

The rest of the paper is organized as follows. In Section 2, we develop the MILP. Section 3 develops the modified receding horizon task assignment heuristic. The branch and bound algorithm is provided in Section 4. Our numerical studies are discussed in Section 5. We provide concluding remarks and future directions in Section 6.

2 MILP for UAV Scheduling and Design

We develop a MILP to select resources for and schedule a fleet of capacitated UAVs. The UAVs must provide persistent uninterrupted service across the field to deterministic missions as in [17].

We discretize the time-space trajectories (missions) into segments called "split jobs"; refer to Fig. 2. Since the trajectories are deterministic, each split job has known start/end times and locations. Each split job has a start and end coordinate denoted as (x_s,y_s) and (x_e,y_e), respectively. These depend on how the trajectories are discretized. Thus the number of split jobs and their locations/times are determined up front and are given as input data for the MILP. If many split jobs are used, the MILP's optimal value may be improved at the cost of increased computational complexity.

Let i and j be the index for split jobs, s the index for replenishment stations, k the index for UAVs, N_J the number of split jobs, N_{UAV} the number of UAV candidates in the system, N_{STA} the number of service station candidates, N_R the maximum number of flights per UAV during the time horizon and M a sufficiently large number. During service, the UAVs take off from a station and enter the field to process split jobs. After serving its designated split jobs, it returns to any of the stations. This "station to split jobs to station" travel is defined as one UAV flight. UAVs can have multiple flights during the horizon. We use r to index a UAV's r^{th} flight.

Let D_{ij} be the travel distance from the end point of split job i or station i to the start point of split job j or station j. We use the standard Euclidean

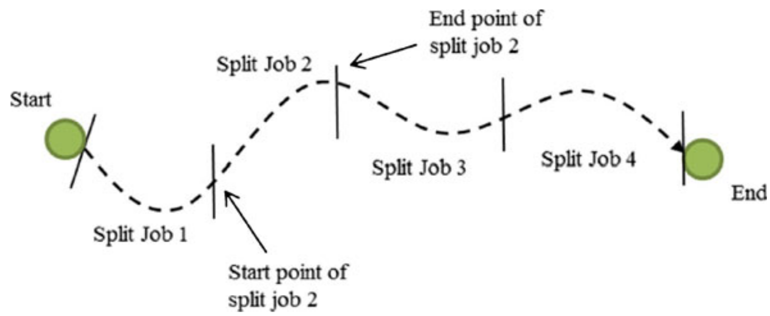

Fig. 2 Split jobs for a moving target or patrol path

distance between points. Note that D_{ij} need not equal D_{ji}. We use P_i to denote the process time of split job i or replenishing time at station i. The UAV replenishing time at a station is constant. Let E_i be the start time of split job i (and thus the end time is $E_i + P_i$), q_k the maximum flight time of UAV k, S_{ok} the initial station location of UAV k, TS_k the travel speed of UAV k, C_k the purchase cost of UAV k, C_s the purchase cost of station s and W_d the cost of UAV travel per unit distance.

Assume that all UAVs start with no fuel and immediately replenish at their initial station (this is convenient for our model). UAVs expend no fuel when waiting (it can stand by on the ground). All input parameters such as N_J, N_{UAV}, N_{STA}, N_R, D_{ij}, P_i, E_i, q_k, C_k, C_s, W_d, S_{ok} and TS_k are deterministic. The candidate station locations are all given and fixed.

The notation for indices is given as follows:

- UAV flight index: r in $R = \{1, \ldots, N_R\}$;
- UAV index: k in $K = \{1, \ldots, N_{UAV}\}$;
- Split job index: i, j in $\Omega_J = \{1, \ldots, N_J\}$;
- Set of UAV flight start recharge stations: $\Omega_{SS} = \{N_J{+}1, N_J + 3, \ldots, N_J + 2 \cdot N_{STA} - 1\}$;
- Set of UAV flight end recharge stations: $\Omega_{SE} = \{N_J + 2, N_J + 4, \ldots, N_J + 2 \cdot N_{STA}\}$;
- Set of all job and recharge stations: $\Omega_A = (\Omega_J \text{ U } \Omega_{SS} \text{ U } \Omega_{SE}) = \{1, \ldots, N_J + 2 \cdot N_{STA}\}$.

The decision variables are as follows:

- $X_{ijkr} = 1$ if UAV k serves split job j or replenishes at station j after processing split job i or replenishing at station i during the r^{th} flight; 0, otherwise.
- C_{ikr} is job i's start time by UAV k during its r^{th} flight or UAV k's replenishment start time at station i; otherwise its value is 0.
- $U_k^{UAV} = 1$ if UAV candidate k is selected to be used; 0, otherwise.
- $U_s^{STA} = 1$ if station candidate s is selected to be used; 0, otherwise.

It is convenient to allocate two indices to each station. Station s is assigned indices $N_J + 2s - 1$ and $N_J + 2s$. The first is the start station index in Ω_{SS}; it is used when a UAV starts its flight from station s. The second is the end station index in Ω_{SE}; it is used when a UAV ends its flight at station s.

We adopt the MILP from [17] for the scheduling component, modify their objective function and add constraints (22), (23) and (24). Our MILP is provided in Appendix 1.

The objective function (9) seeks to minimize the total costs: UAV travel costs, UAV purchase costs and station purchase costs. Constraint (10) guarantees that all UAVs start their first flight from their initial station. Constraints (11)–(14) are service station constraints. Constraint (11) ensures that UAV k flies to split job j in Ω_J U Ω_{SE} from a station every flight. Constraint (12) guarantees that UAV k finishes its flight at one and only one service station per flight. Constraint (13) ensures that UAV k's end station on its r^{th} flight and start station on its $(r + 1)^{th}$ flight are identical. Constraint (14) implies that the finish time of a UAV's r^{th} flight is equal to the start time of a UAV's $(r + 1)^{th}$ flight at that same station. Constraints (15)–(17) are split job assignment constraints. Constraint (15) guarantees that all split jobs in Ω_J receive service. Constraint (16) ensures that a UAV not finish its flight at a split job. Constraint (17) prevents each UAV from finishing a flight at a start station s with index in Ω_{SS}. (This is just a notational issue related to each station having two indices.)

Constraint (18) gives the relationship between the start time of split job i or station i and the start time of its successor served by UAV k during its r^{th} flight. Constraints (19)–(20) ensure that every split job in Ω_J is served at its pre-determined start time. Constraint (21) forces UAVs to return to a station before their fuel is depleted. Constraint (22) ensures that only selected (purchased) UAVs serve split jobs. Constraints (23) and (24) ensure that only selected (purchased) stations are used. Constraints (25) and (26) define the ranges of the decision variables. If we set $N_R = 1$ (the maximum number of flights allowed), constraints (13) and (14) should be deleted.

3 Modified Receding Horizon Task Assignment

We employ a modified receding horizon task assignment (RHTA) heuristic to address the computational complexity inherent in the MILP. The RHTA was first developed in [18]. We pursue an RHTA approach because it is commonly employed in other contexts and has been used successfully for UAVS. For persistent UAV operations, it was first extended in [10]. Their formulation allows for only a single (possibly multiport) service station. There is no consideration of the resource selection problem.

RHTA is an iterative method that breaks a large problem into smaller parts. A reduced complexity IP is used in the smaller parts. For a given instant in time, let rF(k) be the remaining flight time of UAV k, at(k) the available time of UAV k (when it will be finished its current assignment), P the maximum petal size, M_k the split job and station sequence that is visited by UAV k, uUAV(k) the usage of UAV candidate k, uSTA(s) the usage of station candidate s and W the split job list to be assigned. A petal is a sequence of split jobs to be served by a UAV. For example, the petal {3, 4, 5} indicates that a UAV will serve split jobs 3, 4 and 5 in that order. Petal {4, 3, 5} is different. Since our RHTA includes system design as well as scheduling, we call it $RHTA_d$. The detailed pseudo-code is provided in Appendix 2.

The overall procedure follows:

- STEP 1. Enumerate all feasible petals for each UAV k and calculate the required travel distance for each. A petal can contain up to P split jobs. A petal is feasible for UAV k if it can serve the sequence of split jobs and return to a service station. (Lines 5–14 in Appendix 2.)
- STEP 2. Solve a single IP for all UAVs to minimize the travel and resource costs. The IP selects up to one petal per UAV. (Line 15 in Appendix 2; see Eqs. 1–5 below.)
- STEP 3. For all UAV k, assign the first split job of the selected petal to UAV k's M_k. Update assigned UAV k's remaining fuel and next available time. Remove the assigned split job from W. (Lines 16–25 in Appendix 2.)
- STEP 4. Send UAVs that do not have any feasible petals to a service station; update their fuel and next available times. Return to STEP 1 if W is not empty. (Lines 26–32 in Appendix 2.)
- STEP 5. Assign any UAVs not located at a station to a station. (Lines 33–36 in Appendix 2.)

STEP 2 requires an IP. Let p be the petal index, S_{kp} the required travel distance of petal p by UAV k and N_{kp} the total number of feasible petals of UAV k. A_{kjp} indicates whether split job j is in petal p of UAV k. $A_{kjp} = 1$, if split job j is in petal p of UAV k; 0, otherwise. Decision variable $X_{kp} = 1$, if UAV k selects petal p; 0, otherwise. The remaining notation is as before.

The IP for Step 2 follows:

$$Min\ W_d \cdot \sum_{k=1}^{N_{UAV}} \sum_{p=1}^{N_{kp}} S_{kp} X_{kp} + \sum_{k=1}^{N_{UAV}} C_k U_k^{UAV} + \sum_{s=1}^{N_{STA}} C_s U_s^{STA} \quad (1)$$

subject to

$$\sum_{k=1}^{N_{UAV}} \sum_{p=1}^{N_{kp}} A_{kip} X_{kp} \le 1 (i \in W) \quad (2)$$

$$\sum_{i \in W} \sum_{k=1}^{N_{UAV}} \sum_{p=1}^{N_{kp}} A_{kip} X_{kp} \geq P \quad (3)$$

$$\sum_{p=1}^{N_{kp}} X_{kp} = U_k^{UAV} \; (k = 1, ..., N_{UAV}) \quad (4)$$

$$U_{s_{ok}}^{STA} \geq U_k^{UAV} \; (k = 1, ..., N_{UAV}) \quad (5)$$

The objective function (1) is to minimize the total system cost. Constraint (2) ensures that split job i in W is selected at most one time in all petals and UAVs. Constraint (3) requires that at least P split jobs be selected. Constraints (4) and (5) ensure that only selected (purchased) UAVs and selected (purchased) stations provide service. Constraint (5) states that only selected stations can serve as initial UAV locations.

4 Branch and Bound Method

We develop a branch and bound (B&B) algorithm that reduces the computation time of the MILP model while guaranteeing an optimal solution. In the worst case, the B&B explores all feasible nodes to obtain a solution. Several properties of the problem are exploited to reduce the search space. The B&B seeks to solve the exact same problem as the MILP model. Our B&B uses a breadth first search approach because the properties we exploit to reduce the search space compare and eliminate nodes at the same level of the B&B tree. The detailed pseudo code of the B&B algorithm is provided in Appendix 3; the properties that it exploits are detailed in this section.

4.1 Basic Notation

Node N_a is an $N_J \times 2$ matrix, with elements denoted as n_{ij}. Here n_{i1} indicates the UAV ID assigned for split job i immediately after that UAV replenishes at station n_{i2}. $n_{i1} \in \{0,1,\ldots,N_{UAV}\}$ and $n_{i2} \in \{0, N_J + 1, \ldots, N_J + N_{STA} + 1\}$. (Note that we only require and use a single index for each station here.) If n_{i2} is $N_J + N_{STA} + 1$, the UAV is assigned split job i without replenishment immediate prior to serving job i. The case $n_{i2} = 0$ is an initial value that will be removed after the algorithm is complete. We use $n_{i1}(a)$ and $n_{i2}(a)$ as the value of the n_{i1} and n_{i2} elements of node (matrix) N_a. N_0 is the initial node of the B&B tree, and is the matrix of 0 elements. These 0's indicate that no UAV is assigned for any split job. In this node, all UAVs are located at their initial station. There are $N_J + 1$ levels in the B&B tree. One split job is assigned when passing from one level to the next. At level L, L split jobs have been assigned.

Note that node N_a contains information only on the assigned split jobs, assigned UAVs and previously visited stations. It excludes the decision variable C_{ikr} of the MILP model. The entries in each node are integer valued. As a consequence, the number of nodes will be finite.

To illustrate the notation, consider an example.

Example 1 We consider a system with two stations, one UAV and three split jobs. Table 1 provides the split job data. Stations 1 and 2 are located at coordinates (50, 50) and (160, 50), respectively. The UAV's maximum flight time is 8 min, maximum flight speed is 80 m/min and fuel replenishment at a station requires 1 min. The candidate UAV is initially located at station 1. W_d, C_k and C_s are \$1/m, \$50 and \$40, respectively.

Figure 3 shows a subset of the nodes that will be generated in the B&B tree for Example 1. Nodes N_a, N_b and N_c are child nodes of N_0. N_d, N_e and N_f are child nodes of N_b.

It will be convenient to consider slightly different notation for the set of split jobs and stations. As before, we will use $\Omega_J = \{1, 2, \ldots, N_J\}$ as the set of split job indices. Let $U_J(a)$ be the set of split job indices for jobs that have not yet been assigned in node N_a. That is,

$$U_J(a) = \{l \in \Omega_J | n_{l1}(a) = 0\}.$$

Table 1 Split job information for example

Split job	Start point	End point	Process time (min)	Split job start time (min)
1	50,130	100,130	3	1
2	210,130	260,130	8	1
3	260,130	310,130	9	1

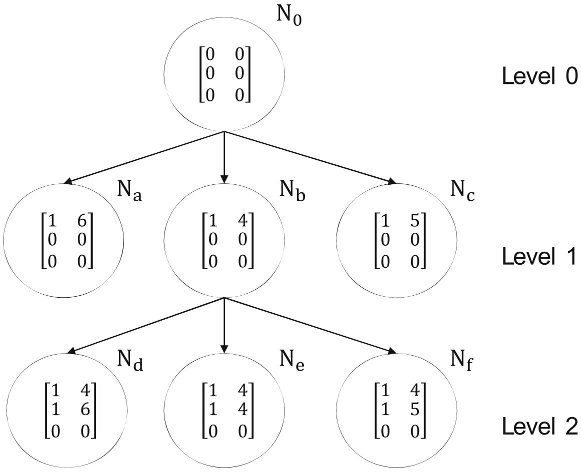

Fig. 3 Subset of nodes for the B&B tree in Example 1

For our B&B method, we will use a single label for each station. Let $\Omega_S = \{N_J + 1, \ldots, N_J + N_{STA}\}$ be the set of these station indices.

Every node in the B&B tree has attributes associated with it. When creating the B&B tree, new nodes are generated from existing nodes. The new nodes are referred to as child nodes; the node from which they derive is called the parent node. The attributes of the children (as well as their feasibility) are determined based on the attributes of their parents.

4.2 Node Attributes

To each node we associate several attributes:

- the objective function value,
- remaining flight time for each UAV,
- location of each UAV,
- available time for each UAV, and
- ST (the sequence of split jobs and stations visited by all UAVs) for each UAV.

These attributes can be obtained from the attributes of the parent node via simple calculations and based on the UAV allocated to the split job selected in that node.

As will be discussed in detail in Section 4.3, the next level of the B&B tree is created by generating child nodes for each node at the lowest level of the tree (these are called the parent nodes to those child nodes). We require some notation. We use $N_{a'}$ to denote a child of node N_a. Each child node is generated by assigning a UAV to a split job which was not assigned in its parent node. Excepting the row associated with this new split job assignment, N_a is identical to $N_{a'}$. The new row in $N_{a'}$, we call it the j^{th} row (corresponding to the new assignment of split job j), has elements $n_{j1}(a') = k'$ and $n_{j2}(a') = s'$ (for k' a UAV and s' a station).

Let $t_{avail}(k,a)$ be the available time of UAV k to begin its next travel to a split job or station in N_a. Let $t_{tra}(k,a,b)$ be the travel time of UAV k from location a to location b, a' be the index of a child node of node N_a, loca(k,a) be the location of UAV k after serving a split job or replenishing in node N_a and rF(k,a) be the remaining flight time of UAV k in N_a. Let obj(a) be the objective function value of node N_a. Let S(a) denote the collection of stations that have been opened (purchased for service) in node N_a. That is, S(a) is the set of station indices s such that $n_{i2}(a) = s$, for some i. For example, in node N_f of Fig. 3, S(f) = {4,5} (since 0 is not a station). (Note that, in that example, the index 6 is also not a station.). Similarly, K(a) is the set of UAV indices k such that $n_{i1}(a) = k$, for some i. cSTA(a') = $C_{s'}$ if s'$\notin$ S(a); 0, otherwise. cUAV(a') = $C_{k'}$, if k'$\notin$ K(a); 0, otherwise. The sequence of split jobs and stations that are visited by all UAVs are recorded as the attribute $ST_k(a)$. It is the sequence of split job and station indices visited by UAV k from N_0 to N_a. Table 2 provides the ST_k of node N_f.

Consider any child of node N_a, call it $N_{a'}$, the calculations to determine its attributes are given in Appendix 4. When the station s' in $N_{a'}$ does not appear in N_a (there is no i for which $n_{i2}(a) = s'$), that station is newly purchased in child node $N_{a'}$. Similarly, for UAV k'.

Example 1 revisited Consider Example 1. Some attributes of the nodes in Fig. 3 are provided in Table 3. For example, the obj(b) and rF(1,b) is 170 and 6, respectively. The obj(d) is calculated by adding additional costs to obj(b). That is, we

Table 2 $ST_k(f)$

UAV	Sequence of tasks
1	4,1,5,2

Table 3 Attributes of nodes

Node	Obj. value	rF	Feasibility
N_0	0	0	Y
N_a	N/A	N/A	N
N_b	170	6	Y
N_c	N/A	N/A	N
N_d	280	3.625	Y
N_e	443	4.7625	Y
N_f	404	5.825	Y

add the travel distance from split job 1 to split job 2 which is 110m. (Dummy station 6 has no cost.) rF(1,d) is calculated by subtracting the additional fuel consumption from rF(1,b). This consumption is due to travel from split job 1 to split job 2 and the processing time of split job 2.

4.3 Generation of Child Nodes

Starting from node N_0, child nodes are generated as follows. The child node of node N_a, call it $N_{a'}$, is constructed by replacing some row j of N_a with the property that $n_{j1}(a) = 0$ and $n_{j2}(a) = 0$. Otherwise all rows are identical in N_a and $N_{a'}$. The new row of $N_{a'}$ has some value $n_{j1}(a') = k'$ and $n_{j2}(a') = s'$ (for some UAV k' and some station s').

Child nodes generated from their parent node should satisfy two tests (based on their attributes) to be considered feasible and included in the tree: a start time constraint and a remaining fuel constraint. Let s*(j) be the index of the station closest to the end point of split job j. (Any one will do if there are ties, as we only use it to check for sufficient fuel and not to assign the next station.)

The start time constraint of node $N_{a'}$ is satisfied if either of the following conditions is satisfied:

- Condition 1 (C1): $s' = N_J + N_{STA} + 1$ and $t_{avail}(k',a) + t_{tra}(k',loca(k',a),j) \le E_j$, or
- Condition 2 (C2): $N_J + 1 \le s' \le N_J + N_{STA}$ and $t_{avail}(k',a) + t_{tra}(k',loca(k',a),s') + P(s') + t_{tra}(k', s', j) \le E_j$.

The fuel constraint is satisfied if either of the following conditions is satisfied:

- Condition 3 (C3): $s' = N_J + N_{STA} + 1$ and $rF(k',a) - t_{tra}(k',loca(k',a),j) - P(j) - t_{tra}(k',j,s^*(j)) \ge 0$, or
- Condition 4 (C4): $N_J + 1 \le s' \le N_J + N_{STA}$, $rF(k',a) - t_{tra}(k',loca(k',a),s') \ge 0$ and $q(k') - t_{tra}(k', s', j) - P(j) - t_{tra}(k',j,s^*(j)) \ge 0$.

If $N_{a'}$ does not satisfy conditions 1 or 2, every descendant of $N_{a'}$, at the last level of the B&B tree, is infeasible for the MILP. This immediately follows since the newly assigned split job j has start time E(j) and UAV k' is not available in time to serve it. Further, split job j will never be assigned to another UAV as the B&B process proceeds.

If $N_{a'}$ does not satisfy conditions 3 or 4, every descendant of $N_{a'}$, at the last level of the B&B tree, is infeasible for the MILP. This immediately follows since $rF(k',a') < t_{tra}(k',j,s^*(j))$; the UAV does not possess sufficient fuel to reach any station from the end point of split job j. Further, the remaining B&B process will not correct this.

We refer to an intermediate node (prior to the last level N_J), as level-feasible if it satisfies the start time constraint (either C1 or C2) and the fuel constraint (C3 or C4). In the last level of the B& B tree, if a node satisfies these constraints we call it MILP-feasible. Since each UAV can be sent to a station, MILP-feasible nodes provide feasible solutions to the MILP formulation. (The optimal assignment of UAVs to stations after all split jobs have been assigned is conducted at the end of the B&B algorithm).

4.4 Structural Properties to Reduce the Search Space

Here we develop several properties that help to reduce the computation via reduction of the search space. Let C_{b-a} be the cost of stations that are opened at node N_a and not opened at node N_b. For example, if stations 2, 4 and 5 are opened in node N_a and stations 3 and 4 are opened in node N_b, then C_{b-a} is $C_2 + C_5$ (not including C3).

Definition 1 For a given level in the B&B tree, we say that node N_a is **dominated** by node N_b if

$$n_{i1}(a) = n_{i1}(b), \quad \forall i \in \{1, ..., N_J\}, \tag{6}$$

$$rF(k, a) \le rF(k, b), \quad \forall k \in \{1, ..., N_{UAV}\}, \tag{7}$$

$$obj(a) \geq obj(b) + C_{b-a}. \quad (8)$$

Node N_b is said to **dominate** node N_a.

Lemma 1 *If N_a is dominated by N_b, then every feasible child of N_a is dominated by some feasible child of N_b.*

Proof Consider any level-feasible child of node N_a, call it $N_{a'}$. Let j be the split job that is assigned in $N_{a'}$ but not N_a (the new split job assigned when creating the child). For this split job j (row j), let $n_{j1}(a') = k'$ and $n_{j2}(a') = s'$. There is also a child of N_b, call it $N_{b'}$, with $n_{j1}(a') = k'$ and $n_{j2}(a') = s'$ (since split job j is unassigned in N_b by Eq. 6 and the child generation stage exhaustively creates all possible remaining split job assignments; Lines 11–20 in Appendix 3). Thus, since $n_{i1}(a) = n_{i1}(b)$ for all i by Eq. 6 and $n_{j1}(a') = n_{j1}(b') = k'$, we have $n_{i1}(a') = n_{i1}(b')$ for all i. That is, Eq. 6 is true for $N_{a'}$ and $N_{b'}$.

That the child $N_{b'}$ is level-feasible (it satisfies the start time and fuel constraints) can be readily shown since $t_{avail}(k',a) = t_{avail}(k',b)$ and $loca(k',a) = loca(k',b)$ by Eq. 6, $n_{j1}(b') = n_{j1}(a') = k'$, $n_{j2}(b') = n_{j2}(a') = s'$ and $rF(k',a) \leq rF(k',b)$ by Eq. 7.

We now show Eq. 7 for the children. Since $rF(k',a) \leq rF(k',b)$ by Eq. 7, $loca(k',a) = loca(k',b)$ by Eq. 6, $n_{j1}(b') = n_{j1}(a') = k'$ and $n_{j2}(b') = n_{j2}(a') = s'$, and using the definition of the child node attributes, we have $rF(k',a') \leq rF(k',b')$. Since UAV k' was the only one assigned when passing from node N_a (N_b) to $N_{a'}$ ($N_{b'}$), all other UAVs' fuel levels do not change, so that $rF(k,a') \leq rF(k,b')$ by Eq. 7 for all $k \neq k'$. Therefore, $rF(k,a') \leq rF(k,b')$ for all k.

We prove Eq. 8 for the children. There are five cases and for all cases $cUAV(a') = cUAV(b')$ by Eq. 6 and $n_{j1}(a') = n_{j2}(b') = k'$.

- Case I: $n_{j2}(a') = n_{j2}(b') = s' = N_J + N_{STA} + 1$. Since $loca(k',a) = loca(k',b)$ by Eqs. 6, 8 and $C_{b'-a'} = C_{b-a}$, we have $obj(a) + D_{loca(k',a),j} = obj(a') \geq obj(b') + C_{b'-a'} = obj(b) + C_{b-a} + D_{loca(k',b),j}$.
- Case II: $N_J + 1 \leq n_{j2}(a') = n_{j2}(b') = s' \leq N_J + N_{STA}$, $s' \in S(a)$ and $s' \in S(b)$. Since $loca(k',a) = loca(k',b)$ by Eqs. 6, 8, $n_{j2}(b') = n_{j2}(a') = s'$ and $C_{b'-a'} = C_{b-a}$, we have $obj(a) + D_{loca(k',a),nj2(a')} + D_{nj2(a'),j} = obj(a') \geq obj(b') + C_{b'-a'} = obj(b) + C_{b-a} + D_{loca(k',b),nj2(b')} + D_{nj2(b'),j}$.
- Case III: $N_J + 1 \leq n_{j2}(a') = n_{j2}(b') = s' \leq N_J + N_{STA}$, $s' \notin S(a)$ and $s' \notin S(b)$. Since $loca(k',a) = loca(k',b)$ by Eqs. 6, 8, $n_{j2}(b') = n_{j2}(a')$ and $C_{b'-a'} = C_{b-a}$, we have $obj(a) + D_{loca(k',a),nj2(a')} + D_{nj2(a'),j} + C_{nj2(a')} = obj(a') \geq obj(b') + C_{b'-a'} = obj(b) + C_{b-a} + D_{loca(k',b),nj2(b')} + D_{nj2(b'),j} + C_{nj2(b')}$.
- Case IV: $N_J + 1 \leq n_{j2}(a') = n_{j2}(b') = s' \leq N_J + N_{STA}$, $s' \notin S(a)$ and $s' \in S(b)$. Since $loca(k',a) = loca(k',b)$ by Eqs. 6, 8, $n_{j2}(b') = n_{j2}(a')$, $C_{nj2(a')} \geq 0$ and $C_{b'-a'} = C_{b-a}$, we have $obj(a) + D_{loca(k',a),nj2(a')} + D_{nj2(a'),j} + C_{nj2(a')} = obj(a') \geq obj(b') + C_{b'-a'} = obj(b) + C_{b-a} + D_{loca(k',b),nj2(b')} + D_{nj2(b'),j}$.
- Case V: $N_J + 1 \leq n_{j2}(a') = n_{j2}(b') = s' \leq N_J + N_{STA}$, $s' \in S(a)$ and $s' \notin S(b)$. Since $loca(k',a) = loca(k',b)$ by Eqs. 6, 8, $n_{j2}(b') = n_{j2}(a')$ and $C_{b'-a'} = C_{b-a} - C_{nj2(b')}$, we have $obj(a) + D_{loca(k',a),nj2(a')} + D_{nj2(a'),j} = obj(a') \geq obj(b') + C_{b'-a'} = obj(b) + C_{b-a} + D_{loca(k',b),nj2(b')} + D_{nj2(b'),j} + C_{nj2(b')}$. □

Proposition 1 *Consider a dominated node and suppose it has a MILP-feasible descendant. Then, there is a MILP-feasible descendant of the dominating node with at least as good objective function value.*

As a consequence, we can ignore dominated nodes when proceeding to the next level of the B&B tree.

Proof Applying Lemma 1 recursively, any level-feasible descendant of a dominated node is guaranteed to be dominated by some level-feasible descendant of the dominating node (at the same level of the B&B tree). As such, at the bottom of the tree, when all jobs have been assigned, if the dominated node generated a MILP-feasible solution, the dominating node will also have generated one with at least as good objective function value. □

Definition 2 UAV A and B are said to be **identical**, if their initial location, maximum travel speed

Table 4 The example of ST_k of N_a

UAV	Sequence of tasks
1	7,1,8 2
2	7,3,4
3	8,5
4	8,6

and maximum flight time are the same. UAVs A and B are said to be in the same **UAV class**.

The UAV classes partition the set of UAVs $\{1, 2, \ldots, N_{UAV}\}$. Let $N_{class} \le N_{UAV}$ denote the number of such UAV classes. Label these classes from 1 to N_{class} and use ω_m to denote the m^{th} UAV class (it is a set of UAV indices, where all of those UAVs are identical). Consider some node N_a, let $\Theta_m(a)$ be the set of $ST_k(a)$, where $k \in \omega_m$. Consider the example $ST_k(a)$ task sequences given in Table 4. Suppose that UAVs 1 and 2 are in a UAV class and UAVs 3 and 4 are in another UAV class and there are 6 split jobs (with indices 1, 2, 3, 4, 5 and 6) and 2 stations (with indices 7 and 8). We thus have $N_{class} = 2$, $\omega_1 = \{1,2\}$, $\omega_2 = \{3,4\}$, $\Theta_1(a) = \{ (7,1,8\ 2), (7,3,4)\}$ and $\Theta_2(b) = \{(8,5), (8,6)\}$.

Proposition 2 *For nodes N_a and N_b at the same level, If $\Theta_m(a) = \Theta_m(b)$ for all m, then node N_a is dominated by node N_b.*

Proof Since $\Theta_m(a) = \Theta_m(b)$ for all m, there is a UAV k* in node N_b which is identical with k in node N_a and $ST_k(a) = ST_{k^*}(b)$. Since k and k* are identical and all attributes of k in N_a and k* in N_b are the same (since $\Theta_m(a) = \Theta_m(b)$ for all m), exchanging the index of k and k* in N_a without changing $ST_k(a)$ provides an equivalent schedule and design solution as in N_a. We thus have

$$n_{i1}(a) = n_{i1}(b) \quad \forall i \in \{1, \ldots, N_J\}$$

$$rF(k,a) = rF(k,b) \quad \forall k \in \{1, \ldots, N_{UAV}\}$$

$$obj(a) = obj(b) \quad \text{and } C_{b-a} = 0.$$

Thus, conditions (6), (7) and (8) hold. □

Since these nodes are dominated, we can ignore them by Proposition 1.

4.5 Algorithmic Methods to Reduce the Search Space

We now turn our attention to the pruning of nodes based on the comparison with an upper bound on the objective function value obtained from a MILP-feasible solution of the problem.

The $RHTA_d$ heuristic may be able to generate such a MILP-feasible solution. Let UB denote the objective function value of the MILP-feasible solution obtained from the $RHTA_d$; set $UB = \infty$, if no such feasible solution is obtained.

Proposition 3 *If a level-feasible node N_a has objective function value $obj(a) > UB$, the objective function value of any MILP-feasible final descendant of N_a, call it $N_{a''}$, will satisfy $obj(a'') > UB$.*

This follows immediately from the properties of child nodes. As such, we can prune node N_a and all of its descendants from further consideration.

Further pruning is possible based on the comparison of the $RHTA_d$ UB value with a lower bound on the objective function value of any MILP-feasible descendant of a level-feasible node N_a. This lower bound will be obtained by the consideration of the unassigned split jobs in node N_a.

Recall that Ω_J, $U_J(a)$ and Ω_S are the sets of split job indices, remaining split job indices at node N_a and station indices, respectively. For level-feasible node N_a, let

$$L(a) := obj(a) + w_d \times \max\left\{ \sum_{i \in U_J(a)} \Delta_1(i), \sum_{i \in U_J(a)} \Delta_2(i) \right\},$$

where we define the function $\Delta_1(i) := \min_{j \in \Omega_J \bigcup \Omega_{STA} \setminus \{i\}} D_{ij}$ and define the function $\Delta_2(i) := \min_{j \in \Omega_J \bigcup \Omega_{STA} \setminus \{i\}} D_{ji}$. As we will show, L(a) is a lower bound on the objective function value for any MILP-feasible descendant of N_a. Here, $\Delta_1(i)$ is the minimum distance a UAV will travel starting from the end point of split job i to to the start location of any next possible job/station. Similarly, $\Delta_2(i)$ is the minimum distance a UAV will travel starting from the end location of any possible job/station to the start location of split job i.

Proposition 4 *If a level-feasible node N_a has L(a) > UB, the objective function value of any MILP-feasible final descendant of N_a, call it $N_{a''}$, will satisfy obj(a") > UB.*

Proof The objective function value of any MILP-feasible descendent of N_a is obj(a) plus the cost of additional stations opened and UAVs used in branching to $N_{a''}$ plus the cost of travel distance required to complete the remaining split jobs with index in $U_J(a)$. Ignoring the new station and new UAV costs gives the bound obj(a") ≥ obj(a) + the cost of travel distance required to serve the remaining split jobs. This travel distance is at least as great as max $\left\{\sum_{i\in U_J(a)} \Delta_1(i),\ \sum_{i\in U_J(a)} \Delta_2(i)\right\}$ Multiplying by the unit cost of travel gives the result. □

As such, we can prune N_a and all its descendants from the B&B tree.

Note that that it is possible to obtain a tighter bound on obj($N_{a''}$). There are at least two ways. First, Δ_1(i) can be increased (thereby tightening the lower bound) by restricting to $j \in U_J(a) \bigcup \Omega_{STA} \setminus \{i\}$. Second, it is possible to determine in some cases when a new station will be opened (we drop the new station costs in the proof of Proposition 4). However, these improved bounds are only helpful in the B&B method *if they reduce the overall computation required*. Based on our studies, these tightened bounds—while allowing slightly improved pruning—in fact require more computation overall.

4.6 Bounds on the Number of Nodes

Typically, the more nodes there are in the B&B tree, the greater the computational time required to obtain an optimal solution. As such, the number of nodes is a measure of computational complexity. We next obtain a bound on the number of nodes in the B&B tree as a function of number of split jobs, UAV candidates and station candidates.

Proposition 5 *The number of level-feasible nodes (including MILP-feasible nodes at the final level of the B&B tree) is less than or equal to $\sum_{L=1}^{N_J} N_{node}^L + 1$. Here N_{node}^L is the maximum possible number of nodes in level L and is calculated as $N_{node}^L = N_{node}^{L-1} \times (N_J - L + 1) \times N_{UAV} \times (N_{STA} + 1)$; $N_{node}^0 = 1$.*

Proof In level 0 there is one node, $N_{node}^0 = 1$. For any level L, the B&B algorithm in Appendix 3 exhaustively generates no more than one node for each UAV, station (including the direct flight option) and remaining split job. Thus, there are no more than $N_{UAV} \cdot (N_{STA} + 1) \cdot (N_J - L + 1)$ child nodes generated from each node in level L. The result follows. □

In order to determine the dominant nodes of Proposition 1, the B&B algorithm compares the attributes in Eqs. 7 and 8 for nodes pairwise at a given level of the B&B tree. We next obtain a bound on the number of these attribute comparisons.

Proposition 6 *Counting each of the pairwise comparisons in Eqs. 7 and 8 as one comparison, the number of attribute comparisons conducted in our B&B algorithm when testing the dominance of Proposition 1 is less than or equal to*

$$2\cdot\sum_{L=1}^{N_J} \binom{(N_{STA}+1)^L \cdot L!}{2} \cdot \binom{N_J}{L} \cdot (N_{UAV})^L.$$

Proof Proposition 1 compares nodes satisfying Eq. 6, that is nodes N_a and N_b with $n_{i1}(a) = n_{i2}(b)$ for all i. For a level L, there are at most $\binom{N_J}{L} \times (N_{UAV})^L$ sets of nodes that satisfy this property. (So each set comprises all nodes satisfying Eq. 6 for particular fixed values of n_{i1}, i = 1, …, N_J. These sets all have the same number of nodes in them.) In each of these sets, there are no more than $(N_{STA} + 1)^L \cdot L!$ nodes. This is because, each of the L rows that have been assigned can have any of $N_{STA} + 1$ values for their station. In addition, there are L! ways to reach such a node. For example, if split jobs 1, 3 and 6 have been assigned in a node, there are 3! orders in which these assignments could have been made. That is, the split jobs could have been assigned in the order 1, 3, 6 (1 in level 1, 3 in level 2, 6 in level 3) or 1, 6, 3 or 3, 1, 6, etc. Inside each of these sets, we then must compare two attributes (Eqs. 7 and 8) for the nodes pairwise. □

Note that Proposition 6 only bounds the number of attribute comparisons and does not consider the effort required to identify the sets of nodes that satisfy Eq. 6.

Table 5 Split job information for persons 1 and 2

Split job	Start point	End point	Process time (min)	Split job start time (min)
1	596,167	432,94	3	5
2	432,94	262,74	3	8
3	262,74	142,186	3	11
4	142,186	6,266	3	14
5	458,64	565,15	3	8

5 Numerical Study

We next conduct numerical tests of our MILP, B&B and $RHTA_d$. We study various cost values and models. We increase the number of resource candidates to explore the limitations of each approach. All tests were implemented on a personal computer with Intel(R) Core(TM)2 Quad CPU Q8400, 2.66 GHz and 4.00 GB RAM. We used ILOG CPLEX 12.4 to solve the MILP in Section 3 and solve the IP inside $RHTA_d$. We implement $RHTA_d$ and B&B using NetBeans IDE 7.1.2 and JDK 1.6.

Figure 4 depicts the paths for our study. There are six moving persons on the KAIST campus. Persons 1, 2, 3, 4, 5 and 6 move from S1 to E1 (Task 1), S2 to E2 (Task 2), S3 to E3 (Task 3), S4 to E4 (Task 4), S5 to E5 (Task 5) and S6 to E6 (Task 6), respectively. We want to select resources (design) and schedule the system. The persons walk at a constant 4 km/hour. Person 1, 2, 3, 4, 5 and 6 start to travel at time 5, 8, 17, 20, 9 and 30 min and finish their travel at time 17, 11, 20, 24, 14 and 33 min, respectively. They should be observed during their travel by a UAV. We assume that each candidate service station initially has two candidate UAVs. The time to refresh a UAV at a station is 1 min. Each UAV has a maximum travel time of 8 min and maximum speed of 160 m/min. First, we consider persons 1 and 2. We create 5 split jobs from the tasks for person 1 and 2. Split jobs 1–4 and 5 are for persons 1 and 2, respectively. Each is 3 min in duration; see Table 5.

We initially set W_d = \$10/m, C_k = \$400 (UAV cost) and C_s = \$100 (station cost). Later, we will consider a range of values for C_s. Three station candidates are located at coordinates (400 m, 150 m), (200 m, 50 m), (0 m, 200 m). Each has two UAV candidates. These candidate resources can be purchased for use or not. The MILP provides

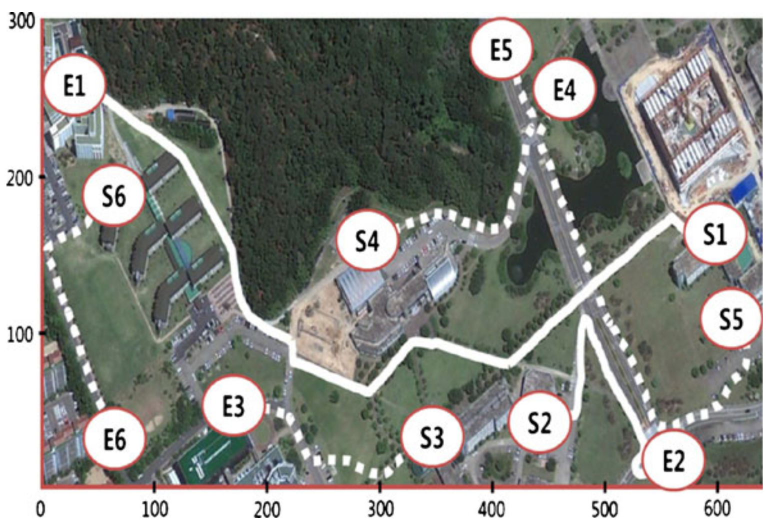

Fig. 4 Six moving objects in a field of operations

Table 6 Optimal resources and schedule (Cs = $100)

UAV	Split job served	Start station	End station
1	1,2	1	2
2	5	1	1
3	None	N/A	N/A
4	3,4	2	3
5	None	N/A	N/A
6	None	N/A	N/A
Selected UAVs = 1,2,4 / Selected stations = 1,2,3			
Travel distance = 691 m			
Obj. value = 8,410			

the following optimal solution with cost $8410. UAV 1 departs from station 1 to serve split jobs 1 and 2; the UAV then returns to station 2. UAV 2 departs from station 1 to serve split job 5. It then returns to station 1. Split jobs 3 and 4 are served by UAV 4; this UAV starts and ends at station 2 and station 3, respectively. UAVs 3, 5 and 6 are not used (we do not purchase them). See Table 6.

We consider the performance with an increased number of resource candidates. We use $C_k = \$400$ for all k, $W_d = \$10/m$ and $C_s = \$100$ for all s.

Note that a B&B which allows more resource candidates (in the given locations) automatically considers all cases with fewer resources. It can simply decide to ignore the extra resources. As such, a B&B that allows more candidate resources (at given locations) will have at least as good an objective function value. As such, it may provide better solutions to the overall design and scheduling problem. Refer to Table 7.

We consider numerous cases by varying the number of customers, the number of stations and C_s values. Table 8 shows the input parameters for our numerical study. We compare the MILP, B&B and $RHTA_d$. We fix the UAV cost at $C_k = \$400$ for all k and set $W_d = \$10$/meter and p = 5.

Varying parameters, we consider 120 test cases. The B&B method and MILP via CPLEX obtain optimal solutions in 56 and 34 cases, respectively. Otherwise, they return an "out of memory" error. The B&B method solves all cases solved by the MILP via CPLEX. The $RHTA_d$ obtains feasible solutions in all 120 test cases studied. For cases solved by the MILP via CPLEX, the average computational times and objective function value from the $RHTA_d$, B&B method and the MILP for each C_s value are provided in Table 9.

The B&B method provides much faster computational time compared to the MILP via CPLEX and also guarantees an optimal solution. The B&B method is 767, 999 and 221 times faster than the MILP via CPLEX for $C_s = \$100$, \$400 and \$2000, respectively. The $RHTA_d$ a is further 61, 46 and 23 times faster than the B&B method for these cases, respectively.

The average percent gap between the objective function value provided by the $RHTA_d$ and the optimal solution value for $C_s = \$100$, \$400 and \$2000 is 3.88 %, 3.42 % and 10.18 %, respectively. The detailed results for cases that B&B was able to solve are provided in Appendix 5.

Table 7 Sensitivity of the methods to the number of resource candidates ($W_d = 10$, $C_s = \$100$, $C_k = \$400$)

# of station	# of UAV	Obj. value
3	6	8,410
6	12	4,450
9	18	4,450
12	24	4,450
15	30	4,450
18	36	4,430
21	42	4,270
24	48	4,260

Table 8 The input parameters for our study

# of customers	# of stations	Cs
2	3	$100
3	6	$400
4	9	$2000
5	12	
6	15	
	18	
	21	
	24	

Table 9 Average computational time of the methods

Cs	Objective function value			Computational time		
	$RHTA_d$	B&B	MILP	$RHTA_d$	B&B	MILP
100	7535.3	7242.7	7242.7	0.11	6.68	2158.22
400	8930	8624.6	8624.6	0.07	3.20	3195.50
2000	15585	13998.3	13998.3	0.09	2.07	458.11

6 Concluding Remarks

A system consisting of a fleet of UAVs and automated service stations can enable the persistent pursuit of multiple missions across a field of operations. The number/location of stations, the number of UAVs and their schedules will determine system performance and cost.

Here, we develop a mixed integer linear program (MILP) for concerted resource selection (system design) and scheduling. The MILP seeks to minimize the total system cost (travel and resources) while ensuring that every mission is provided at least one UAV at all times. The missions are time-space trajectories that must be observed. UAVs may return to the field after replenishing themselves at geographically distributed service stations. Multiple customer missions can occur simultaneously. Each mission can be prosecuted by several UAVs as required. UAVs are able to conduct multiple tasks between visits to the stations. The MILP uses the concept of a split job to discretize the mission paths.

The concerted resource selection and scheduling approach appears to be the first of its kind in the persistent UAV literature. It promises to provide significant improvement in overall cost when compared to solving either problem independently. However, the computation required for concerted optimization is greater than either problem alone.

To address the complexity inherent in the MILP formulation, we developed a modified receding horizon task assignment (RHTA) heuristic for our problem called $\mathrm{RHTA_d}$. It iteratively breaks the problem into smaller optimizations. We developed a branch and bound method (B&B) that reduces the computational time and guarantees an optimal solution (given sufficient computational capacity). Numerous structural and algorithmic properties are developed to prune inferior nodes.

We conducted a numerical study to compare the performance of the MILP, B&B method and $\mathrm{RHTA_d}$. The B&B method is about 500 times faster than the MILP via CPLEX (and provides an optimal solution). The $\mathrm{RHTA_d}$ is about a further 50 times faster than the B&B method, however, it may sacrifice optimality.

It may be wise to consider capacitated service stations that can serve a limited number of UAVs at one time. Our deterministic mission assumptions should be relaxed to allow stochastic models.

Appendix

A1 MILP Model

$$\text{Min } W_d \sum_{k\in K}\sum_{r\in R}\sum_{i\in\Omega_A}\sum_{j\in\Omega_A} D_{ij}\cdot X_{ijkr} + \sum_{k\in K} C_k \cdot U_k^{UAV} + \sum_{s\in\Omega_{SS}} C_s\cdot U_s^{STA}, \tag{9}$$

subject to

$$\sum_{j\in\Omega_J\cup\Omega_{SE}} X_{s_{ok},jk1} = 1 \qquad (k\in K), \tag{10}$$

$$\sum_{s\in\Omega_{SS}}\sum_{j\in\Omega_J\cup\Omega_{SE}} X_{sjkr} = 1 \qquad (k\in K,\ r\in R), \tag{11}$$

$$\sum_{s\in\Omega_{SE}}\sum_{i\in\Omega_J\cup\Omega_{SS}} X_{iskr} = 1 \qquad (k\in K,\ r\in R), \tag{12}$$

$$\sum_{i\in\Omega_J\cup\Omega_{SS}} X_{iskr} = \sum_{i\in\Omega_J\cup\Omega_{SE}} X_{s-1,ikr+1} (k\in K,\ r=1\ldots N_R-1, s\in\Omega_{SE}), \tag{13}$$

$$C_{skr} = C_{s-1,kr+1} \qquad (k\in K,\ r=1\ldots N_R-1, s\in\Omega_{SE}), \tag{14}$$

$$\sum_{k\in K}\sum_{r\in R}\sum_{i\in\Omega_A} X_{ijkr} = 1 \qquad (j\in\Omega_J), \tag{15}$$

$$\sum_{j\in\Omega_A} X_{ijkr} - \sum_{j\in\Omega_A} X_{jikr} = 0 \qquad (i\in\Omega_J, k\in K, r\in R), \tag{16}$$

$$\sum_{i\in\Omega_J\cup\Omega_{SS}} X_{iskr} = 0 \quad (k\in K,\ r\in R, s\in\Omega_{SS}), \tag{17}$$

$$C_{ikr} + P_i + D_{ij}/TS_k - C_{jkr} \le M(1 - X_{ijkr}) \quad (i\in\Omega_J\cup\Omega_{SS},\ j\in\Omega_J\cup\Omega_{SE}, k\in K,\ r\in R), \tag{18}$$

$$M \cdot \sum_{j \in \Omega_J \cup \Omega_{SE}} X_{ijkr} \geq C_{ikr} (i \in \Omega_J \cup \Omega_{SS}, k \in K, \ r \in R), \tag{19}$$

$$\sum_{k \in K} \sum_{r \in R} C_{ikr} = E_i \qquad (i \in \Omega_J), \tag{20}$$

$$\sum_{i \in \Omega_A} \sum_{j \in \Omega_A} D_{ij} / TS_k \cdot X_{ijkr} + \sum_{i \in \Omega_J} \sum_{j \in \Omega_A} P_i \cdot X_{ijkr} \leq q_k (k \in K, \ r \in R), \tag{21}$$

$$\left(1 - U_k^{UAV}\right) \leq X_{s_{ok}, s_{ok}+1, kr} \qquad (k \in K, \ r \in R), \tag{22}$$

$$U_{s_{ok}}^{STA} \geq U_k^{UAV} \qquad (k \in K), \tag{23}$$

$$U_s^{STA} \geq X_{iskr} \qquad (s \in \Omega_{SE}, i \in \Omega_J, k \in K, r \in R), \tag{24}$$

$$C_{ikr} \geq 0 \qquad (k \in K, \ r \in R, i \in \Omega_A), \tag{25}$$

$$U_k^{UAV}, \ U_s^{STA}, \ X_{ijkr} \in \{0, 1\} \ (k \in K, \ r \in R, i \in \Omega_A, \ j \in \Omega_A), \tag{26}$$

A2 Detailed Pseudo Code of RHTA$_d$

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, usage of UAV, usage of station, cost of UAV, cost of station, 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 ($D(cL(k), w_1) + \sum_{i=1}^{n_c-1} D(w_i, w_{i+1}) + D(W_{nc}, s) + \sum_{i=1}^{n_c} P(w_i) \leq rF(k)$ and $at(k) + D(cL(k), w_1) / TS(k) \leq E(w_1)$ and $E(w_{i-1}) + P(w_{i-1}) + D(w_{i-1}, w_i) / TS(k) \leq E(w_i)$ for i=1,…,nc) **do**
S_i ¡ç $D(cL(k), w_1) + \sum_{i=1}^{n_c} D(w_{i-1}, w_i)$;
P_i ¡ç $[w_1, ..., w_{n_c}]$;
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}
uUAV(k)←true; C_k←0; \\ {update UAV usage status}
uSTA(iSTA(k))←true; $C_{iSTA(k)}$←0; \\ {update station usage status}
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 $((D(cL(k),w)/TS(k)+P(w)+D(w,s)/TS \geq rF(k)$ or $D(cL(k),w)/TS(k)+at(k) \geq E(w))$ and $cL(k) \neq s$) **do**
$cost_{kws} \leftarrow C_s$ + D(cl(k),s);
$s_{min} \leftarrow argmin_s\ cost_{kws}$; \\ {Choose the best station}
$M_k \leftarrow [M_k, s_{min}]$; rF(k)←maxF(k); at(k)←at(k) + D(cl(k),s) + p(s); cL(k)←s_{min}; uSTA(s_{min})←true; $C_{iSTA(k)} \leftarrow 0$;
for all UAV k **do** // {send UAV at end of job to station}
for all station s **do**
if !(cL(k)==s) **do**
$cost_{ks} \leftarrow C_s$ + D(cl(k),s); $s_{min} \leftarrow argmin_s\ cost_{ks}$; \\ {Choose the best station}; $M_k \leftarrow [M_k\ s_{min}]$; uSTA($s_{min}$)←true;

A3 Detailed Pseudo Code of B&B

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, usage of UAV, usage of station, cost of UAV, cost of station, initial location of UAV, available time of UAV, remaining fuel time, upper bound of object value)
Calculate add_i (minimum required distance of each split job)
set N_0 with initial system state
While level of branch and bound is equal to N_J **do**
While all nodes N_P in previous level **do**
for all split job i in parent node N_P **do**
if split job i is not assigned **do**
for all UAV k **do**
if E[i] < t_{avail}[k] **continue;**
for all station s, direct flight **do**
if UAV satisfy strict remaining fuel constraints and time constraint **do**
Generate candidate node N_C
Assign UAV k to split job i and set attributes of N_C
Calculate low bound of N_C
if L(c) > UB continue;
for all nodes N_G in current level **do**
if N_C is dominated by or identical with N_G **break;**
if N_G is dominated by N_C **do** remove N_G and insert N_C in current set of nodes
else insert N_C in current set of nodes
While all nodes N in level N_J **do**
count the number of used UAV candidates (Set count)
Set input variables (oSTA: set of already open station candidates; noSTA: set of not open station candidates)
if the number of oSTA == N_{STA}
send UAVs at end of job to closest station
else
for all number of m of the number of used UAV candidates m = 1,…,count **do**
generate aSTA: set of additional open station (pick m station among noSTA)
for all aSTA **do**
set pSTA ← oSTA + aSTA;
additional station cost ← sum of cost of aSTA
additional distance cost ← minimum travel distance from end of job to station of pSTA for all UAVs
Find minimum additional station cost + additional distance cost
update node state
Find minimum object value among all nodes
Output the UAV schedule and objective value

A4 Attribute Calculations for Child Nodes

Remaining fuel :

$$rF\left(k',a'\right)=\begin{cases} rF\left(k',a\right)-t_{tra}\left(k',loca\left(k',a\right),j\right)-P\left(j\right), & if\ s'=N_J+N_{STA}+1,\\ q\left(k'\right)-t_{tra}\left(k's',j\right)-P\left(j\right), & if\ N_J+1\le s'\le N_J+N_{STA}.\end{cases}$$

UAV′s available time :

$$t_{\text{avail}}\left(k',a'\right)=E\left(j\right)+P\left(j\right)$$

Sequence of tasks :

$$ST\left(k',a'\right)=\begin{cases} \left\{ST\left(k',a\right),j\right\}, & if\ \ s'=N_J+N_{STA}+1,\\ \left\{ST\left(k',a\right),s',j\right\}, & if\ N_J+1\le s'\le N_J+N_{STA}.\end{cases}$$

Location of UAV :

$$loca\left(k',a'\right)=j$$

Objective value :

$$obj\left(a'\right)=\begin{cases} obj\left(a\right)+D_{loca(k',a),j}+cUAV(a'), & if\ s'=N_J+N_{STA}+1\\ obj\left(a\right)+D_{loca(k',a),s'}+D_{s',j}+cUAV\left(a'\right)+cSTA\left(a'\right), & if\ N_J+1\le s'\le N_J+N_{STA}.\end{cases}$$

A5 Experiment Result

C_s	# of customer	# of station	# of UAV	$RHTA_d$		B&B		MILP*	
				Comp. time	Obj. value	Comp. time	Obj. value	Comp. time	Obj. value
100	2	3	6	0.343	8410	0.047	8410	4.49	8410
	2	6	12	0.047	4450	0.015	4450	14.58	4450
	2	9	18	0.063	4450	0.125	4450	41.87	4450
	2	12	24	0.063	4450	0.514	4450	274.14	4450
	2	15	30	0.094	4450	1.373	4450	293.21	4450
	2	18	36	0.109	4430	5.008	4430	2324.69	4430
	2	21	42	0.125	4270	19.282	4270	2440.01	4270
	2	24	48	0.109	4260	62.01	4260	N/A	N/A
	3	3	6	0.031	10440	0.031	10040	17.23	10040
	3	9	18	0.062	6020	1.123	5620	137.95	5620
	3	12	24	0.078	6020	5.476	5620	575.12	5620
	3	15	30	0.109	6020	19.063	5620	N/A	N/A
	3	18	36	0.125	5610	61.73	5210	N/A	N/A
	3	21	42	0.141	5450	306.275	5050	N/A	N/A
	3	24	48	0.109	5440	1260.361	5040	N/A	N/A

	4	3	6	0.046	13120	0.219	12740	33.25	12740
	4	6	12	0.125	9090	5.304	8130	110.1	8130
	4	9	18	0.062	9050	59.078	8130	3983	8130
	4	12	24	0.109	8300	861.215	7900	N/A	N/A
	5	3	6	0.437	18360	2.371	17830	22084	17830
	5	6	12	0.078	11390	55.505	10440	N/A	N/A
	6	3	6	0.125	20330	16.645	18340	N/A	N/A
Average for cases solved by MILP				**0.114**	**7535.333**	**6.677**	**7242.667**	**2158.219**	**7242.667**
400	2	3	6	0.124	9310	0.032	9310	4.24	9310
	2	6	12	0.046	5650	0.047	5650	17.34	5650
	2	9	18	0.063	5650	0.218	5650	172.78	5650
	2	15	30	0.094	5650	2.309	5650	2837.91	5650
	2	18	36	0.125	5630	8.018	5630	5284.76	5630
	2	21	42	0.141	5470	27.159	5470	N/A	N/A
	2	24	48	0.109	5470	83.476	5470	N/A	N/A
	3	3	6	0.015	11340	0.032	10940	21.41	10940
	3	6	12	0.046	7520	0.344	7120	47.61	7120
	3	9	18	0.063	7520	2.277	7120	820.75	7120
	3	12	24	0.078	7520	11.466	7120	4450.31	7120
	3	15	30	0.094	7520	41.449	7120 7	N/A	N/A
	3	18	36	0.125	7110	130.728	6710	N/A	N/A
	3	21	42	0.156	6950	597.029	6550	N/A	N/A
	3	24	48	0.14	6950	2342.391	6550	N/A	N/A
	4	3	6	0.016	14020	0.234	13620	68.44	13620
	4	6	12	0.078	11370	13.494	9930	1200.76	9930
	5	3	6	0.047	19260	2.231	18730	25631.36	18730
	6	3	6	0.047	21230	16.677	19240	N/A	N/A
Average for cases solved by MILP				**0.067**	**8930.000**	**3.196**	**8624.615**	**3195.502**	**8624.615**
2000	2	3	6	0.219	14110	0.078	14110	15.27	14110
	2	6	12	0.109	12690	0.281	11230	108.4	11230
	2	9	18	0.093	12690	1.872	11230	1760.54	11230
	2	12	24	0.109	12110	6.802	11010	N/A	N/A
	2	15	30	0.125	12110	25.771	10450	N/A	N/A
	2	18	36	0.156	12640	170.243	10450	N/A	N/A
	2	21	42	0.172	12830	819.313	10450	N/A	N/A
	2	24	48	0.156	12830	2579.028	10450	N/A	N/A
	3	3	6	0.031	16140	0.047	15740	26.53	15740
	3	6	12	0.063	19060	9.828	13260	704.49	13260
	4	3	6	0.031	18820	0.328	18420	133.45	18420
	5	3	6	0.156	24060	2.761	22180	N/A	N/A
	6	3	6	0.141	26030	18.923	24040	N/A	N/A
Average for cases solved by MILP				**0.091**	**15585.000**	**2.072**	**13998.333**	**458.113**	**13998.333**

* N/A indicates that CPLEX did not return a solution

References

1. Kemper Filho, P., Suzuki, K.A.O., Morrison, J.R.: UAV consumable replenishment: design concepts for automated service stations. J. Intell. Robot. Syst. **61**(1), 369–397 (2011)
2. Suzuki, K.A.O., Kemper Filho, P., Morrison, J.R.: Automatic battery replacement system for UAVs: analysis and design. J. Intell. Robot. Syst. **65**(1), 563–586 (2012)
3. Valenti, M., Dale, D., How, J., Vian, J.: Mission health management for 24/7 persistent surveillance operations. In: AIAA Guidance, Control and Navigation Conference, Myrtle Beach, SC (2007)
4. Godzdanker, R., Rutherford, M.J., Valavanis, K.P.: ISLANDS: a self-leveling landing platform for autonomous miniature UAVs. In: 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2011), pp. 170–175. Budapest, Hungary, 3–7 July 2011
5. Godzdanker, R., Rutherford, M.J., Valavanis, K.P.: Improving endurance of autonomous aerial vehicles through intelligent service-station placement. In: 2012 IEEE International Conference on Robotics and Automation, RiverCentre, pp. 3179–3184. Saint Paul, Minnesota, USA, 14–18 May 2012
6. Ji, M., Xia, J.: Analysis of vehicle requirements in a general automated guided vehicle system based transportation system. Comput. Ind. Eng. **59**(12), 544–551 (2010)
7. Shima, T., Schumacher, C.: 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
8. Zeng, J., Yang, X., Yang, L., Shen, G.: Modeling for UAV resource scheduling under mission synchronization. J. Syst. Eng. Electron. **21**(13), 821–826 (2010)
9. Weinstein, A.L., Schumacher, C.: 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
10. Kim, Y.S., Gu, D.W., Postlethwaite, I.: Real-time optimal mission scheduling and flight path selection. IEEE Trans. Autom. Control **52**(14), 1119–1123 (2007)
11. Alidaee, B., Wang, H., Landram, F.: A note on integer programming formulations of the real-time optimal scheduling and flight selection of UAVS. IEEE Trans. Control Syst. Technol. **17**(12), 839–843 (2009)
12. Shetty, V.K., Sudit, M., Nagi, R.: Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles. Comput. Oper. Res. **35**, 1813–1828 (2008)
13. Alidaee, B., Wang, H., Landram, F.: On the flexible demand assignment problems: case of unmanned aerial vehicles. IEEE Trans. Automation Sci. Eng. **8**(12), 865–868 (2011)
14. Bethke, B., Valenti, M., How, J.: UAV task assignment. Robot. Autom. Mag. **15**(6), 39–44 (2008)
15. Nigam, N., Bieniawski, S., Kroo, I., Vian, J.: Control of multiple UAVs for persistent surveillance: algorithm and flight test results. IEEE Trans. Control Syst. Technol. **20**(13), 1236–1251 (2012)
16. Kim, J., Song, B., Morrison, J.R.: On the scheduling of systems of heterogeneous UAVs and fuel service stations for long-term mission fulfillment. J. Intell. Robot. Syst. **70**(1), 347–359 (2013)
17. Song, B., Kim, J., Kim, J., Park, H., Morrison, J.R.: Persistent UAV service: an improved scheduling formulation and prototypes of system components. In: Proc. 2013 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 915–925. Atlanta, GA, 28–31 May 2013
18. Alighanbari, M.: Task assignment algorithms for team of UAVs in dynamic environments. Master Thesis, Massachusetts Institute of Technology (2004)