J Intell Robot Syst (2016) 84:241–258
DOI 10.1007/s10846-015-0280-5

# Rolling Horizon Path Planning of an Autonomous System of UAVs for Persistent Cooperative Service: MILP Formulation and Efficient Heuristics

**Byung Duk Song · Jonghoe Kim · James R. Morrison**

Received: 15 December 2014 / Accepted: 23 September 2015 / Published online: 21 October 2015


**Abstract** A networked system consisting of unmanned aerial vehicles (UAVs), automated logistic service stations (LSSs), customer interface software, system orchestration algorithms and UAV control software can be exploited to provide persistent service to its customers. With efficient algorithms for UAV task planning, the UAVs can autonomously serve the customers in real time. Nearly uninterrupted customer service may be accomplished via the cooperative hand-off of customer tasks from weary UAVs to ones that have recently been replenished at an LSS. With the goal of enabling the autonomy of the task planning tasks, we develop a mixed integer linear programming (MILP) formulation for the problem of providing simultaneous UAV escort service to multiple customers across a field of operations with multiple sharable LSSs. This MILP model provides a formal representation of our problem and enables use in a rolling horizon planner via allowance of arbitrary UAV initial locations and consumable reservoir status (e.g., battery level). As such, it enables automation of the orchestration of system activities. To address computational complexity, we develop efficient heuristics to rapidly derive near optimal solutions. A receding horizon task assignment (RHTA) heuristic and sequential task assignment heuristic (STAH) are developed. STAH exploits properties observed in optimal solutions obtained for small problems via CPLEX. Numerical studies suggest that RHTA and STAH are 45 and 2100 times faster than solving the MILP via CPLEX, respectively. Both heuristics perform well relative to the optimal solution obtained via CPLEX. An example demonstrating the use of the approach for rolling horizon planning is provided.

**Keywords** Persistent UAV service · Cooperative UAV service · UAV task planning · Mixed integer linear programming · Heuristic

## 1 Introduction

The integration of unmanned aerial vehicles (UAVs), automated logistics service stations (LSSs), communication networks, control algorithms, system orchestration algorithms and software can enable the UAVs to provide a persistent service to multiple customers. The UAVs provide service to the customers in turn, handing the tasks to replacement UAVs in a cooperative fashion as required. LSSs situated across the field

B. D. Song · J. Kim · J. R. Morrison (✉)
Department of Industrial and Systems Engineering, KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea

James R. Morrison
e-mail: james.morrison@kaist.edu;
http://xS3D.kaist.edu

of operations replenish consumables, such as a UAV's battery, and enable the persistent operations. The communication network distributes the system state information and control/planning commands across the field to the networked resources. Ensuring that essential resources are provided to the UAVs before they are required while maximizing the system objectives are tasks conducted by the orchestration algorithms. The control algorithms direct the UAV actions in real time. Taken together, such an automated system – in which UAVs cooperate to provide nearly persistent service to its customers – may find application roles in surveillance, border patrol, target tracking, disaster search and rescue, crop dusting and aerial photography.

Here we develop a mixed integer linear programming (MILP) formulation and heuristics to enable the autonomous orchestration of the path planning tasks for such a system. We focus on a system of UAVs and LSSs distributed across a field of operations and connected via networked communication systems. The objective is to simultaneously provide nearly continuous escort or surveillance to customers while minimizing total UAV travel distance (as a surrogate for energy expenditures). We assume that the customers largely traverse known time-space trajectories and deviations are addressed via a flexible MILP together with a rolling horizon approach.

### 1.1 Relevant Literature

Here we review relevant work on automated LSSs, networked cooperative systems and orchestration methods.

Automated LSSs for UAVs have been studied, developed or used for demonstrations in [1–9]. In [8], the location of stations on an n by n grid is determined by solving the p-median problem. In [4] and [9], automatic battery replacement systems were developed and tested to support nearly uninterrupted UAV flights.

Networked cooperative systems also have been reported in [10–13]. In [10], the development of a novel real time operation environment for networked control systems was discussed. In [11], an intelligent cooperative control architecture was developed; it included a cooperative planner and learning algorithm to control UAVs with fuel limitations and stochastic risks. The authors in [12] developed a system that focused on the interaction between unmanned ground vehicles (UGVs) and UAVs to extend the endurance of the UAVs. The UGVs act as docking stations and host the UAVs during the mission. The synchronization and coordination is managed by a ground control station (GCS) based on C code and the robot operating system (ROS). In [13], localizing a stationary target in a GPS denied environment was addressed using a team of UAVs equipped with bearing-only sensors. The UAVs used cooperative localization to localize themselves and the target.

We now turn our attention to efforts related to system orchestration (task planning). In [14], the approaches for such problems are categorized into Markov decision processes (MDP), integer programming (we will call it MILP for convenience) and game theory. For MDP and MILP approaches, there have been some efforts to incorporate LSSs into the planning process.

MDP approaches have been studied in [1, 4–6, 11]. Centralized real time algorithms to enable persistent operation in a stochastic environment, directing UAVs to conduct multiple flights in the planning horizon and including visits to LSSs, were conducted. Uncertainties such as UAV health and fuel levels were addressed. They combined approximate dynamic programming and reinforcement learning to address challenges associated with MDP formulations. Indoor demonstration of persistent operations was discussed with LSSs positioned at a single location.

MILP approaches with fuel awareness have been conducted in [2, 3, 17, 18, 20–25]. In [17–22], fuel limitations are incorporated, but persistence is not pursued (the UAVs do not return to the field after depletion). In our efforts detailed in the sequel, our decision variables are inspired by the classical MILP formulation for the VRP as used in [21].

MILP based research incorporating logistics tasks for persistent operations has been pursued since about 2007. In [2], a receding horizon task assignment (RHTA) approach was developed to enable persistence with a single site for the LSSs. In [3], policies and reduced MILP models were used to manage the health of UAVs.

MILP based methods for persistent operations of a system of UAVs with LSSs distributed across a field of operations were conducted in [23–25]. Their task was to provide uninterrupted 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 [24]. The UAVs used vision feedback for target tracking and landing [26]. Recently, in [27], a single UAV routing problem where there are multiple depots and the vehicle is allowed to refuel at any depot was studied. Each target is visited at least once by the UAV and total fuel consumption is minimized via objective function.

The authors in [25] pursued a similar problem to that of [23, 24]. However, in addition to determining which UAV should accomplish which task, 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) were developed and studied.

These MILP formulations are not suitable for real time application. They do not allow for arbitrary UAV location and fuel levels. The exceptions are [2, 3], which do not allow for multiple sharable LSSs distributed across the field.

The MDP approach benefits from an explicit consideration of random events. However, it suffers from the curse of dimensionality together with strong assumptions on the random variables. As such, with approximations seeking to overcome the inherent computational intractability, it can be useful for systems with significant randomness. The MDP approach also benefits from easy applicability in the real time contex; the state based decisions it provides can be used directly for task planning.

The MILP approaches benefit from a formulation that directly optimizes the objective under consideration and provides an optimal solution under two conditions: the MILP can be solved and the deterministic assumptions employed are not violated. The formal MILP formulations typically assume some initial location for the resources. The MILP can be used in a rolling horizon approach for real time task planning if the formulation allows for arbitrary resource location and consumables (e.g., fuel). Heuristics such as the RHTA can be used, at the expense of optimality for the single stage optimization, to allow for real time decision making and computational tractability.

### 1.2 Contribution

In this study, we consider a system of UAVs and LSSs geographically distributed across a field of operations. Heterogeneous UAVs are considered; the UAVs have different maximum travel speeds and fuel capacity. During service, a drained UAV may hand off its mission to a replacement UAV and travel to the LSS for replenishment. After replenishment, UAVs can return to serve customers. The goal is to assign UAV task which provide nearly uninterrupted service to customers while minimizing the total UAV travel distance. Customers provide their time-space trajectory when they request service. It may change if the customer desires, but they will inform the system of their new trajectory.

Our customer paths are assumed largely deterministic (they can deviate in the rolling horizon approach) with randomness coming from the arrival of new customers. As such, we consider the MILP formulation pursued here as more appropriate than an MDP approach.

The persistent UAV systems studied in [1–4, 15, 16] considered a single location for their LSSs. They developed real time centralized task planning methods. Here, we also focus on centralized task planning for persistent UAV service. However, we consider LSSs geographically spread across a field of operations– our focus is on pushing toward large scale persistent operations.

We will formulate our problem as a MILP, but structure it such that it can directly be used for rolling horizon real time task planning. The exact solutions to this formulation provide insight into the solution structure that we exploit to develop a heuristic that is about 2,100 times faster than solving the MILP via CPLEX.

Our efforts extend the work of [23–25] to allow for use in real time path planning. This is done by allowing for arbitrary initial fuel levels and location for each of the UAVs. Further, rather than requiring that all jobs be served (as in [23–25]), which may not be feasible in general, we allow for jobs to be dropped where required.

Toward real time path planning, the contributions of this paper are as follows. We

- Describe a UAV system to cooperatively provide persistent automated security escort service and detail how a MILP formulation may be used in a rolling horizon approach for real time path planning (Section 2).

- Develop a MILP formulation for the problem allowing for arbitrary UAV initial locations and fuel levels and enabling the rolling horizon approach (Section 3).
- Develop computationally efficient heuristics capable of solving the problem in near real time. Inspired by optimal solutions obtained via CPLEX for small problem instances, we develop the sequential task assignment heuristic (STAH) (Section 4.3). We extend RHTA approaches to our context (Section 4.4).
- Conduct numerical experiments (Section 5). RHTA and STAH are about 45 and 2,100 times faster than the MILP solved via CPLEX. The real time rolling horizon approach is demonstrated via example.

It is important to note that we do not include flight dynamics in our models. This decision is intentional, well motived by the literature and necessary for computational tractability. In many industries, a hierarchical decomposition of decisions is used for computational and modeling tractability. For systems of UAVs, the task planning tier seldom includes detailed information about flight dynamics. In [16–25] and [27, 28], task planning is conducted in the absence of flight dynamics. Detailed information about exact flight paths and flight dynamics are often considered at lower levels of detail when controlling the UAVs to accomplish their assigned tasks. As is well supported by the literature, we relegate detailed flight dynamics information as outside of our scope. As we shall see, task planning is already highly computationally intractable.

Some parts of the paper, including ideas and models detailed here, appeared in the conference paper [28]. Section 3, the allowance for not all split jobs to be served in the MILP, modified RHTA and STAH heuristics and numerical studies are newly developed in this full version of the paper.

## 2 Persistent Cooperative UAV Service

### 2.1 System Architecture

Systems of unmanned aerial vehicles (UAVs) hold much application promise due to their high capabilities. However, their capabilities are restricted by a fundamental dependence on a consumable energy

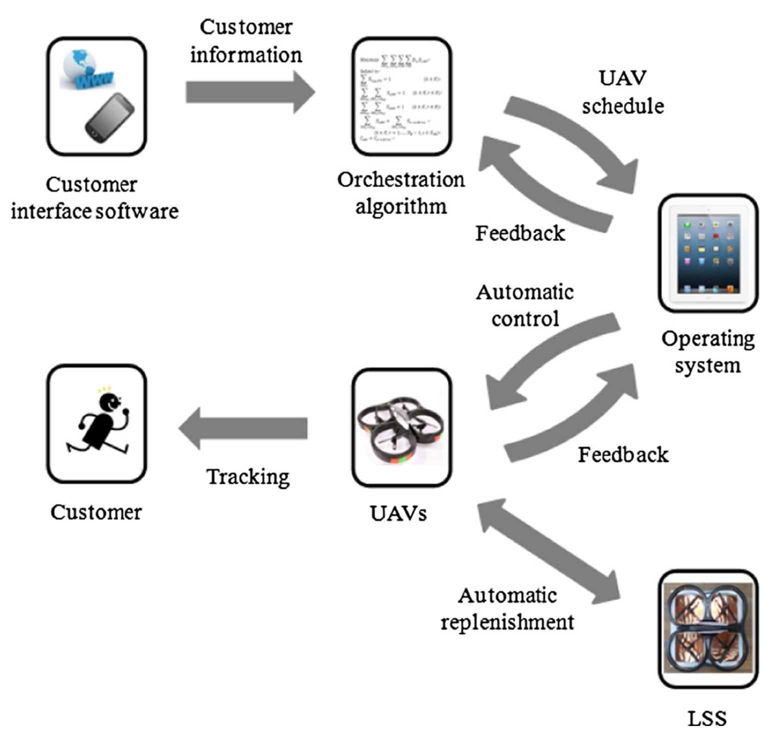



**Fig. 1** Persistent automated UAV service

source. To remove these limitations and enable continuous UAV missions, one may use an automated system consisting of the convergence of UAV control systems, automatic recharge platforms, network communication methodologies and optimization techniques for orchestrating the component activities. Figure 1 illustrates such an autonomous system. The system supervisor receives information about the deterministic customer paths (GPS coordinates) and times from the service requester. The supervisor coordinates the UAV task plans. Controllers next receive the schedule via the communication network and execute the commands at their designated times. Each UAV relies on the communication network to send vision and geolocation information to their controllers. Drained UAVs hand off their mission to fully charged UAVs and travel to the automatic LSSs for fuel recharge services.

To achieve real time path planning, one can directly solve the MILP or use STAH or RHTA heuristics developed in the sequel in a rolling horizon approach. The optimization routine is called either when specific events occur or at fixed time intervals.

If any component of the communication network or controllers fails, the UAVs are assumed to execute their original plans until they must return to a station.

### 2.2 Event Based Rolling Horizon Operation

During the operation of a system of UAVs, many events may occur: 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 accounting for these disturbances. Persistent UAV service should be pursued. In the event based rolling horizon approach, new coordinated UAV task plans are derived and dispatched across the communication network every time there is a new event. During this process, more than one event may occur. These will be stacked in an event queue. Multiple events that occur during the processing of the previous event are merged into a single new system state and treated as an event. Figure 2 describes the procedure of event based rolling horizon operation. There, $e$ indicates the time at which a particular event is processed. The events in the event queue represent unanticipated system state changes and will all be grouped into the next event. Red arrows indicate the flow of information across the communication network.

### 2.3 Time Based Rolling Horizon Operation

The time based approach is similar. At given intervals of time, if there is a deviation from the previously

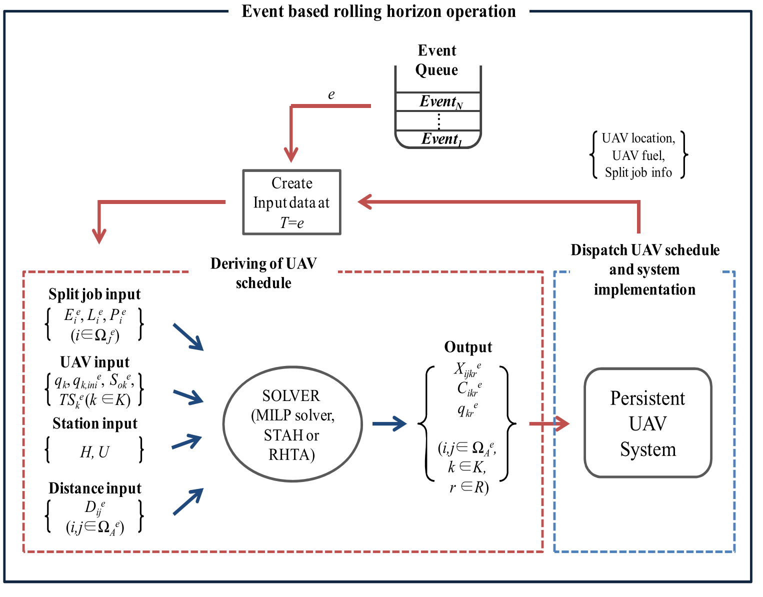



**Fig. 2** Event based rolling horizon operation

determined plan, new UAV path plans are determined using the MILP, STAH or RHTA. Let $l$ and $t$ be the index of the increment (e.g., the 347$^{\text{th}}$ time increment) of the planning process and the duration of the increment, respectively. Also let $p(< t)$ be the planning time allocated/required. At every time instant $T = t \cdot l$, deviation from the existing plans is checked. If there are deviations, a new plan is derived by time $T = t \cdot l + p$ and new UAV task plans are automatically dispatched across the communication network. Figure 3 depicts the procedure of time based rolling horizon operation.

## 3 Mathematical Formulation

We now develop a MILP model for the coordinated task planning problem that allows for arbitrary initial UAV location and battery or fuel level. It extends the models of [23–25]. The arbitrary initial conditions allow it to be used together with a rolling horizon approach.

We assume that all UAV locations, UAV fuel levels and customer time-space trajectories are known. (Changes are accounted for in the rolling horizon method described in Section 2.) Customer trajectories are discretized by dividing them into segments called split jobs that may each be served by a different UAV. The conversion from trajectory to split jobs is up to the system designer – we typically use a fixed interval of time for splitting the trajectories such as 15 seconds or one minute. There is a tradeoff between optimality and computation. More split jobs will allow for greater pursuit of the system objectives at the cost of more computation.

Each split job $i$ is defined by a start location $(x_{is}, y_{is})$ and end location $(x_{ie}, y_{ie})$. It has a strict start time $Ei$ and end time $Li$. Its processing time $P_i = Ei\text{-}Li$ is the duration of time the customer is following that segment of their path. (We assume the speed during that split job is constant.) Figure 4 shows an example time-space trajectory divided into five split jobs. The travel distance $D_{ij}$ from split job $i$'s end point or station $i$ to split job $j$'s start point or station $j$ is calculated using the Euclidean distance. These distances form an asymmetric network because $D_{ij}$ need not equal to $D_{ji}$ (the start and end points change).

We assume the following are given and constant parameters of the problem: split job start/end points, split job start/end times, station locations, current UAV locations and UAV fuel levels. We will use two indices for each LSS to distinguish between arriving UAVs and departing UAVs.

### 3.1 Notation

$i, j$: Indices for jobs
$s$: Index for stations

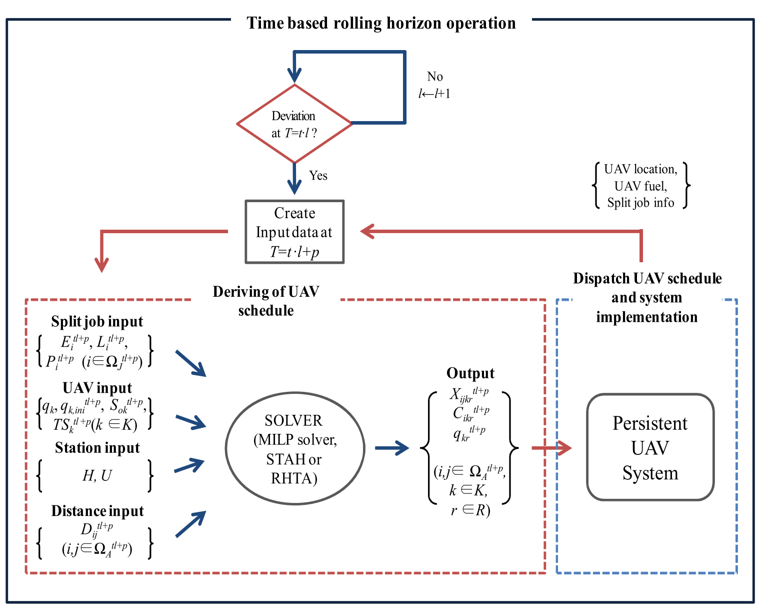



**Fig. 3** Time based rolling horizon operation

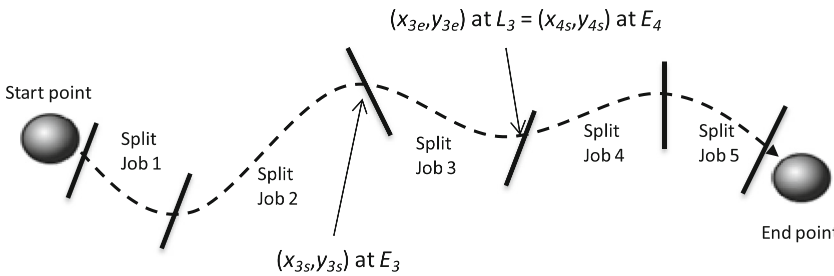



**Fig. 4** A time-space trajectory is divided into split jobs

$k$: Index for UAVs
$r$: Index of a UAV's $r^{\text{th}}$ 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 $j$
$Ei$: 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 $k$
$TS_k$: Travel speed of UAV $k$
$w_1$: Weight factor between objective criteria, $0 \leq w_1 \leq 1$.
$w_2$: Positive integer Scaling factor on the number of served jobs in objective function.
$\Omega_J$: $= \{1, \ldots, N_J\}$, Set of split jobs
$\Omega_{SS}$: $= \{N_J+1, N_J+3, \ldots, N_J+2\cdot N_{\text{STA}}-1\}$, set of UAV flight start stations
$\Omega_{SE}$: $= \{N_J+2, N_J+4, \ldots, N_J+2\cdot N_{\text{STA}}\}$, set of UAV flight end stations
$\Omega_A$: $= (\Omega_J U \Omega_{SS} U \Omega_{SE}) = \{1, \ldots, N_J + 2\cdot N_{STA}\}$, set of all jobs and recharge stations
$\Omega_{INI}$: $= \{1_{INI}, \ldots, K_{INI}\}$, 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 $r^{\text{th}}$ flight; 0, otherwise.
$C_{ikr}$: Real number decision variable, split job $i$'s start time by UAV $k$ during its $r^{\text{th}}$ 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

### 3.2 Recharge/Refuel Time Function

UAVs are assumed to fully replenish their fuel source when they visit an LSS. In this study, the duration of replenishment time depends on the remaining fuel level of the UAV:

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

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

where $H$ and $U$ are constants. For the first visit to an LSS, Eq. 1 is used to calculate the replenishment time of a UAV. After the first visit, Eq. 2 is applied because each UAV was fully charged in a previous station visit.

### 3.3 MILP Path Planning Formulation

Appendix A1 provides the MILP model for our system. The objective function (11) minimizes the sum of weighted total travel distance and number of served jobs. Weight factor $w_1$ balances the two objectives. The constant $w_2$ converts the objectives into the same unit. High values of $w_1$ will give energy efficient UAV schedules by the reducing flight distance (speeds are assumed constant in each split job and between them). However, it may decrease the service quality due to the presence of uncovered split jobs. On the other hand, low value of $w_1$ will guarantee service quality by serving as many as split jobs possible.

Constraints (12–17) coordinate the UAV paths. Constraint (12) ensures that each UAV starts its flight from its initial location or start station. It allows UAVs to serve split jobs or move to an LSS. Constraint (13) guarantees each UAV finish its flight at an LSS. The dual index for the LSSs is used in constraint (14). When UAV $k$ finishes its $r^{\text{th}}$ flight at LSS $s-1$, its $r+1^{\text{th}}$ flight starts at LSS $s$. Constraint (15) ensures that a UAV cannot finish its flight at its start station or UAV initial location (start stations have a different index than end stations). Constraint (16) required that split jobs in $\Omega_J$ be served by at most one UAV. As such, a split job need not be served at all. UAVs do not finish their flights at a split job via constraint (17).

Constraints from Eq. 18 to Eq. 23 determine the start time of split jobs and replenishment at LSSs. Constraint (18) requires the finish time of a UAV's $r^{\text{th}}$ flight and the start time of the UAV's $r+1^{\text{th}}$ flight to be same. Constraint (19) dictates that the split job start time or LSS visit start time (when consumable replenishment begins) equals that of the next task for each UAV in its $r^{\text{th}}$ flight. Constraint (20) determines the job start time of unserved split jobs. $C_{ikr}$ is set to zero if split job $i$ is not assigned to UAV $k$'s $r^{\text{th}}$ flight. Constraints (21) and (22) dictate the recharge time for each UAV and the split job start time after replenishment. They are distinguished based on the flight order. If $r=2$, which means the start of second flight, constraints (21) and Eq. 1 determine the initial fuel level of UAV $k(q_{k,ini})$. If $r>2$, constraint (22) and Eq. 2 are used instead. Constraint (23) forces UAVs to provide service at the correct start time of split job $i$ in $\Omega_J$.

Fuel restrictions are described via constraint (24), (25) and (26). Constraint (24) ensures that UAVs do not fly longer than they have fuel to fly. Constraints (25) and (26) ensure that the decision variables $q_{kr}$ obey their appropriate range.

Finally, constraints (27), (28) and (29) specify the real and binary decision variables for our MILP. We require that $N_R>1$ due to constraints (14) and (18). $N_R$ is the maximum number of flights allowed for each UAV; this nonnegative integer value can be set arbitrarily.

## 4 Addressing Computational Complexity

MILP formulations are computationally complex. In this section, we will discuss the problem complexity via transforming approach. We also discuss how we solve our MILPs as well as develop a heuristic and extend an RHTA approach to significantly improve computation.

### 4.1 Equivalence with an NP-hard Problem

In this section, we study the computational complexity of the proposed problem by transforming it to a capacitated vehicle routing problem (CVRP). Such problems belong to the NP-hard class; see [29]. The transformation approach is a widely used methodology to assess the complexity of such problems. In [30], the authors studied the complexity of a capacitated arc routing problem using a transformation to the CVRP. In [31], the authors transform their CVRP problem to a Multiple Travelling Salesman Problem and Bin Packing Problem. The following steps transform our problem to a CVRP.

**STEP1**: *Time* is removed from the model:

- 1-1) Decision variable $C_{ikr}$ (Eq. 28) is deleted.
- 1-2) Constants $E_i$, $L_i$ are deleted.
- 1-3) Constraints (19) $\sim$ (24) are deleted.

**STEP2**: Our heterogeneous UAVs are considered as identical (each with the same fuel capacity):

- 2-1) $TS_k$ is deleted from the model.
- 2-2) $Q_k$ is replaced with $Q$.

**STEP3**: Replace *fuel restriction* with *capacity restriction*

- 3-1) $Pi$ is considered as demand of customer $i$.
- 3-2) $Q$ stands for maximum capacity of vehicle.
- 3-3) Decision variable $q_{kr}$ is deleted:
  - – Equation 29 is deleted.
  - – Constraints (26) and (27) are deleted.
- 3-4) Change constraint (25) to

$$\sum_{i\in\Omega_J}\sum_{j\in\Omega_A} P_i\cdot X_{ijk}\leq Q\;(k\in K)$$

- 3-5) $D_{ij}$ is considered as travelling cost in CVRP.

**STEP4**: Only one flight is considered:

4-1) $X_{ijk}$ will replace the decision variable $X_{ijkr}$

T4-2) Dimension $r$ is deleted in constraints (12) $\sim$ (18) and (25)

4-3) We do not distinguish between start station and end station.

**STEP5**: Multiple depots will be unified into single depot:

5-1) $N_{STA}$ becomes 1.

5-2) '0' becomes the index of the single depot instead of set of UAV stations.

**STEP6**: Downgrade of problem characteristics:

6-1) Split jobs will be defined as stationary jobs

– $(x_{ie}, y_{ie}) = (x_{is}, y_{is})$. Therefore, $D_{ij} = D_{ji}$.

– Asymmetric graph becomes symmetric graph.

6-2) UAVs will initially located at *the* (single) depot instead of arbitrary initial locations.

6-3) Serve every customer instead of maximize job coverage:

– Objective function will be changed to

$$Minimize \sum_{i\in\Omega_A}\sum_{j\in\Omega_A}\sum_{k\in K} D_{ij}\cdot X_{ijk}.$$

– Change constraint (17) to

$$\sum_{k\in K}\sum_{i\in\Omega_A} X_{ijk} = 1\ (j\in\Omega_J).$$

Via the simplification approach, the transformed formulation is a capacitated vehicle routing problem. Therefore, the proposed problem also belongs to the NP-hard class. As discussed in [32], obtaining optimal solutions is time-consuming and computationally intractable. To address the computational tractability of our problem, we developed an efficient heuristic called STAH as well as an RHTA heuristic.

### 4.2 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. We will compare the results with our heuristics.

### 4.3 Sequential Task Assignment Heuristic

The content and presentation of this subsection first appeared in the conference paper [28]. We quote it here for clarity and completeness.

We develop the Sequential Task Assignment Heuristic (STAH) to address the computational intractability of the MILP formulation. Refer to Appendix A2 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).

Let $l$ be the split job index in $P_t, t \in P$. Starting with $t = 1$ and $l = 1$, 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 an LSS). 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 an LSS). 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 LSS 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$(a)$ = 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. Let $D(k)$ be the indicator for feasibility of the direct flight.

$$N_D(k) = max\{n \in Z_+ \mid D_{Cl(k),l'}/TS_k + \sum\nolimits_{i=l}^{l+n-1} Pi + D_{(l+n-1)'',s((l+n-1)'')}/TS_k) \leq Cq(k)\} \quad (3)$$

$$D(k) = 1-[I\{N_D(k) \geq 1\} * I\{A(k)+D_{Cl(k),l'}/TS_k \leq E_l\}] \quad (4)$$

Here $Z_+$ is the non-negative integers. $N_D(k) = 0$ if no such value exists. 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. Let $Ind(\text{k})$ be the indicator for feasibility of the indirect flight.

$$N_I(k) = max\{n \in Z_+ \mid D_{Cl(k),s(Cl(k),l')}/TS_k \leq Cq(k), D_{s(Cl(k),l'),l'}/TS_k + \sum\nolimits_{i=l}^{l+n-1} Pi + D_{(l+n-1)'',s((l+n-1)'')}/TS_k \leq q_k\} \quad (5)$$

$$Ind(k) = 1 - I\{N_I(k) \geq 1\} * I\left\{D_{Cl(k),s(Cl(k),l')}/TS_k + U + (H/q_k)\cdot[q_k - (Cq(k) - D_{Cl(k),s(Cl(k),l')}/TS_k)] + D_{s(Cl(k),l'),l'}/TS_k < E_l - A(k)\right\} \quad (6)$$

Here $Z_+$ is the non-negative integers. $N_I(k) = 0$ if no such value exists. The indicator function $I\{.\}$ is 1 if the condition $\{.\}$ is true, and 0 otherwise.

Together, these feasibility indicators will be used to check that a particular UAV assignment for the split job is feasible for constraints (23–24). They also enforce constraints (21–22). If no UAVs are feasible for the split job, the split job cannot be served. Set $l = l + 1$ and repeat the feasibility check.

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

$$V_D(\text{k}) = \left\{\alpha\cdot\beta\,(N_D\,(k)) - (1-\alpha)\cdot D_{Cl(k),l'}\right\} - M\cdot D(k) \quad (7)$$

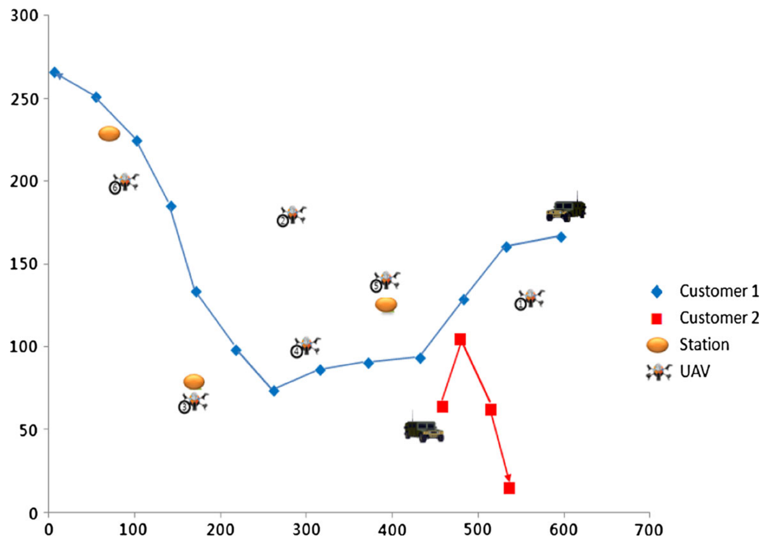
$$V_I\,(\text{k}) = \left\{\alpha\cdot\beta\,(N_I(k)) - (1-\alpha)\left(D_{Cl(k),s(kl')} + D_{s(kl'),l'}\right)\right\} - M\cdot Ind(k) \quad (8)$$

$M$ is a large positive value, $\alpha$ and $\omega$ 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 split jobs via that kind of flight. This procedure is repeated until all split jobs have been investigated or assigned. Then, we proceed to the next customer. After every split job in $\bigcup_{t\in P} P_t$ is investigated or assigned, all UAVs not at an LSS travel to the nearest station. STAH is complete.

## 4.4 Receding Horizon Task Assignment

The RHTA is a popular heuristic. We modify the RHTA$_\text{d}$ from [25] for our purposes. As our system components are determined initially, we remove the resource selection decisions and constraints from RHTA$_\text{d}$. The replenishment time for a UAV at an LSS

**Fig. 5** Geographical information of Example 1




is adjusted for our assumptions (1–2). Appendix A3 provides the detailed pseudo code. Internal to the $RHTA_d$ is an IP sub-problem; we remove the resource selection components from it. We delete constraints (12–13) in [25] and replace the objective function (1) in [25]:

$$Min\ w_1 \sum_{k=1}^{N_{UAV}} \sum_{p=1}^{N_{kp}} S_{kp} X_{kp} - (1 - w_1) \cdot w_2 \cdot \sum_{i \in W} \sum_{k=1}^{N_{UAV}} \sum_{p=1}^{N_{kp}} A_{kip} X_{kp} \tag{9}$$

We replace the constraint (11) in [25]:

$$\sum_{i \in W} \sum_{k=1}^{N_{UAV}} \sum_{p=1}^{N_{kp}} A_{kip} X_{kp} \leq P \tag{10}$$

## 5 Numerical Examples

In this section, we provide example problems to study computational behavior of the MILP, STAH and RHTA. The MILP was solved directly via CPLEX. These studies use a personal computer with Intel(R) Core(TM)2 Quad CPU Q8400, 2.66 GHz and 4.00 GB RAM.

### 5.1 Example 1: Scheduling Two Customers

The geographical locations of customer paths, LSSs and UAVs for Example 1 are depicted in Fig. 5. Split job data is given in Table 1. The trajectory of customer 1 was divided into 12 split jobs. Customer 2 requires 3 split jobs. UAV 3 and 5 are initially located at station 2 and 1, respectively. Other UAVs are located on the field initially. Initial fuel levels of each UAVs are set to {3, 8, 8, 6, 6, 8} and maximum

**Table 1** Splitjob information of example 1

| Customer | Split job | Start point x | Start point y | End point x | End point y | Start time |
|---|---|---|---|---|---|---|
| 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 |

**Table 2** Split job information at $T = 11$

| Customer | Split job | Start point x | Start point y | End point x | End point y | Start time |
|---|---|---|---|---|---|---|
| 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 |

travelling time of UAVs are set to 12. The three stations are located at the x-y coordinates (394, 126), (170, 79) and (72, 229). The LSS constants $H = 3$ and $U = 0.5$.

CPLEX, STAH and RHTA are implemented and compared in terms of total travelling distance and computational time. $w_1$ and $w_2$ are set to 0.3 and 50. For STAH, $\alpha$ and $\beta$ are set to 0.3 and 50. We set P as 5 for RHTA. The MILP via CPLEX obtained an optimal solution with 809.067 of total travelling distance in 71.59 seconds. All 15 split jobs were served. Let $I_k$ be the initial location of each UAV, $k \in K$, $1, 2, \ldots, 15$ be the split job index and $S_1$, $S_2$, $S_3$ be the station notation. CPLEX determines the UAV split job assignments as follows; UAV1 : $\{I_1, 1, 2, S_1\}$, UAV2: $\{I_2, S_1, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, S_3\}$, UAV3: $\{I_3(S_2)\}$, UAV4: $\{I_4, 13, 14, 15, S_1\}$, UAV5: $\{I_5(S_1)\}$ and UAV6: $\{I_6, S_3\}$. STAH obtained the same travelling distance with an alternate UAV schedule in 0.006 seconds. In the STAH solution, UAV 5 performs split jobs from 3 to 12 instead of UAV 2. UAV 2 is tasked to do $\{I_2, S_1\}$ while UAV5 conducts the sequence of split jobs $\{I_5(S_1), 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, S3\}$. RHTA served every split job and obtained a solution with 911.949 total travelling distance in 0.187 s. This is a 12.72 % gap.

### 5.2 Example 2: Real Time Operation

We now consider real time operation. During the UAV service of Example 1, a new customer request arrives at time $T = 10$. Customer 3 wishes to be escorted from $T = 13$ to $T = 19$. Therefore to serve customer 1 and 3, we reallocate system tasks from $T = 11$. All split jobs for customer 2 have been served by $T = 11$. Table 2 provides information on the 6 remaining split jobs for customer 1 and 6 newly added split jobs for customer 3 from $T = 11$.

At $T = 11$, UAV 1 and 5 are at station 1 with fully charged energy sources. UAV 2 is at the end point of split job 6 in Table 1 with 7.629 remaining fuel. UAV 3 is fully charged and located at station 2. UAV 4 is at the end point of split job 15 in Table 1 with 2.325 remaining fuel. Fully charged UAV 6 is at station 3. Figure 6 depicts the layout.

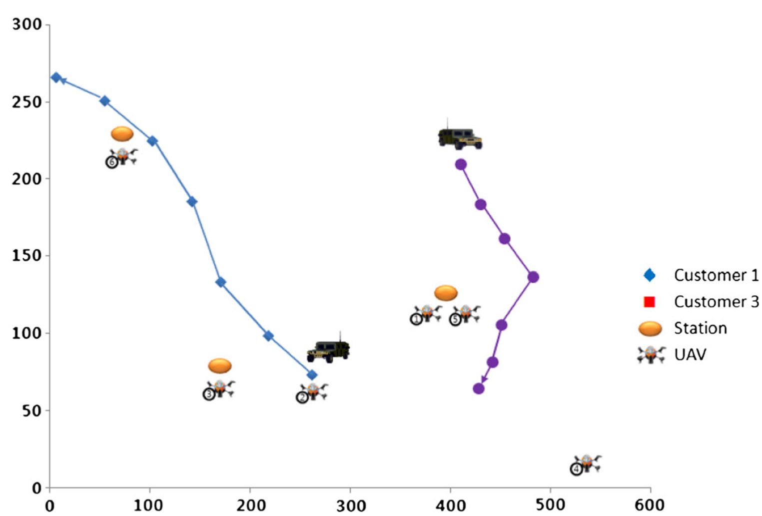



**Fig. 6** Geographical information at T=11

**Table 3** Numerical comparison on problem size

| System parameters | | | CPLEX | | | | STAH | | | | | RHTA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $NJ$ | $N_{STA}$ | $NUAV$ | CPU time | Total travelling distance | Num of served job | Obj. value | CPU time | Total travelling distance | Num of served job | Obj. value | Gap (%) | CPU time | Total travelling distance | Num of served job | Obj. value | Gap (%) |
| 15 | 3 | 6 | 8.47 | 662.570 | 15 | 198.77 | 0.004 | 665.261 | 15 | 199.58 | 0.406 | 0.186 | 664.857 | 15 | 199.46 | 0.345 |
| 15 | 3 | 6 | 71.59 | 809.067 | 15 | 242.72 | 0.006 | 809.067 | 15 | 242.72 | 0.000 | 0.190 | 911.949 | 15 | 273.58 | 12.72 |
| 30 | 3 | 6 | N/A | N/A | N/A | – | 0.008 | 1010.24 | 30 | 303.07 | – | 0.277 | 1236.70 | 30 | 371.01 | – |
| 45 | 3 | 6 | N/A | N/A | N/A | – | 0.015 | 1314.34 | 45 | 394.30 | – | 0.368 | 1811.63 | 45 | 543.49 | – |
| 45 | 5 | 10 | N/A | N/A | N/A | – | 0.015 | 1346.96 | 45 | 404.09 | – | 0.500 | 1656.67 | 45 | 497.00 | – |
| 60 | 3 | 6 | N/A | N/A | N/A | – | 0.016 | 1642.34 | 60 | 492.70 | – | 0.458 | 2390.35 | 60 | 717.11 | – |
| 60 | 5 | 10 | N/A | N/A | N/A | – | 0.031 | 1565.90 | 60 | 469.77 | – | 0.645 | 1997.99 | 60 | 599.40 | – |

The revised schedule assigns UAV 2 to the remaining trajectory of customer 1 as in the original schedule. UAV 5 serves customer 3 instead of resting at station 1 in the example 1.

### 5.3 Various Problem Sizes

Consider Example 1. We vary the number of split jobs used for the same customer paths. Refer to Table 3. $w_1$ and $w_2$ were set to 0.3 and 50. For STAH, $\alpha$ and $\beta$ were set to 0.3 and 50. We set the initial fuel of each UAV $(q_{1,ini}, \ldots, q_{6,ini}) = (6, 8, 8, 10, 6, 8)$ and (3,8,8,6,6,8) for cases of $(\mathrm{N_J}, \mathrm{N_{STA}}, \mathrm{N_{UAV}}) = (15, 3, 6)$. Initial fuel level significantly changes the computation time. Due to the higher level of initial fuel, the sum of travelling distances for the UAVs is reduced.

For the cases with more than $\mathrm{N_J} = 15$ split jobs, initial fuel of each UAV $(q_{1,ini}, \ldots, q_{10,ini})$ was set to (6,8,8,10,6,8,6,8,8,10). CPLEX issues an out of memory error; it cannot solve the problem. STAH and RHTA can derive feasible solutions with a second. STAH is at least 31 times faster than RHTA in these examples. Due to the strong computational power, STAH and RHTA may be effective for use in task planning in real time UAV service.

## 6 Concluding Remarks

We developed a task planning model for a system of UAVs to provide nearly uninterrupted security escort service. Logistic service stations (LSSs) enable the persistent operation of the system. A mixed integer linear program (MILP) was developed which allows for arbitrary initial UAV locations and fuel levels. As such it can be used in a rolling horizon formulation. The objective balances between system service quality (via the minimizing number of split jobs not served) and energy efficiency (via minimizing the total flight distance). As the problem is NP-hard, we developed a new heuristic and extended a classic one. The heuristic that we developed, STAH, was inspired by features observed in optimal solutions obtained in small problems (and solved via CPLEX). The RHTA heuristic was extended based on one in [25]. We studied the computational behavior of the approaches via examples. RHTA and STAH were at least 45 and 2,100 times faster than CPLEX, respectively. Their solutions, though suboptimal, were acceptable. We also considered an example in which a new customer arrived and the schedule required adjustment. Real time operation, including disturbances such as new customers, unexpected disturbances and deviations from the plan may be addressed by the rolling horizon formulation.

**Acknowledgments** The work reported here has been supported by National Research Foundation of Korea (NRF) 2013R1A1A2012292

## Appendix

A1: MixedInteger Linear Program

$$Minimize\ w_1 \sum_{i\in\Omega_A}\sum_{j\in\Omega_A}\sum_{k\in K}\sum_{r\in R} D_{ij}\cdot X_{ijkr} + (1-w_1)\cdot w_2 \cdot \left(|J| - \sum_{i\in\Omega_J}\sum_{\Omega_J\cup\Omega_{SE}}\sum_{k\in K}\sum_{r\in R} X_{ijkr}\right) \tag{11}$$

*Subject to*

$$\sum_{i\in\Omega_{SS}\cup\Omega_{INI}}\sum_{j\in\Omega_J\cup\Omega_{SE}} X_{ijkr} = 1 \quad (k\in K, r\in R) \tag{12}$$

$$\sum_{i\in\Omega_J\cup\Omega_{SS\cup u}\Omega_{INI}}\sum_{s\in\Omega_{SE}} X_{iskr} = 1 \quad (k\in K, r\in R) \tag{13}$$

$$\sum_{i\in\Omega_J\cup\Omega_{SS}} X_{iskr} = \sum_{i\in\Omega_J\cup\Omega_{SE}} X_{s-1,ikr+1} \quad (k\in K, r=1,\ldots N_R-1, s\in\Omega_{SE}) \tag{14}$$

$$\sum_{i\in\Omega_J\cup\Omega_{SS}} X_{iskr} = 0 \quad (k\in K, r\in R, s\in\Omega_{SS}\cup\Omega_{INI}) \tag{15}$$

$$\sum_{k\in K}\sum_{r\in R}\sum_{i\in\Omega_A} X_{ijkr} \le 1 \quad (j\in\Omega_J) \tag{16}$$

$$\sum_{j\in\Omega_A} X_{ijkr} - \sum_{j\in\Omega_A} X_{jikr} = 0 \quad (i\in\Omega_J, k\in K, r\in R) \tag{17}$$

$$C_{skr} = C_{s-1,kr+1} \quad (k\in K, r=1,\ldots N_R-1, s\in\Omega_{SE}) \tag{18}$$

$$C_{ikr} + P_i + \frac{D_{ij}}{TS_k} - C_{jkr} \le M\left(1 - X_{ijkr}\right) \quad (i\in\Omega_J, j\in\Omega_J\cup\Omega_{SE}, k\in K, r\in R) \tag{19}$$

$$M\cdot\sum_{j\in\Omega_J\cup\Omega_{SE}} X_{ijkr} \ge C_{ikr} \quad (i\in\Omega_J\cup\Omega_{SS}, k\in K, r\in R) \tag{20}$$

$$C_{ikr} + RT_f + \frac{D_{ij}}{TS_k} - C_{jkr} \le M\left(1 - X_{ijkr}\right) \quad (i\in\Omega_{SS}, j\in\Omega_J\cup\Omega_{SE}, k\in K, r=2) \tag{21}$$

$$C_{ikr} + RT_r + \frac{D_{ij}}{TS_k} - C_{jkr} \le M\left(1 - X_{ijkr}\right) \quad (i\in\Omega_{SS}, j\in\Omega_J\cup\Omega_{SE}, k\in K, r>2) \tag{22}$$

$$\sum_{k\in K}\sum_{r\in R} C_{ikr} = E_i \quad (i\in\Omega_J) \tag{23}$$

$$\sum_{i\in\Omega_A}\sum_{j\in\Omega_A}\frac{D_{ij}}{TS_k}\cdot X_{ijkr} + \sum_{i\in\Omega_J}\sum_{j\in\Omega_A} P_i\cdot X_{ijkr} \le q_{kr} \quad (k\in K, r\in R) \tag{24}$$

$$q_{kr} \le q_{k,ini} \quad (k\in K, r=1) \tag{25}$$

$$q_{kr} \le q_k \quad (k\in K, r\neq 1) \tag{26}$$

$$C_{ikr} \ge 0 \quad (k\in K, r\in R, i\in\Omega_A) \tag{27}$$

$$q_{kr} \ge \quad (k\in K, r\in R) \tag{28}$$

$$X_{ijkr} \in \{0,1\} \quad (i\in\Omega_A, j\in\Omega_A k\in K, r\in R) \tag{29}$$

A2: Pseudo Code of STAH

**Algorithm 1** Sequential task assignment heuristic

Let $\mathrm{MD_D}(k)$ and $\mathrm{MD_I}(k)$ be the moving distance by direct and indirect flight of UAV k.<br>
Let $l$ be the split job index in $P_t, t \in P$. Let Obj.value be the objective value and set Obj.value =0.<br>
Set Obj.value = 0, D(k)= 0, Ind(k)= 0, $\mathrm{MD_D}(k) = 0$, $\mathrm{MD_I}(k) = 0$, $t = 1$ and $l = 1$.<br>
**For** all customer t, \\\\ {from customer 1 to $|P|$}<br>
  **While** $l < |P_t|$, **do** \\\\ { find feasible split job that UAVs can serve}<br>
    **While** D(k)+Ind(k) = 0, **do**<br>
      Calculate D(k) and Ind(k) for all UAV k.<br>
      **IF** $\sum_{k \in K} D(k) + Ind(k) = 0$, **do**<br>
        $l \leftarrow l+1$.<br>
      **IF** $l = |P_t|$, **do** \\\\ {break while and investigate next customer}<br>
        **break;**<br>
  **For** all UAV k, calculate $V_D(k)$ and $V_I(k)$. Select the biggest Value.<br>
    Let $k$’ be the selected UAV.<br>
    **IF** $V_D(k')$ is selected, **do**<br>
     $\mathrm{MD_D}(k) += \left(\mathrm{D}_{Cl(k),l'}\right)$; \\\\ {Calculate moving distance of direct flight of UAV k’}<br>
     Obj.value$+= \left(\mathrm{w_1} \cdot \mathrm{MD_D}(k) - (1 - \mathrm{w_1}) \cdot \mathrm{w_2} \cdot N_{\mathrm{D}}\left(k'\right)\right)$;<br>
     $\mathrm{Cl}(k') \leftarrow (l+\mathrm{N_D}\left(k'\right) - 1)''$; \\\\ {updated UAV k’ information}<br>
     $\mathrm{Cq}(k') \leftarrow \mathrm{Cq}(k') - \mathrm{MD_D}(k)/\mathrm{TS}(k') - P \cdot \mathrm{N_D}(k')$;<br>
     $\mathrm{A}(k') \leftarrow \mathrm{L}_{l+\mathrm{N_D}(k')-1}$;<br>
     $l \leftarrow l + \mathrm{N_D}\left(k'\right)$; \\\\ {updated earliest start split job information}<br>
     $l' \leftarrow (l + \mathrm{N_D}(k'))'$;<br>
     $l \leftarrow (l + \mathrm{N_D}(k'))''$;<br>
**ELSE IF** $V_I(k')$ is selected, **do**<br>
     $\mathrm{MD_I}(k) += \left(\mathrm{D}_{Cl(k),s(kl')} + \mathrm{D}_{s(kl'),l'}\right)$; \\\\ {Calculate moving distance of indirect flight of UAV k’}<br>
     Obj.value$+= \left(\mathrm{w_1} \cdot \mathrm{MD_I}(k) - (1 - \mathrm{w_1}) \cdot \mathrm{w_2} \cdot N_{\mathrm{I}}\left(k'\right)\right)$;<br>
     $\mathrm{Cl}(k') \leftarrow (l + \mathrm{N_I}\left(k'\right) - 1)''$; \\\\ { updated UAV k’ information}<br>
     $\mathrm{Cq}(k') \leftarrow q(k')\mathrm{D}_{s\left(kl'\right),l'}/\mathrm{TS}(k) - P \cdot \mathrm{N_I}(k')$;<br>
     $\mathrm{A}(k') \leftarrow \mathrm{L}_{l+\mathrm{N_I}\left(k'\right)-1}$;<br>
     $l \leftarrow l + \mathrm{N_I}(k')$; \\\\ {updated earliest start split job information}<br>
     $l' \leftarrow (l + \mathrm{N_I}(k'))'$;<br>
     $l \leftarrow (l + \mathrm{N_I}(k'))''$;<br>
**End for** \\\\ Finish UAV assignment<br>
**For** all UAV k, **do** \\\\ {send UAVs to the stations which are not located in station}<br>
 **IF** Cl(k) $!\in \Omega_{\mathrm{SE}}$<br>
s(Cl(k)) = argmin $s \in \Omega_{SE}\{D\mathrm{Cl}(k)_{,s}\}$<br>
 $obj.value += \left(\mathrm{D}_{\mathrm{Cl(k)},\mathrm{S_{cl(k)}}}\right)$;<br>
**End for**

A3: Pseudo Code of RHTA

**Algorithm 2** Receding horizon task assignment

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$\leftarrow$ 1;
**For** all numbers $n_c$ of jobs to visit $n_c \leftarrow 1, \ldots, P$ **do**
**For** all combinations C of nc jobs **do**
**For** all permutations i of jobs $[w_1, \ldots, w_{nc}]$ in C, with $i \leftarrow 1 \ldots n_c!$ **do**
**For** all station s **do**
**if** $(D(cL(k), w_1) + \sum_{i=1}^{i=n_c-1} D(w_i, w_{i+1}) + D(W_{nc}, s) + \sum_{i=1}^{i=n_c} P(w_i) \leq rF(k)$ 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 $i = 1, \ldots$,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;**
$i_{min} \leftarrow \mathrm{argmin}_i\, S_i$; \\ { Choose the best feasible permutation} $S_{vp} \leftarrow S_{imin}$; $P_{vp} \leftarrow P_{imin}$; p $\leftarrow$ 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**
$w_{opt} \leftarrow P_{vp}$ (1); \\ {Pick the first job in the permutation}
$M_k \leftarrow [M_k w_{opt}]$; \\ { Adds the job to the mission list of UAV k}
rF(k) $\leftarrow$ rF(k) – D(cL(k), $w_{opt}$)/TS(k) – P($w_{opt}$); \\ { update remaining fuel time of UAV k}
at(k) $\leftarrow$ E($w_{opt}$)+P($w_{opt}$); \\ {update available time}
cL(k) $\leftarrow$ $w_{opt}$; \\ { update current location}
W $\leftarrow$ 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**
Assign UAV to closest station ($S_{min}$)
$M_k \leftarrow [M_k, s_{min}]$; rF(k)$\leftarrow$ $max$F(k); at(k)$\leftarrow$at(k) + D(cl(k),s) + U +H$\cdot$ (maxF(k)-rF(k)) /maxF(k) ;cL(k)$\leftarrow$smin ;
**for** all UAV k **do** // { send UAV at end of job to station}
**if** !(cL(k)==s)**do**
Assign UAV to closest station;

## References


1. Valenti, M., Dale, D., How, J., Vian, J.: Mission health management for 24/7 persistent surveillance operations. In: Proceedings of the AIAA Guidance, Control and Navigation Conference, Myrtle Beach, SC, August (2007)
2. Bethke, B., Valenti, M., How, J.: UAV task assignment. IEEE Robot. Autom. Mag. **15**(1), 39–44 (2008)
3. Nigam, N., Bieniawski, S., Kroo, I., Vian, J.: Control of multiple UAVs for persistent surveillance: algorithm flight test results. IEEE Trans. Control Syst. Technol. **20**(5), 1236–1251 (2012)
4. Ure, N.K., Chowdhary, G., Toksoz, T., How, J.P., Vavrina, M.A., Vian, J.: An automated battery management system to enable persistent missions with multiple aerial vehicles. IEEE/ASME Trans. Mechatron. **20**(1), 275–286 (2015)
5. Kemper, P.F., Suzuki, K.A.O., Morrison, J.R.: UAV consumable replenishment: design concepts for automated service station. J. Intell. Robot. Syst. **61**(1), 369–397 (2011)
6. Suzuki, K.A.O., Kemper, P.F., Morrison, J.R.: Automated battery replacement system for UAVs: analysis and design. J. Intell. Robot. Syst. **65**(1), 563–586 (2012)
7. Swieringa, K.A., Hanson, C.B., Richardson, J.R., White, J.D., Hasan, Z., Qian, E., Girard, A.: Autonomous battery swapping systems for small-scale helicopters. In: Proceedings of the IEEE International Conference on Robotics and Automation(ICRA), Anchorage, Alaske, pp. 3335–3340 (2010)
8. Godzdanker, R., Rutherford, M.J., Valavanis, K.P.: ISLAND: a self-leveling landing platform for autonomous miniature UAVs. In: Proceedings IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Budapest, Hungary, pp. 170–175 (2011)
9. Fujii, K., Higuchi, K., Pekimoto, J.: Endless flyer: a continuous flying drone with automatic battery replacement. In: Proceedings of the IEEE International Conference on Ubiquitous Intelligence & Computing and IEEE International Conference on Autonomic & Trusted Computing, Sorrento Peninsula, Italy, pp. 216–223 (2013)
10. Kim, W., Ji, K., Ambike, A.: Real-time operating environment for networked control systems. IEEE Trans. Autom. Sci. Eng. **3**(3), 287–296 (2005)
11. Geramifard, A., Redding, J., Roy, N., How, J.P.: UAV cooperative control with stochastic risk models, pp. 3393–3398 (2011)
12. Cocchioni, F., Pierfelice, V., Benini, A., Mancini, A., Frontoni, E., Zingaretti, P., Ippoliti, G., Longhi, S.: Unmanned ground and aerial vehicles in extended range indoor and outdoor missions (2014)
13. Sharma, R.: Observability based control for cooperative localization. In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, pp. 134–139 (2014)
14. Ponda, S.S., Johnson, L.B., Geramifard, A., How, J.P.: Cooperative mission planning for multi-UAV teams. Handbook of Unmanned Aerial Vehicles, pp. 1447–1490 (2014)
15. Geramifard, A., Redding, J., How, J.P.: Intelligent cooperative control architecture: a framework for performance improvement using safe learning. J. Intell. Robot. Syst. **73**(1), 83–103 (2013)
16. Ure, N.K., Chowdhary, G., How, J.P., Vavrina, M.A., Vian, J.: Health aware planning under uncertainty for UAV missions with heterogeneous teams. In: Proceedings European Control Conference, Zurich, Switzerland, pp. 3312–3319 (2013)
17. Shima, T., Schumacher, C.: Assignment of cooperating UAVs to simultaneous tasks using genetic algorithm. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco (2005)
18. Zeng, J., Yang, X., Yang, L., Shen, G.: Modeling for UAV resource scheduling under mission synchronization. J. Syst. Eng. Electron. **21**(5), 821–826 (2010)
19. Weinstein, A.L., Schumacher, C.: UAV scheduling via the vehicle routing problem with time windows. In: Proceedings of the AIAA Infotech@Aerospace 2007 Conference and Exhibit, Rohnert Park, California (2007)
20. Kim, Y.S., Gu, D.W., Postlethwaite, I.: Real-time optimal mission scheduling and flight path selection. IEEE Trans. Autom. Control **52**(6), 1119–1123 (2007)
21. 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**(4), 839–843 (2009)
22. Shetty, V.K., Sudit, M., Nagi, R.: Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles. Comput. Oper. Res. **35**(6), 1813–1828 (2008)
23. Kim, J., Song, B.D., 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)
24. Song, B.D., Kim, J., Kim, J., Park, H., Morrison, J.R., Shim, D.H.: Persistent UAV service: an improved scheduling formulation and prototypes of system components. J. Intell. Robot. Syst. **74**(1), 221–232 (2014)
25. Kim, J., Morrison, J.R.: On the concerted design and scheduling of multiple resources for persistent UAV operations. J. Intell. Robot. Syst. **74**(1), 479–498 (2014)
26. Kim, J., Shim, D.H., Morrison, J.R.: Tablet PC-based visual target-following system for quadrotors. J. Intell. Robot. Syst. **74**(1), 85–95 (2014)
27. Sundar, K., Rathinam, S.: Algorithms for routing an unmanned aerial vehicle in the presence of refeuling depots. IEEE Trans. Autom. Sci. Eng. **11**(1), 287–294 (2014)
28. Song, B.D., Kim, J., Morrison, J.R.: Towards real time scheduling for persistent UAV service: A rolling horizon MILP approach, RHTA and the STAH heuristic, pp. 505–516 (2014)
29. Toth, P., Vigo, D.: Models, relaxations and exact approaches for the capacitated vehicle routing problem. Discret. Appl. Math. **123**(1–3), 487–512 (2002)
30. Longo, H., Aragao, M.P., Uchoa, E.: Solving capacitated arc routing problems using a transformation to the CVRP. Comput. Oper. Res. **33**(6), 1823–1837 (2006)
31. Ralphs, T.K., Kopman, L., Pulleyblank, W.R., Trotter, L.E.: On the capacitated vehicle routing problem. Math. Program. **94**(2–3), 343–359 (2003)
32. Ho, W., Ho, G.T.S., Ping, J., Lau, H.C.W.: A hybrid genetic algorithm for the multi-depot vehicle routing problem. Eng. Appl. Artif. Intell. **21**(4), 548–557 (2008)

**Byung Duk Song** received the M.S. and Ph.D degrees in Industrial and Systems Engineering from Korea Advanced Institute of Science and Technology (KAIST) and B.S. degree in Industrial Engineering from Korea University. His research interests include vehicle routing problem, manufacturing process optimization, scheduling and supply chain management. His email address is <bdsong@kaist.ac.kr>

**Jonghoe Kim** received B.S., M.S. and Ph.D. degrees in Industrial and Systems Engineering from KAIST (Korea Advanced Institute of Science and Technology), South Korea. He is currently an Associate Research Fellow in the Defense Modeling and Simulation Division, KIDA (Korea Institute for Defense Analyses), South Korea. His research interests include vehicle routing problem, concerted system design and scheduling, real time system control and discrete event simulation. His email address is <windf@kaist.ac.kr>

**James R. Morrison** received one B.S. degree in electrical engineering and a second B.S. degree in mathematics from the University of Maryland, College Park. He received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana–Champaign, USA, in 1997 and 2000, respectively. He was with the Fab Operations Engineering Department at the IBM Corporation from 2000 to 2005. He is currently an Associate Professor in the Department of Industrial and Systems Engineering, KAIST, South Korea. His research interests include semiconductor wafer fabrication, robotic service agents, design of services and ecodesign. Prof. Morrison has served as the corresponding Co-Chair of the IEEE Robotics and Automation Society Technical Committee on Semiconductor Manufacturing Automation since January 2009.