Connection Science

ISSN: 0954-0091 (Print) 1360-0494 (Online) Journal homepage: www.tandfonline.com/journals/ccos20

UAV mission scheduling with completion time, flight distance, and resource consumption constraints

Keqin Li

To cite this article: Keqin Li (2023) UAV mission scheduling with completion time, flight distance, and resource consumption constraints, Connection Science, 35:1, 2281250, DOI: 10.1080/09540091.2023.2281250

To link to this article: https://doi.org/10.1080/09540091.2023.2281250

Published online: 10 Nov 2023.

CONNECTION SCIENCE
2023, VOL. 35, NO. 1, 2281250
https://doi.org/10.1080/09540091.2023.2281250

UAV mission scheduling with completion time, flight distance, and resource consumption constraints

Keqin Li

Department of Computer Science, State University of New York, New Paltz, NY, USA

ABSTRACT

Unmanned aerial vehicles (UAVs) are widely used in various military and civilian applications. UAV mission scheduling is a key issue in UAV applications and a central topic in UAV research. UAV task scheduling should include several constraints into consideration, such as completion time constraint, flight distance constraint, and resource consumption constraint. Furthermore, UAV task scheduling should be studied within the traditional framework of combinatorial optimisation. In this paper, we consider optimal mission scheduling for heterogeneous UAVs with completion time, flight distance, and resource consumption constraints. The contributions of the paper are summarised as follows. We define two combinatorial optimisation problems, namely, the NFTM (number of finished tasks maximisation) problem and the RFTM (reward of finished tasks maximisation) problem. We construct an algorithmic framework for both NFTM and RFTM problems, so that our heuristic algorithms (four for NFTM and two for RFTM) can be presented in a unified way. We derive upper bounds for optimal solutions, so that our heuristic solutions can be compared with optimal solutions. We experimentally evaluate the performance of our heuristic algorithms. To the best of our knowledge, this is the first paper studying UAV mission scheduling with time, distance, and resource constraints as combinatorial optimisation problems.

ARTICLE HISTORY
Received 12 October 2023
Accepted 1 November 2023

KEYWORDS
Algorithmic framework; combinatorial optimisation; heterogeneous UAVs; heuristic algorithm; mission scheduling

1. Introduction

1.1. Background

Unmanned aerial vehicles (UAVs), also called drones, are widely used in various military and civilian applications, such as construction inspection, disaster management, forest restoration, precision agriculture, remote sensing, search and rescue, security and surveillance, traffic monitoring (SCE). UAVs have created a new type of distributed systems and dynamic environments (Machovec et al., 2023).

UAV *mission scheduling* is a key issue in UAV applications and a central topic in UAV research. A typical scenario involves multiple heterogeneous UAVs (with different initial locations, flight speeds, maximum flight distances, maximum flight times, and maximum

CONTACT Keqin Li lik@newpaltz.edu Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

resource consumptions) and multiple heterogeneous tasks (with different positions, processing times, deadlines, resource requests, and rewards). UAVs are actually mobile and location-sensitive (Tan et al., 2021) servers with limited energy capacity (C.-I. Li et al., 2021), which move around to process tasks. UAV mission scheduling is essentially to dispatch UAVs to fly, process, and complete tasks with various optimisation objectives under various conditions and constraints.

UAV task scheduling (especially for rescue missions (Alhaqbani et al., 2021) and other similar tasks) should include several constraints into consideration, such as completion time constraint, flight distance constraint, and resource consumption constraint. (1) Completion time constraint – For a time-critical task, there is a deadline to complete the task. For instance, in disaster rescue, the survival time of a victim is very limited. (2) Flight distance constraint – A UAV has certain flight distance limitation due to limited energy supply (fuel or electricity). (3) Resource consumption constraint – A UAV can only carry a certain amount of resources required and requested by tasks due to limited capacity and space. Furthermore, UAV task scheduling should be studied within the traditional framework of combinatorial optimisation (K. Li, 2023).

1.2. Related work

It has been pointed out that there are two main considerations in UAV mission scheduling, i.e. task assignment and flight planning (Bellingham et al., 2003; Peng et al., 2021; Sebbane, 2021). Extensive research has been conducted in *task assignment* [including such methods as fish-inspired algorithm (Alhaqbani et al., 2021), autonomous task allocation (Aljalaud & Kurdi, 2021), leader-follower coalition (J. Chen & Sun, 2011), dynamic grouping allocation (X. Chen et al. 2019), distributed task allocation (Cui et al., 2022), decentralised auction algorithm (Hu & Yang, 2018), double-layer deep reinforcement learning (Mao et al., 2022), human-agent collaboration (Ramchurn et al., 2015), mixed integer linear program (Schumacher et al., 2003), negotiation (Sujit et al., 2006), simulation-based system (Sung et al., 2019), team-based approach (Venugopalan et al., 2015), quantum genetic algorithm (Z. Wang & Yan, 2021), digital twin (Yi et al., 2023), and clone selection (Zhang & Chen, 2021)] and *flight planning* [including such methods as decentralised algorithm (Bertuccelli et al., 2009), neural network (Filho et al., 2022), auction algorithm (Fu et al., 2019), cooperative planning (L. Geng et al. 2014), particle swarm optimisation (N. Geng et al., 2021), simulated annealing and local search (Ozkan, 2021), auction bidding and resolution (Sullivan et al. 2019), distributed particle swarm optimisation (Y. Wang et al., 2019), online algorithm (Yao & Ansari, 2020), deep migration reinforcement learning (Yin et al., 2022), and bat algorithm (Zhou et al., 2021)]. It is also noticed that the two problems of task allocation and route planning should be considered together (Yan et al., 2021).

A combinatorial optimisation approach has been adopted in K. Li (2023), where task scheduling on heterogeneous UAVs was treated as NP-hard optimisation problems and heuristic algorithms were designed and analysed. However, task completion time constraint, UAV flight distance constraint, and UAV resource consumption constraint were not taken into account.

1.3. Contributions

In this paper, we consider optimal (rescue) mission scheduling for heterogeneous UAVs with completion time (e.g. survival time), flight distance, and resource consumption constraints. The contributions of the paper are summarised as follows.

- We define two combinatorial optimisation problems, namely, the NFTM (number of finished tasks maximisation) problem and the RFTM (reward of finished tasks maximisation) problem.
- We construct an algorithmic framework for both NFTM and RFTM problems, so that our heuristic algorithms (four for NFTM and two for RFTM) can be presented in a unified way.
- We derive upper bounds for optimal solutions, so that our heuristic solutions can be compared with optimal solutions.
- We experimentally evaluate the performance of our heuristic algorithms.

To the best of our knowledge, this is the first paper studying UAV mission scheduling with time, distance, and resource constraints as combinatorial optimisation problems.

The rest of the paper is organised as follows. In Section 2, we present preliminary information, including a UAV mission scheduling model and our problem definitions. In Section 3, we develop our heuristic algorithms. In Section 4, we derive upper bounds for optimal solutions. In Section 5, we conduct an experimental performance evaluation for our heuristic algorithms. In Section 6, we summarise the paper.

2. Preliminaries

In this section, we present preliminary information, including a UAV mission scheduling model and our problem definitions. Table A1 lists all the notations and definitions used in the paper.

2.1. Scheduling model

In this section, we describe our UAV mission scheduling model.

Our model includes m UAVs: $u_1, u_2, \ldots, u_m$, and n missions (tasks) in a three-dimensional space.

A UAV is specified as $u_i = (position(u_i), speed(u_i), maxdistance(u_i), maxresource(u_i))$, where $position(u_i)$ is the initial location of u_i, $speed(u_i)$ is the flight speed of u_i, $maxdistance(u_i)$ is the maximum flight distance of u_i, and $maxresource(u_i)$ is the maximum resource consumption of u_i. For convenience, we also define $maxtime(u_i)$ to be the maximum flight time of u_i, which is $maxdistance(u_i)/speed(u_i)$.

Let $L = (t_1, t_2, \ldots, t_n)$ be a list of tasks. A task is specified as $t_j = (position(t_j), ptime(t_j), deadline(t_j), request(t_j), reward(t_j))$, where $position(t_j)$ is the position of t_j, $ptime(t_j)$ is the processing time of t_j, $deadline(t_j)$ is time deadline to complete t_j, $request(t_j)$ is the amount of resource request of t_j, and $reward(t_j)$ is the reward of completing t_j.

Our scheduling problems essentially find $route(u_i)$, i.e. the flight route of u_i, for all i. A flight route is actually $(t_{i,1}, t_{i,2}, \ldots, t_{i,n_i})$, i.e. a sequence of tasks.

Let $dist(p, q)$ be the distance between locations p and q.

For a $route(u_i) = (t_{i,1}, t_{i,2}, \ldots, t_{i,n_i})$ of u_i, the total flight distance of u_i is

$$distance(u_i) = dist(position(u_i), position(t_{i,1})) + \sum_{k=1}^{n_i-1} dist(position(t_{i,k}), position(t_{i,k+1})).$$

The total flight time of u_i is

$$ftime(u_i) = distance(u_i)/speed(u_i).$$

The total processing time of u_i is

$$ptime(u_i) = \sum_{k=1}^{n_i} ptime(t_{i,k}).$$

The total time of u_i is the total flight time + the total processing time:

$$time(u_i) = ftime(u_i) + ptime(u_i).$$

The total resource consumption of u_i is

$$resource(u_i) = \sum_{k=1}^{n_i} request(t_{i,k}).$$

Initially, the current location of u_i is

$$location(u_i) = position(u_i),$$

and when u_i moves to $t_{i,k}$,

$$location(u_i) = position(t_{i,k}).$$

The completion time of $t_{i,k}$ is

$$ctime(t_{i,k}) = \frac{1}{speed(u_i)} \left(dist(position(u_i), position(t_{i,1})) + \sum_{k'=1}^{k-1} dist(position(t_{i,k'}), position(t_{i,k'+1})) \right) + \sum_{k'=1}^{k} ptime(t_{i,k'}).$$

Let F be the set of finished tasks, i.e.

$$F = \{t_j \mid ctime(t_j) \le deadline(t_j)\}.$$

Let N be the number of finished tasks, i.e.

$$N = |F|.$$

Let R the total reward of finished tasks, i.e.

$$R = \sum_{t_j \in F} reward(t_j).$$

2.2. Optimization problems

In this section, we define our combinatorial optimisation problems.

We define two UAV mission scheduling problems.

The first problem is called *number of finished tasks maximisation* (NFTM). Given *m* heterogeneous UAVs and a list of tasks, the NFTM problem is to maximise the number of finished tasks, such that the completion of each task does not exceed its time deadline, the total flight distance of a UAV does not exceed its maximum flight distance, and the total resource consumption of a UAV does not exceed its maximum resource consumption.

Problem 2.1: Number of Finished Tasks Maximization (NFTM)

> *Input*: m UAVs: $u_1, u_2, \ldots, u_m$, where $u_i = (position(u_i), speed(u_i), maxdistance(u_i), maxresource(u_i))$, and a list of tasks $L = (t_1, t_2, \ldots, t_n)$, where t_j=$(position(t_j), ptime(t_j), deadline(t_j), request(t_j))$.
> *Output*: $route(u_i)$ for all i, such that N is maximised and $ctime(t_j) \le deadline(t_j)$ for all j, $distance(u_i) \le maxdistance(u_i)$ for all i, and $resource(u_i) \le maxdistance(u_i)$ for all i.

We would like to mention that the NFTM problem is NP-hard even if there is no distance and resource consideration, i.e. $maxdistance(u_i) = \infty$ and $maxresource(u_i) = \infty$ for all i, and $request(t_j) = 0$ for all j, and all tasks have a common time deadline, i.e. $deadline(t_j) = D$ for all j (K. Li 2023).

The second problem is called *reward of finished tasks maximisation* (RFTM). Given *m* heterogeneous UAVs and a list of tasks, the RFTM problem is to maximise the total reward of finished tasks, such that the completion of each task does not exceed its time deadline, the total flight distance of a UAV does not exceed its maximum flight distance, and the total resource consumption of a UAV does not exceed its maximum resource consumption.

Problem 2.2: Reward of Finished Tasks Maximization (RFTM)

> *Input*: m UAVs: $u_1, u_2, \ldots, u_m$, where $u_i = (position(u_i), speed(u_i), maxdistance(u_i), maxresource(u_i))$, and a list of tasks $L = (t_1, t_2, \ldots, t_n)$, where t_j=$(position(t_j), ptime(t_j), deadline(t_j), request(t_j), reward(t_j))$.
> *Output*: $route(u_i)$ for all i, such that R is maximised and $ctime(t_j) \le deadline(t_j)$ for all j, $distance(u_i) \le maxdistance(u_i)$ for all i, and $resource(u_i) \le maxdistance(u_i)$ for all i.

It is clear that NFTM is a special case of RFTM (when all rewards are identical). Therefore, the RFTM problem is also NP-hard.

3. Heuristic algorithms

In this section, we develop our heuristic algorithms.

3.1. An algorithmic framework

In this section, we present an *algorithmic framework* for both NFTM and RFTM, such that our heuristic algorithms (four for NFTM and two for RFTM) can be presented in a unified way.

Algorithmic Framework

for (each u_i) **do**
  $route(u_i) \leftarrow$ an empty list;
  $time(u_i) \leftarrow 0$;
  $distance(u_i) \leftarrow 0$;
  $resource(u_i) \leftarrow 0$;
  $location(u_i) \leftarrow position(u_i)$;
end do;
for (each t_j) **do**
  calculate $best(t_j)$;
end do;
$N \leftarrow 0$;
$R \leftarrow 0$;
while (there is still task in L) **do**
  if ($best(t_j)$ is undefined for all t_j in L)
    break;
  find t_j such that $gain(best(t_j), t_j)$ is the (minimum for NFTM)/(maximum for RFTM);
  $u_i \leftarrow best(t_j)$;
  append t_j to $route(u_i)$;
  remove t_j from L;
  $N \leftarrow N + 1$;
  $R \leftarrow R + reward(t_j)$;
  $time(u_i) \leftarrow time(u_i) + ftime(u_i, t_j) + ptime(t_j)$;
  $distance(u_i) \leftarrow distance(u_i) + dist(location(u_i), position(t_j))$;
  $resource(u_i) \leftarrow resource(u_i) + request(t_j)$;
  $location(u_i) \leftarrow position(t_j)$;
  update $gain(u_i, t_j)$ and $best(t_j)$ for all t_j in L;
end do;
return N for NFTM or R for RFTM.

We define a condition:

$$\begin{aligned} feasible(u_i, t_j) = {} & (time(u_i) + ftime(u_i, t_j) + ptime(t_j) \le deadline(t_j)) \\ & \text{and } (distance(u_i) + dist(location(u_i), position(t_j)) \le maxdistance(u_i)) \\ & \text{and } (resource(u_i) + request(t_j) \le maxresource(u_i)), \end{aligned}$$

which means that based on its current situation, u_i can flight to t_j and process t_j, without violating any time deadline, flight distance, or resource consumption constraint.

Let $gain(u_i, t_j)$ be an evaluation function of u_i and t_j, only if $feasible(u_i, t_j)$ is true. The exact definition of $gain(u_i, t_j)$ depends on a specific algorithm.

We define $best(t_j)$ to be the u_i with the minimum/maximum $gain(u_i, t_j)$:

$$u_i = \text{argmin/argmax}\{gain(u_i, t_j)\}, \text{forall } u_i \text{ such that } feasible(u_i, t_j) = \text{true},$$

where, for NFTM, we choose the minimum, and for RFTM, we choose the maximum. Note that $best(t_j)$ is undefined, if there is no u_i such that $feasible(u_i, t_j)$ is true.

All our heuristic algorithms follow the same algorithmic framework. Lines (1)–(12) initialise the UAVs and the tasks. The main body of the algorithm is in lines (13)–(27). In each repetition of the while-loop, the t_j which has the minimum $gain(best(t_j), t_j)$ for NFTM or the maximum $gain(best(t_j), t_j)$ for RFTM is identified (line (16)), i.e. a greedy method is adopted. Task t_j is then assigned to $u_i = best(t_j)$ (lines (17)–(21)), and the status of u_i is updated (lines (22)–(25)). All remaining tasks also update their status (line (26)). The while-loop is repeated until there is no more task to schedule (line (13)) or no task can be scheduled anymore due to time deadline, flight distance, and resource consumption constraints (lines (14)–(15)).

It is clear that the most time-consuming step is line (26), which takes $O(mn)$ time. Since the while-loop can be repeated n times, the overall time complexity of the algorithm is $O(mn^2)$.

3.2. *Algorithms for NFTM*

In this section, we present four heuristic algorithms for the NFTM problem. Each algorithm has its own $gain(u_i, t_j)$.

- Algorithm 1: Earliest Deadline First (EDF)

$$gain(u_i, t_j) = (deadline(t_j), dist(location(u_i), position(t_j)) \times request(t_j)).$$

- Algorithm 2: Shortest Distance First (SDF)

$$gain(u_i, t_j) = (dist(location(u_i), position(t_j)), deadline(t_j) \times request(t_j)).$$

- Algorithm 3: Least Request First (LQF)

$$gain(u_i, t_j) = (request(t_j), deadline(t_j) \times dist(location(u_i), position(t_j))).$$

- Algorithm 4: EDF-SDF-LQF

$$gain(u_i, t_j) = (deadline(t_j) \times dist(location(u_i), position(t_j)) \times request(t_j), j).$$

Note that the result of $gain(u_i, t_j)$ is a pair of values to break ties. To compare a pair, we define $(u_1, u_2) < (v_1, v_2)$ if and only if $(u_1 < v_1)$, or, $(u_1 = v_1)$ and $(u_2 < v_2)$.

3.3. *Algorithms for RFTM*

In this section, we present two heuristic algorithms for the RFTM problem. Each algorithm has its own $gain(u_i, t_j)$.

- Algorithm 5: Highest Reward First (HRF)

$$gain(u_i, t_j) = (reward(t_j), 1/(deadline(t_j) \times dist(location(u_i), position(t_j)) \times request(t_j))).$$

- Algorithm 6: EDF-SDF-LQF-HRF

$$gain(u_i, t_j) = (reward(t_j)/(deadline(t_j) \times dist(location(u_i), position(t_j)) \\ \times request(t_j)), 1/j).$$

Note that Algorithms 5 and 6 become Algorithm 4 if all rewards are identical.

Similarly, to compare a pair, we define $(u_1, u_2) > (v_1, v_2)$ if and only if $(u_1 > v_1)$, or, $(u_1 = v_1)$ and $(u_2 > v_2)$.

4. Upper bounds

In this section, we derive upper bounds for optimal solutions.

4.1. An upper bound for NFTM

In this section, we derive an upper bound for the optimal solution N^* of the NFTM problem. We give three possible upper bounds and then take the minimum of them.

We define N_t to be the number of t_j's such that

$$\min_{1 \le i \le m} \{dist(position(u_i), position(t_j))/speed(u_i)\} + ptime(t_j) \le deadline(t_j),$$

where the left-hand side is the minimum possible completion time of t_j. It is clear that $N \le N_t$.

We define

$$Distance = \sum_{i=1}^{m} maxdistance(u_i).$$

Let $dist(t_j)$ be the minimum distance to reach t_j, i.e.

$$dist(t_j) = \min \left\{ \min_{1 \le i \le m} \{dist(position(u_i), position(t_j))\}, \min_{j' \ne j} \{dist(position(t_{j'}), position(t_j))\} \right\},$$

where only those u_i's with

$$dist(position(u_i), position(t_j))/speed(u_i) + ptime(t_j) \le deadline(t_j)$$

are considered. Assume that

$$dist(t_1) < dist(t_2) < \cdots < dist(t_n).$$

Let $N_d = k$, where k is the largest integer satisfying:

$$dist(t_1) + dist(t_2) + \cdots + dist(t_k) \le Distance.$$

It is clear that $N \le N_d$.

We define

$$Resource = \sum_{i=1}^{m} maxresource(u_i).$$

Assume that

$$request(t_1) < request(t_2) < \cdots < request(t_n),$$

where each t_j satisfies

$$\min_{1 \le i \le m} \{dist(position(u_i), position(t_j))/speed(u_i)\} + ptime(t_j) \le deadline(t_j).$$

Let $N_r = k$, where k is the largest integer satisfying:

$$request(t_1) + request(t_2) + \cdots + request(t_k) \le Resource.$$

It is clear that $N \le N_r$.

To summarise, we get an upper bound for the NFTM problem: $N^* \le N_{ub} = \min\{N_t, N_d, N_r\}$.

4.2. An upper bound for RFTM

In this section, we derive an upper bound for the optimal solution R^* of the RFTM problem. We give three possible upper bounds and then take the minimum of them.

Let $time(t_j)$ be the minimum time to reach t_j + the processing time of t_j:

$$time(t_j) = dist(t_j) \Big/ \max_{1 \le i \le m} \{speed(u_i)\} + ptime(t_j)\ .$$

We define

$$Time = \sum_{i=1}^{m} maxtime(u_i).$$

Assume that

$$reward(t_1)/time(t_1) > reward(t_2)/time(t_2) > \cdots > reward(t_n)/time(t_n),$$

where each t_j satisfies

$$\min_{1 \le i \le m} \{dist(position(u_i), position(t_j))/speed(u_i)\} + ptime(t_j) \le deadline(t_j).$$

k is the largest integer satisfying:

$$time(t_1) + time(t_2) + \cdots + time(t_k) \le Time.$$

Let

$$\begin{aligned} R_t &= reward(t_1) + reward(t_2) + \cdots + reward(t_k) \\ &\quad + (reward(t_{k+1})/time(t_{k+1}))(Time - (time(t_1) + time(t_2) + \cdots + time(t_k)). \end{aligned}$$

It is clear that $R \le R_t$.

Assume that

$$reward(t_1)/dist(t_1) > reward(t_2)/dist(t_2) > \cdots > reward(t_n)/dist(t_n),$$

where each t_j satisfies

$$\min_{1\le i\le m} \{dist(position(u_i), position(t_j))/speed(u_i)\} + ptime(t_j) \le deadline(t_j).$$

k is the largest integer satisfying:

$$dist(t_1) + dist(t_2) + \cdots + dist(t_k) \le Distance.$$

Let

$$\begin{aligned} R_d &= reward(t_1) + reward(t_2) + \cdots + reward(t_k) \\ &\quad + (reward(t_{k+1})/dist(t_{k+1}))(Distance - (dist(t_1) + dist(t_2) + \cdots + dist(t_k)). \end{aligned}$$

It is clear that $R \le R_d$.

Assume that

$$reward(t_1)/request(t_1) > reward(t_2)/request(t_2) > \cdots > reward(t_n)/request(t_n),$$

where each t_j satisfies

$$\min_{1\le i\le m} \{dist(position(u_i), position(t_j))/speed(u_i)\} + ptime(t_j) \le deadline(t_j).$$

k is the largest integer satisfying:

$$request(t_1) + request(t_2) + \cdots + request(t_k) \le Resource.$$

Let

$$\begin{aligned} R_r &= reward(t_1) + reward(t_2) + \cdots + reward(t_k) \\ &\quad + (reward(t_{k+1})/request(t_{k+1}))(Resource - (request(t_1) + request(t_2) \\ &\quad + \cdots + request(t_k)). \end{aligned}$$

It is clear that $R \le R_r$.

To summarise, we get an upper bound for the RFTM problem: $R^* \le R_{ub} = \min\{R_t, R_d, R_r\}$.

5. Experimental performance evaluation

In this section, we conduct an experimental performance evaluation for our heuristic algorithms.

Table 1. Simulation results of algorithm EDF (99% Confidence Interval = $\pm 0.46928\%$).

n	$\tau = 30$	$\tau = 50$	$\tau = 70$	$\tau = 90$
20	0.96879	0.96879	0.96879	0.96879
40	0.98300	0.98174	0.98313	0.98269
60	0.98621	0.98779	0.98669	0.98743
80	0.98874	0.98708	0.98900	0.98782
100	0.98894	0.98999	0.98904	0.98138
120	0.98874	0.98923	0.98095	0.91100
140	0.99018	0.98584	0.92321	0.80585
160	0.98890	0.95761	0.83408	0.71618
180	0.98612	0.89413	0.75084	0.64512
200	0.97099	0.82152	0.68318	0.58864

5.1. Parameter setting

Assume that a three-dimensional space $[-3000, 3000] \times [-3000, 3000] \times [0, 300]$ has coordinates measured in meters (m). A position is specified as (x, y, z). The distance between two positions $p_1 = (x_1, y_1, z_1)$ and $p_2 = (x_2, y_2, z_2)$ is

$$dist(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}.$$

We consider $m = 4$ UAVs with the following parameters. The $position(u_i)$'s are $(d, 0, 100)$, $(0, d, 100)$, $(-d, 0, 100)$, $(0, -d, 100)$, where $d = 2000$ m. The $speed(u_i)$'s are i.i.d. random variables uniformly distributed in the range $[20, 30]$ m/s. The $maxtime(u_i)$'s are i.i.d. random variables uniformly distributed in the range $[1, 2]$ hours, i.e. $[3600, 7200]$ seconds. The $maxdistance(u_i)$'s are i.i.d. random variables uniformly distributed in the range $[72, 216]$ km, i.e. $[72000, 216000]$ m. The $maxresource(u_i)$'s are i.i.d. random variables uniformly distributed in the range $[0.8, 1.2] \times 10.5 \times (n/m)$, where $10.5 = (1 + 20)/2$.

The number of tasks is $n = 20, 40, \ldots, 200$. The $position(t_j)$'s are i.i.d. random variables uniformly distributed in the space $[-3000, 3000] \times [-3000, 3000] \times [0, 300]$. The $ptime(t_j)$'s are i.i.d. random variables uniformly distributed in the range $[\tau, 2\tau]$, where $\tau = 30$, 50, 70, 90 s. The $deadline(t_j)$'s are i.i.d. random variables uniformly distributed in the range $[600, 6000]$ s, i.e. $[10, 100]$ min. The $request(t_j)$'s are i.i.d. random variables uniformly distributed in the set $\{1, 2, \ldots, 20\}$. The $reward(t_j)$'s are i.i.d. random variables uniformly distributed in the set $\{1, 2, \ldots, 10\}$.

5.2. Simulation results

In this section, we show our simulation results.

In Tables 1–6, we demonstrate our experimental data for the six algorithms respectively. In each table, for each combination of n and τ, we generate $M = 500$ random samples of input (i.e. m random UAVs and n random tasks), execute the corresponding algorithm, calculate the upper bound N_{ub} or R_{ub}, and record the ratio N/N_{ub} or R/R_{ub}. The average of the M ratios is shown in the table, together with the maximum 99% confidence interval of all the data in the table.

We have the following observations.

Table 2. Simulation results of algorithm SDF (99% Confidence Interval = ±0.56354%).

n	$\tau = 30$	$\tau = 50$	$\tau = 70$	$\tau = 90$
20	0.96427	0.96304	0.96122	0.95682
40	0.98077	0.97127	0.95773	0.94147
60	0.97686	0.95861	0.93333	0.90358
80	0.96766	0.93646	0.90027	0.86006
100	0.95532	0.91155	0.86334	0.81139
120	0.93995	0.88432	0.82392	0.76838
140	0.92456	0.85614	0.79140	0.72371
160	0.90604	0.83067	0.75564	0.68581
180	0.89036	0.80217	0.72295	0.64858
200	0.87328	0.77614	0.69010	0.61395

Table 3. Simulation results of algorithm LQF (99% Confidence Interval = ±0.65958%).

n	$\tau = 30$	$\tau = 50$	$\tau = 70$	$\tau = 90$
20	0.94659	0.94899	0.94869	0.94594
40	0.97213	0.96499	0.95492	0.94350
60	0.96773	0.94442	0.91920	0.89714
80	0.94702	0.90696	0.87351	0.82865
100	0.91255	0.86458	0.81323	0.76968
120	0.88618	0.82171	0.76416	0.71238
140	0.85169	0.78143	0.72001	0.66233
160	0.81928	0.74586	0.67831	0.61639
180	0.79194	0.71003	0.63661	0.57572
200	0.75945	0.67618	0.59904	0.53799

Table 4. Simulation results of algorithm EDF-SDF-LQF (99% Confidence Interval = ±0.51538%).

n	$\tau = 30$	$\tau = 50$	$\tau = 70$	$\tau = 90$
20	0.95972	0.95982	0.95821	0.95772
40	0.98237	0.98140	0.97977	0.97578
60	0.98684	0.98371	0.97479	0.95891
80	0.98823	0.97656	0.95413	0.91940
100	0.98442	0.96106	0.91993	0.87167
120	0.97905	0.93774	0.88147	0.81223
140	0.96626	0.90575	0.83333	0.75970
160	0.95241	0.87554	0.79106	0.71064
180	0.93625	0.84094	0.74889	0.66408
200	0.91596	0.81025	0.70996	0.62270

- For the NFTM problem, when n and τ are small, Algorithm EDF has the best performance in the sense that it has the highest N/N_{ub} ratio. As n and τ increase, Algorithm EDF-SDF-LQF performs the best.
- For the RFTM problem, Algorithm EDF-SDF-LQF-HRF consistently has the best performance in the sense that it has the highest R/R_{ub} ratio.
- For all algorithms, as n and τ increase, the ratios N/N_{ub} and R/R_{ub} decrease, because the number of completed tasks decreases due to more tasks missing their time deadlines and insufficient flight distance and resource supplies.
- Since N and R are compared with N_{ub} and R_{ub} respectively, the actual performance ratios N/N^* or R/R^* should be higher than those in the tables.

Table 5. Simulation results of algorithm HRF (99% Confidence Interval = ±0.61934%).

n	$\tau = 30$	$\tau = 50$	$\tau = 70$	$\tau = 90$
20	0.98581	0.98542	0.98431	0.98157
40	0.99186	0.98774	0.97966	0.97129
60	0.98514	0.97326	0.95532	0.93622
80	0.97147	0.94784	0.92281	0.89399
100	0.95395	0.92390	0.88792	0.85624
120	0.93422	0.89484	0.85166	0.81520
140	0.91448	0.86560	0.82076	0.77901
160	0.89281	0.83922	0.79080	0.75529
180	0.87456	0.81260	0.76273	0.74092
200	0.85233	0.79053	0.74130	0.73689

Table 6. Simulation results of algorithm EDF-SDF-LQF-HRF (99% Confidence Interval = ±0.66539%).

n	$\tau = 30$	$\tau = 50$	$\tau = 70$	$\tau = 90$
20	0.96337	0.96337	0.96350	0.96547
40	0.98614	0.98675	0.98471	0.98190
60	0.99171	0.98761	0.98031	0.96884
80	0.99172	0.98093	0.96153	0.93703
100	0.98772	0.96713	0.93266	0.89437
120	0.98198	0.94844	0.90143	0.84890
140	0.97118	0.92401	0.86701	0.80946
160	0.95940	0.90033	0.83054	0.77652
180	0.94628	0.87335	0.80160	0.76125
200	0.93124	0.84747	0.77305	0.74723

6. Summary

For the first time in the literature, we have investigated optimal task scheduling for heterogeneous UAVs with completion time, flight distance, and resource consumption constraints within the framework of combinatorial optimisation. Our study has three unique features. First, we consider multiple resource and requirement constraints simultaneously. Second, we develop an algorithmic framework for all our heuristic algorithms. Third, we compare our heuristic solutions with optimal solutions.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Alhaqbani, A., Kurdi, H., & Youcef-Toumi, K. (2021). Fish-inspired task allocation algorithm for multiple unmanned aerial vehicles in search and rescue missions. *Remote Sensing*, *13*(1), 27. https://doi.org/10.3390/rs13010027

Aljalaud, F., & Kurdi, H. A. (2021). Autonomous task allocation for multi-UAV systems based on area-restricted search behavior in animals. *Procedia Computer Science*, *191*, 246–253. https://doi.org/10.1016/j.procs.2021.07.031

Bellingham, J., Tillerson, M., Richards, A., & How, J. P. (2003). Multi-task allocation and path planning for cooperating UAVs. In Butenko, S., Murphey, R., & Pardalos, P. M. (Eds.), *Cooperative control: Models, applications and algorithms* (pp. 23–41). Springer.

Bertuccelli, L., Choi, H.-L., Cho, P., & How, J. P. (2009). Real-time multi-UAV task assignment in dynamic and uncertain environments. *AIAA Guidance, Navigation, and Control Conference*, Chicago, Illinois.

Chen, J., & Sun, D. (2011). Resource constrained multirobot task allocation based on leader-follower coalition methodology. *The International Journal of Robotics Research*, *30*(12), 1423–1434. https://doi.org/10.1177/0278364910396552

Chen, X., Zhang, P., Du, G., & Li, F. (2019). A distributed method for dynamic multi-robot task allocation problems with critical time constraints. *Robotics and Autonomous Systems*, *118*, 31–46. https://doi.org/10.1016/j.robot.2019.04.012

Cui, W., Li, R., Feng, Y., & Yang, Y. (2022). Distributed task allocation for a multi-UAV system with time window constraints. *Drones*, *6*(9), 226. https://doi.org/10.3390/drones6090226

Filho, G., Passaro, A., Delfino, G., Santana, L., & Monsuur, H. (2022). Time-critical maritime UAV mission planning using a neural network: An operational view. *IEEE Access*, *10*, 111749–111758. https://doi.org/10.1109/ACCESS.2022.3215646

Fu, X., Feng, P., & Gao, X. (2019). Swarm UAVs task and resource dynamic assignment algorithm based on task sequence mechanism. *IEEE Access*, *7*, 41090–41100. https://doi.org/10.1109/Access.6287639

Geng, L., Zhang, Y. F., Wang, J., Fuh, J. Y. H., & Teo, S. H. (2014). Cooperative mission planning with multiple UAVs in realistic environments. *Unmanned Systems*, *2*(1), 73–86. https://doi.org/10.1142/S2301385014500058

Geng, N., Chen, Z., Nguyen, Q. A., & Gong, D. (2021). Particle swarm optimization algorithm for the optimization of rescue task allocation with uncertain time constraints. *Complex & Intelligent Systems*, *7*(2), 873–890. https://doi.org/10.1007/s40747-020-00252-2

Hu, J., & Yang, J. (2018). Application of distributed auction to multi-UAV task assignment in agriculture. *International Journal of Precision Agricultural Aviation*, *1*(1), 44–50.

Li, C.-I., Yen, L.-H., & Cho, M.-C. (2021). Distributed mission and charging scheduling for UAV swarm to maximize service coverage. *IEEE 94th Vehicular Technology Conference*, Norman, OK, USA.

Li, K. (2023). Heuristic task scheduling on heterogeneous UAVs: A combinatorial optimization approach. *Journal of Systems Architecture*, *140*, 102895. https://doi.org/10.1016/j.sysarc.2023.102895

Machovec, D., Siegel, H. J., Crowder, J. A., Pasricha, S., Maciejewski, A. A., & Friese, R. D. (2023). Surveillance mission scheduling with unmanned aerial vehicles in dynamic heterogeneous environments. *The Journal of Supercomputing*, *79*(12), 13864–13888. https://doi.org/10.1007/s11227-023-05225-z

Mao, X., Wu, G., & Fan, M. (2022). DL-DRL: A double-layer deep reinforcement learning approach for large-scale task scheduling of multi-UAV. arXiv:2208.02447. https://arxiv.org/abs/2208.02447

Ozkan, O. (2021). Optimization of the distance-constrained multi-based multi-UAV routing problem with simulated annealing and local search-based metaheuristic to detect forest fires: The case of Turkey. *Applied Soft Computing*, *113*(Part B), 108015. https://doi.org/10.1016/j.asoc.2021.108015

Peng, Q., Wu, H., & Xue, R. (2021). Review of dynamic task allocation methods for UAV swarms oriented to ground targets. *Complex System Modeling and Simulation*, *1*(3), 163–175. https://doi.org/10.23919/CSMS.2021.0022

Ramchurn, S. D., Fischer, J. E., Ikuno, Y., Wu, F., Flann, J., & Waldock, A. (2015). A study of human-agent collaboration for multi-UAV task allocation in dynamic environments. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence* (pp. 1184–1192). Palo Alto, CA: AAAI Press.

Scholarly Community Encyclopedia. (2023). Applications of unmanned aerial vehicles. https://encyclopedia.pub/entry/25512

Schumacher, C., Chandler, P., & Pachter, M. (2003). UAV task assignment with timing constraints (AFRL-VA-WP-TP-2003-315). Defense Technical Information Center.

Sebbane, Y. B. (2021). *Multi-UAV planning and task allocation*. CRC Press.

Sujit, P. B., Sinha, A., & Ghose, D. (2006). Multiple UAV task allocation using negotiation. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 471–478). New York, NY: Association for Computing Machinery.

Sullivan, N., Grainger, S., & Cazzolato, B. (2019). Sequential single-item auction improvements for heterogeneous multi-robot routing. *Robotics and Autonomous Systems*, *115*, 130–142. https://doi.org/10.1016/j.robot.2019.02.016

Sung, I., Danancier, K., Ruvio, D., Guillemet, A., & Nielsen, P. (2019). A design of a scheduling system for an unmanned aerial vehicle (UAV) deployment. *IFAC (International Federation of Automatic Control) PapersOnLine*, *52*(13), 1854–1859.

Tan, W., Hu, Y., Zhao, Y., Li, W., Zhang, X., & Li, Y. (2021). Mission planning for unmanned aerial vehicles based on Voronoi diagram-tabu genetic algorithm. *Wireless Communications and Mobile Computing*, *2021*, Article ID 4154787. https://doi.org/10.1155/2021/4154787.

Venugopalan, T. K., Subramanian, K., & Suresh, S.. (2015). Multi-UAV task allocation: A team-based approach. In *IEEE Symposium Series on Computational Intelligence* (pp. 45–50), Cape Town, South Africa, 7-10 December 2015.

Wang, Y., Bai, P., Liang, X., Wang, W., Zhang, J., & Fu, Q. (2019). Reconnaissance mission conducted by UAV swarms based on distributed PSO path planning algorithms. *IEEE Access*, *7*, 105086–105099. https://doi.org/10.1109/Access.6287639

Wang, Z., & Yan, X. (2021). Multi-UAV task assignment based on quantum genetic algorithm. *Journal of Physics: Conference Series*, *1824*(1), 012010.

Yan, M., Yuan, H., Xu, J., Yu, Y., & Jin, L. (2021). Task allocation and route planning of multiple UAVs in a marine environment based on an improved particle swarm optimization algorithm. *EURASIP Journal on Advances in Signal Processing*, *2021*(1), https://doi.org/10.1186/s13634-021-00804-9Article number: 94.

Yao, J., & Ansari, N. (2020). Online task allocation and flying control in fog-aided Internet of drones. *IEEE Transactions on Vehicular Technology*, *69*(5), 5562–5569. https://doi.org/10.1109/TVT.25

Yi, B., Lv, J., Wang, X., Chen, J., & Li, K. (2023). Digital twin constructed spatial structure for flexible and efficient task allocation of drones in mobile networks. *IEEE Journal on Selected Areas in Communications*, *41*(11), 3430–3443.

Yin, Y., Guo, Y., Su, Q., & Wang, Z. (2022). Task allocation of multiple unmanned aerial vehicles based on deep transfer reinforcement learning. *Drones*, *6*(8), 215. https://doi.org/10.3390/drones6080215

Zhang, X., & Chen, X. (2021). UAV task allocation based on clone selection algorithm. *Wireless Communications and Mobile Computing*, *2021*, Article ID 5518927. https://doi.org/10.1155/2021/5518927.

Zhou, X., Gao, F., Fang, X., & Lan, Z. (2021). Improved bat algorithm for UAV path planning in three-dimensional space. *IEEE Access*, *9*, 20100–20116. https://doi.org/10.1109/Access.6287639

Appendix

Table A1. Notations and definitions.

Notation	Definition
m	number of UAVs
n	number of tasks
u_i	a UAV
$position(u_i)$	initial location of u_i
$speed(u_i)$	flight speed of u_i
$maxdistance(u_i)$	the maximum flight distance of u_i
$maxresource(u_i)$	the maximum resource consumption of u_i
$maxtime(u_i)$	the maximum flight time of u_i
$route(u_i)$	flight route of u_i, $= (t_{i,1}, t_{i,2}, \ldots, t_{i,n_i})$, a sequence of tasks
$distance(u_i)$	total flight distance of u_i
$ftime(u_i)$	total flight time of u_i
$ptime(u_i)$	total processing time of u_i
$time(u_i)$	total time of u_i, i.e.(total flight time + total processing time) of u_i
$resource(u_i)$	total resource consumption of u_i
$location(u_i)$	current location of u_i
L	a list of tasks
t_j	a task
$position(t_j)$	position of t_j
$ptime(t_j)$	processing time of t_j
$deadline(t_j)$	time deadline to complete t_j
$request(t_j)$	amount of resource request of t_j
$reward(t_j)$	reward of completing t_j
$ctime(t_j)$	completion time of t_j
$dist(t_j)$	the minimum distance to reach t_j
$time(t_j)$	the minimum time to reach t_j + the processing time of t_j
$best(t_j)$	the u_i with the minimum/maximum $gain(u_i, t_j)$
$dist(p, q)$	distance between locations p and q
$ftime(u_i, t_j)$	flight time from u_i to t_j
F	the set of finished tasks
N	the number of finished tasks
N^*	the optimal solution of the NFTM problem
N_t, N_d, N_r, N_{ub}	upper bounds for N^*
R	the total reward of finished tasks
R^*	the optimal solution of the RFTM problem
R_t, R_d, R_r, R_{ub}	upper bounds for R^*
$Distance$	the sum of $maxdistance(u_i)$ for all u_i's
$Resource$	the sum of $maxresource(u_i)$ for all u_i's
$Time$	the sum of $maxtime(u_i)$ for all u_i's