International Journal of Industrial Engineering Computations 15 (2024) 255–276

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Heterogeneous multi-drone and helicopter routing problem for reconnaissance

Peixin Zhao[a], Xiaoyue Zeng[a*] and Chenchen Du[a]

[a]*School of Management, Shandong University, Jinan 250100, China*

C H R O N I C L E

Article history:
Received August 26 2023
Received in Revised Format September 12 2023
Accepted September 29 2023
Available online September 29 2023

Keywords:
Helicopter–drone
Orienteering problem
Adaptive simulated annealing

A B S T R A C T

Helicopters and drones are widely used in military and post-disaster reconnaissance. But less attention has been paid to collaborative reconnaissance between the two, especially when drones can be launched and retrieved multiple times. We propose a synchronous routing problem of helicopter and heterogeneous multi-drone for reconnaissance, which is a new variant of the orienteering problem (OP), where the drones can visit multiple mission nodes and can reconnoiter the retrieval nodes in a single trip, with the goal of maximizing the information collected. The problem is formulated as a mixed integer linear programming (MILP) model, and then an adaptive simulated annealing algorithm (A-SA) is designed to solve the problem. Specifically, a universal high-efficiency heuristics solution evaluation method based on segment sorting is proposed. The time complexity of this method is $O(n)$. The numerical experiments illustrate the accuracy and efficiency of the algorithm. The results also show that allowing the drones to conduct reconnaissance on the retrieval nodes can positively impact the solution.

1. Introduction

Helicopters and drones are modern high-tech aircraft that can perform a variety of reconnaissance missions in various situations. For instance, during post-disaster reconnaissance, they can rapidly arrive at the impacted area equipped with high-definition cameras and transmit real-time data.

The US Army validated the technology of the UH-60 Black Hawk helicopter to launch the ALTIUS-600 drone during the 2020 Project Fusion exercise (Mizokami, 2020), as shown in Fig. 1(a). Similarly, since 2020, Russia has installed drone launch capabilities in Mi-28NM helicopters. South Korea signed an agreement with Israel in the same year to jointly develop air-launched drones for LAH-armed helicopters. Recently, in October 2021, the Defense Advanced Research Projects Agency (DARPA) successfully retrieved an X-61 Gremlin drone from the air during testing, proving the technical feasibility of retrieving small drones in the air and returning them to their mother aircraft (Losey, 2021), as shown in Fig. 1(b). Using a helicopter as the mother aircraft for the aerial launch of drones has many advantages, including fewer takeoff and landing restrictions, fast maneuvering response, a wide range of use, and strong load-bearing capacity. This approach can effectively compensate for the shortcomings of drone endurance. Helicopters can carry a maximum payload of up to many tons, which allows them to carry multiple small drones. Moreover, the drone launch device is a tubular automatic system, allowing for the simultaneous launch of multiple drones. Therefore, helicopter-based deployment of multiple small drones for collaborative reconnaissance has excellent potential.

* Corresponding author
E-mail: xyzeng@mail.sdu.edu.cn (X. Zeng)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2024 Growing Science Ltd.
doi: 10.5267/j.ijiec.2023.9.011

(a) (b)

Fig. 1. (a) Launching drone; (b) Receiving drone

The collaborative reconnaissance approach using helicopters and drones presents various benefits. Firstly, helicopters enable drones to arrive at mission locations from a closer distance, thereby increasing the number of possible launch locations and extending the effective flight range. Secondly, pairing drones with helicopters during reconnaissance operations can expand the reconnaissance range beyond the helicopter's capabilities. Moreover, drones can be employed in hazardous or inaccessible areas for safe reconnaissance. Avi et al. (2022) also noted the advantages of combining drones and helicopters in emergency, medical, or search-and-rescue missions to enhance the situational awareness capability of helicopters.

Collaboration between helicopters and drones is similar to that between trucks and drones. However, the driving range of trucks is generally assumed to be infinite, whereas helicopters have limited range. Moreover, the motion routes of trucks and helicopters differ significantly. Trucks typically travel along highways or roads, and the road restricts their direction. In contrast, helicopters can move freely along a curved or straight route and are not limited by ground routes. Additionally, the collaboration between helicopters and drones is ideal for search and rescue, reconnaissance, and surveillance, whereas the collaboration between trucks and drones mainly applies to logistics.

We drew inspiration from the collaboration between trucks and drones to propose a new orienteering problem (OP) variant. Our variant involves solving the route planning and mission location allocation for collaborative reconnaissance of helicopters and drones. The objective is to gather as much information as possible in the mission area within a limited endurance. An example of a solution is illustrated in Fig. 2 for a problem with 10 mission nodes and 2 drones. Due to the rapid development of technology, drone models are being updated rapidly, making it possible for helicopters to carry multiple heterogeneous drones. Therefore, we refer to this problem as the heterogeneous drones–helicopter orienteering problem (HDHOP).

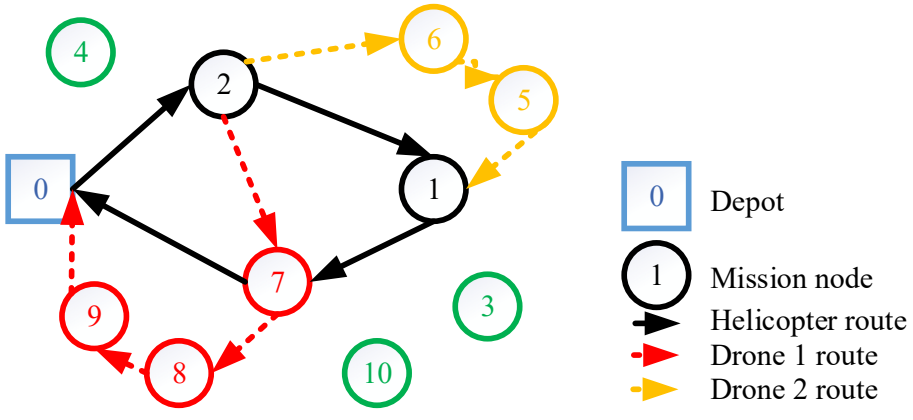

Fig. 2. An illustrative example of the HDHOP solution with ten mission nodes and two drones. The green represents nodes that are not reconnoitered, the black represents nodes that are reconnoitered by the helicopter, the red represents nodes that are reconnoitered by drone 1, and the yellow denotes nodes that are reconnoitered by drone 2

The contributions of this paper are summarized as follows:

- We conducted a study on a new collaborative orienteering problem that involves multiple heterogeneous drones and a helicopter. In this problem, a helicopter carries several small drones, and together the helicopter and drones perform

synchronous routing for reconnaissance. The drones can be launched and retrieved from the helicopter and can conduct reconnaissance in multiple mission locations within their endurance.

- We proposed that the drone retrieval nodes can be reconnoitered by drone. In this case, the drones can fully utilize the remaining battery power if they arrive at the retrieval node before the helicopter. Moreover, we enable the drones to travel directly from the launch nodes to the retrieval nodes without passing through other nodes, which can sometimes optimize the objective. Helicopter launch and retrieval are automated systems, thus we enable the launch and retrieval of drones to be carried out simultaneously with helicopter reconnaissance operations, providing greater flexibility in the timing of launch and retrieval.
- We developed an adaptive simulated annealing (A-SA) algorithm and introduced a new feasibility evaluation method based on segment sorting for the heuristic solution. Significantly, the proposed method has $O(n)$ time complexity.

This paper is structured as follows: Section 2 thoroughly reviews the pertinent literature. In Section 3, an elaborate introduction to the problem is presented, along with the development of a mathematical model. The A-SA algorithm and the segment-sorting evaluation method are outlined in Section 4. Section 5 describes the outcomes of numerical experiments. Finally, Section 6 outlines the conclusion and potential aspects for future research.

2. Literature review

To our knowledge, there has not yet been a direct study of the collaborative routing optimization problem between helicopters and drones. Our work relates to orienteering problems and problems with large aircraft launching small drones. In terms of solving methods, we mainly refer to the relevant techniques of vehicle–drone collaborative problems.

2.1 The orienteering problem

The orienteering problem (OP) is an optimization problem that originated from orienteering, a sport that involves racing and navigating in unfamiliar terrain. The problem has been addressed by several researchers (Golden et al., 1987) and consists in maximizing the total score of a set of nodes with prizes, subject to a constraint on travel costs between nodes. Although there is a vast body of literature on the OP, these foundational problems do not directly apply to the problem at hand. Interested readers are referred to recent surveys on the OP by Vansteenwegen et al. (2011) and Gunawan et al. (2016). While the underlying network of the HDHOP can be transformed into an OP-like problem, the multiple collaboration constraints in the HDHOP make it different from the classic OP. Therefore, the methods developed for the OP cannot be directly used to solve the HDHOP. In this study, we developed a special heuristic algorithm to address the unique characteristics of the HDHOP.

2.2 Problem with launching small drones from large aircraft

In recent years, there has been significant research on launching drones from trucks, but less attention has been paid to launching small drones from large aircraft. To the author's knowledge, only a few papers have focused on this topic. Poikonen and Golden (2020) studied the routing in the sky of airborne warehouses and drone releases to customers, while Wang et al. (2022) built upon this research and were the first to schedule the resupply of flying warehouses with drones from an earthbound depot via a shuttle. Unlike Poikonen and Golden (2020), Wang et al. (2022) assumed a fixed position for the airborne warehouse. Liu et al. (2020) researched the recovery of drone swarms by a mother aircraft and suggested that using "launch–recovery–relaunch" mode could significantly improve the efficiency of drone swarms. In a recent study, Avi et al. (2022) proposed a strategy to enhance the situational awareness of helicopters in emergency, medical, or search-and-rescue missions by employing them in collaboration with unmanned vertical take-off and landing drones. Meanwhile, Wen and Wu (2022) researched a new logistics delivery problem, utilizing a large drone to transport multiple small drones to distribution regions, then the small drones went to serve customers. The small drones could land at an automatic airport, distinguishing their study from previous ones.

2.3 Synchronization routing problem for a truck with multiple drones

The truck–drone routing problem was initially proposed by Murray and Chu (2015) as a flying sidekick traveling salesman problem (FSTSP) with a single truck–drone pair. Agatz et al. (2018) later presented a similar problem, called the traveling salesman problem with drones (TSP-D), which allows the drone to be retrieved by the truck at the same node where the drone was launched. Subsequently, several papers have extended the FSTSP and TSP-D. For a comprehensive review of truck–drone routing problems, readers may refer to Macrina et al. (2020), Chung et al. (2020), and Moshref-Javadi and Winkenbach (2021). Our problem is similar to the synchronization routing problem for a truck with multiple drones.

The traveling salesman problem with multiple drones (TSP-mD) was proposed by Yoon (2018) and Tu et al. (2018). The multiple flying sidekicks traveling salesman problem (mFSTSP) was proposed by Murray and Raj (2020) and Moshref-Javadi et al. (2020) based on Murray and Chu's (2015) FSTSP. In the mFSTSP, a delivery truck and a fleet of drones collaborate to deliver small packages to customers, and Murray and Raj (2020) compared several different models for flight endurance. Subsequently, many studies expanded the research based on a single truck with multiple drones, such as the multi-visit

problem (Luo et al., 2021), non-customer rendezvous locations (Mahmoudi and Eshghi, 2022; Salama and Srinivas, 2022), flexible time windows (Luo et al., 2022), uncertain navigation environments (Zhao et al., 2022), and arc retraversing (Morandi et al., 2023). Cavani et al. (2021) demonstrated an exact decomposition approach based on the compact MILP and a branch-and-cut algorithm that solved TSP-mD instances with up to 24 customers to prove optimality.

The vehicle routing problem with drones (VRPD) is a generalized form of the TSP-D (Wang et al., 2016). In the VRPD, a fleet of trucks collaborates with a fixed number of drones to serve customers. Subsequently, several VRPD variants with multiple drones per truck emerged, and various scenarios were introduced. For example, Wang and Sheu (2019) considered a service hub for storing drones, Kitjacharoenchai et al. (2019) allowed drones to be retrieved from any truck, Schermer et al. (2019) considered launching and retrieving drones at the same node, Kitjacharoenchai et al. (2020) allowed drones to serve multiple customers but prohibited launching or retrieving multiple drones at the same node, and Masmoudi et al. (2022) extended the VRP-D to include a flight of drones equipped with multiple package payload departments and considered customer time windows. Some extensions simplified the problem, while others made the problem more realistic. Tamke and Buscher (2021) derived the first branch-and-cut algorithm, which can optimally solve instances with up to 30 nodes.
All of the existing literature assumes that drone retrieval nodes are serviced by trucks.

2.4 Multiple drones per truck: evaluation of heuristics and solution feasibility

Compared to exact algorithms, well-designed heuristics can find near-optimal solutions for large-scale problems within a reasonable run time for NP-hard problems. Consequently, the research community has widely adopted heuristic approaches to solve the TSP-D, VRP-D, and their variants. Heuristic and metaheuristic techniques for the synchronization routing problem in the scenario of a single truck accompanied by multiple drones were provided by Moshref-Javadi et al. (2020), Murray and Raj (2020), Luo et al. (2021), Mahmoudi and Eshghi (2022), Luo et al. (2022), Salama and Srinivas (2022), and Zhao et al. (2022). The computational complexity of methods for evaluating solution feasibility has hardly been discussed in almost all of the research work in heuristic algorithms. There are three types of solution feasibility evaluation methods in the existing literature. The first method is to check if the changed route segment violates the energy constraints of the drone when generating a new neighborhood (Moshref-Javadi et al., 2020; Kitjacharoenchai et al., 2020; Salama and Srinivas, 2022; Masmoudi et al., 2022), which can be called a drone-level feasibility evaluation. However, this cannot be considered a solution-level feasibility evaluation (Luo et al., 2021) and could result in an infeasible solution being accepted. The second method uses mathematical programming models for evaluation (Murray and Raj, 2020; Mahmoudi and Eshghi, 2022), which takes a long time when calculating large-scale instances. The third method is to make an evaluation based on the characteristics of the problem, which requires a deep understanding of the problem. Only Luo et al. (2021) proposed a two-level solution evaluation method. The solution is presented as a directed acyclic graph, and a critical route method is invoked to perform solution-level feasibility evaluation in $O(n)$ time. However, there are limitations to this method, as it assumes that both the truck and the drone depart from the launch node simultaneously.

2.5 Summary

According to the literature reviewed in this section, four aspects warrant further investigation. Firstly, more attention has been paid to the collaborative operation between large aircraft and small drones, but studies are still needed that address the synchronous work between them. Secondly, in the problem of vehicle and drone collaboration, it has never been considered that the mission of the drone retrieval node is performed by the drone. Thirdly, more reliable and faster evaluation methods are needed to evaluate heuristic solution feasibility. Finally, the orientation problem of helicopter–drone collaboration has yet to be explored.

3. Problem description and mathematical formulation

This section provides a formal problem description and a MILP formulation of the HDHOP.

3.1 Problem description

The HDHOP considers a set of mission nodes, each with corresponding information. A helicopter and multiple drones with different speed and endurance limits will synchronously depart from a depot to collect information at mission nodes and then return to the depot. Each mission node can only be reconnoitered by a helicopter or a drone at most once. The helicopter and the drones can depart (or return) independently or fly in tandem. When flying in tandem, the drones are transported by the helicopter. The drones can fly multiple sorties, reconnoiter at least one mission node at a time, and directly travel from launch nodes to retrieval nodes. The drones can be launched from the helicopter at mission nodes or the depot. Once launched, the drones must return within the endurance limit to the depot or the helicopter at a mission node different from the launch nodes. If a drone or the helicopter arrives at the retrieval node early, it will hover until another aircraft arrives. The retrieval nodes of the drones can be reconnoitered by either the drones or the helicopter. The automatic flight control system controls the launch and retrieval of the drones, so the helicopter's launch and retrieval drone operations can be conducted simultaneously with its reconnaissance operations. The objective of HDHOP is to collect as much information as possible within the endurance limit.

The following conditions are assumed:

- In this study, the preparation time for drone launch and retrieval and the battery replacement time are considered to be zero, as these times can be ignored compared to the travel time from launch nodes to retrieval nodes. Similarly, we assume that the rate of energy consumption by the helicopter and the drones during takeoff, landing, and hovering is the same as it is during constant-speed flight.
- Multiple drones can be launched on one node, but due to technological limitations and flight security considerations, only one drone can be retrieved on one node.
- We assume that the helicopter and the drones fly at a constant speed, and the helicopter's speed is greater than the drones' speed. Their routes are measured using Euclidean distances. This measurement does not consider external factors such as wind speed and terrain.
- If a drone sortie ends at the depot, the drone cannot be re-launched from the depot.
- Each mission location is treated as a node to facilitate modeling and calculation.

3.2 Mathematical formulation

This section introduces the mathematical model for the described problem. We refer to Gonzalez et al.'s (2020) notations and model ideas. Firstly, we define the different sets, parameters, and variables needed to model the problem. Next, we present the constraints and group them based on role.

3.2.1. Notation

Table 1
Sets

$G = \{N, A\}$	Graph defining the set of mission locations or nodes to be reconnoitered and the set of directed links connecting them.
N	Set of nodes of graph G.
C	Set of mission locations that need to be reconnoitered.
o	The origin node of the mission, o ∈ N.
e	The ending node of the mission, e ∈ N.
A	Set of directed links in G.
$\delta^-(i)$	Set of nodes that can be used to arrive at node i ∈ N using links in A.
$\delta^+(i)$	Set of nodes that can be arrived from node i ∈ N using links in A.
K	Set of the drones.

Table 2
Parameters

T_{max}^H	The maximum level of helicopter endurance is expressed in time units.
Q_k	The maximum drone k ∈ K battery capacity (just when it is replaced) is expressed in time units.
s_j^H	Reconnaissance time of the helicopter at mission location j ∈ N, where s_o^H and s_e^H are equal to 0.
s_{jk}^D	Reconnaissance time of the drones at mission location j ∈ N, where s_o^D and s_e^D are equal to 0.
t_{ij}^H	The helicopter running time at the link (i, j) ∈ A.
t_{ijk}^D	The drone k running time at the link (i, j) ∈ A.
r_j	Information size of mission location j ∈ N, where r_o and r_e are equal to 0.
M	A big enough constant.

Table 3
Variables

$z_j^H \in \{0,1\}$	The binary variable equals one if the node j ∈ N is reconnoitered by the helicopter.
$z_j^D \in \{0,1\}$	The binary variable equals one if the node j ∈ N is reconnoitered by any drone.
$x_{ij} \in \{0,1\}$	The binary variable equals one if the link (i, j) ∈ A is traversed by the helicopter.
$y_{ij}^k \in \{0,1\}$	The binary variable equals one if the link (i, j) ∈ A is traversed by the drone k.
$t_j^L \geq 0$	The continuous variable measures the latest earliest departure time from node j ∈ N for the drones and the helicopter.
$0 \leq b_{kj}^- \leq Q_k$	The continuous variable that measures the drone k battery level when the drone is just coming to node j ∈ N.
$0 \leq b_{kj}^+ \leq Q_k$	The continuous variable that measures the drone k battery level when the drone is just departing from node j ∈ N.

3.2.2. Model formulation

We can formulate the HDHOP as the next MILP model:

Objective function

$$max \sum_{j\in N}(z_j^H + z_j^D)r_j \quad (1)$$

subject to the following:

(A) Helicopter route constraints:

$$\sum_{i\in\delta^-(j)} x_{ij} = \sum_{m\in\delta^+(j)} x_{jm} \le 1 \quad \forall j \in C \quad (2)$$

$$\sum_{j\in\delta^+(o)} x_{oj} = 1 \quad (3)$$

$$\sum_{i\in\delta^-(e)} x_{i,e} = 1 \quad (4)$$

(B) Drone route constraints:

$$\sum_{i\in\delta^-(j)} y_{ij}^k = \sum_{m\in\delta^+(j)} y_{jm}^k \le 1 \quad \forall j \in C, k \in K \quad (5)$$

$$\sum_{j\in\delta^+(o)} y_{oj}^k = 1 \quad \forall k \in K \quad (6)$$

$$\sum_{i\in\delta^-(e)} y_{i,e}^k = 1 \quad \forall k \in K \quad (7)$$

(C) Helicopter–drone route constraints:

$$\sum_{k\prime\in K\setminus\{k\},m\in\delta^-(j),m\neq i} y_{mj}^{k\prime} \le |K|(2 - x_{ij} + y_{ij}^k - \sum_{l\in\delta^-(j),i\neq l} y_{lj}^k) \quad (i,j)\in A, \forall k \in K, j \neq e \quad (8)$$

$$\sum_{k\prime\in K\setminus\{k\},i\in\delta^-(j)} y_{ij}^{k\prime} \le |K|(1 + \sum_{i\in\delta^-(j)} x_{ij} - \sum_{i\in\delta^-(j)} y_{ij}^k) \quad j \in C, \forall k \in K \quad (9)$$

(D) Synchronization constraints:

$$t_j^L \ge t_i^L + t_{ij}^H + s_j^H - M\left(2 - x_{ij} - z_j^H\right) \quad (i,j)\in A, \forall k \in K, \ j \neq e \quad (10)$$

$$t_j^L \ge t_i^L + t_{ij}^H - M\left(2 - x_{ij} - z_j^D\right) \quad (i,j)\in A, \forall k \in K, j \neq e \quad (11)$$

$$t_j^L \ge t_m^L + t_{mi}^H + t_{ijk}^D + s_{jk}^D - M\left(2 - y_{ij}^k + x_{ij} - x_{mi}\right) \quad (i,j)\in A, m \in \delta^-(i), \forall k \in K, j \neq e, m \neq j, i \neq o \quad (12)$$

$$t_j^L \ge t_o^L + t_{ojk}^D + s_{jk}^D - M(2 - y_{oj}^k + x_{oj} - \sum_{m\in\delta^+(o),m\neq j} x_{om}) \quad j \in C, \forall k \in K \quad (13)$$

$$t_j^L \le t_i^L + t_{ijk}^D + s_{jk}^D + M(2 - y_{ij}^k + \sum_{n\in\delta^-(j)} x_{nj} - \sum_{m\in\delta^+(i),m\neq j} x_{im}) \quad (i,j)\in A, \forall k \in K, j \neq e \quad (14)$$

$$t_j^L \ge t_i^L + t_{ijk}^D + s_{jk}^D - M(1 - y_{ij}^k + \sum_{m\in\delta^+(i)} x_{im} + \sum_{l\in\delta^-(j)} x_{lj}) \quad (i,j)\in A, \forall k \in K, j \neq e \quad (15)$$

$$t_j^L \le t_i^L + t_{ijk}^D + s_{jk}^D + M(1 - y_{ij}^k + \sum_{m\in\delta^+(i)} x_{im} + \sum_{l\in\delta^-(j)} x_{lj}) \quad (i,j)\in A, \forall k \in K, j \neq e \quad (16)$$

$$t_j^L \ge t_i^L + t_{ijk}^D + s_{jk}^D - M(3 - y_{ij}^k + x_{ij} - \sum_{m\in\delta^-(j),m\neq i} x_{mj} - z_j^D) \quad (i,j)\in A, \forall k \in K, \ j \neq e \quad (17)$$

$$t_j^L \ge t_i^L + t_{ijk}^D - M(3 - y_{ij}^k + x_{ij} - \sum_{m\in\delta^-(j),m\neq i} x_{mj} - z_j^H) \quad (i,j)\in A, \forall k \in K, \ j \neq e \quad (18)$$

$$t_e^L \ge t_i^L + t_{i,e}^H - M(1 - x_{i,e}) \quad i \in N\setminus e \quad (19)$$

$$t_e^L \le T_{max}^H \quad (20)$$

$$t_o^L = 0 \quad (21)$$

(E) Drone battery level constraints:

$$b_{kj}^- \ge Q_k - Q_k(2 - x_{ij} - y_{ij}^k) \quad (i,j)\in A, k \in K \quad (22)$$

$$b_{kj}^+ \ge Q_k - Q_k(2 - x_{ij} - y_{ij}^k) \quad (i,j)\in A, k \in K \quad (23)$$

$$b_{kj}^- \le b_{ki}^+ - t_{ijk}^D + M(1 - y_{ij}^k + x_{ij}) \quad (i,j)\in A, \forall k \in K \quad (24)$$

$$b_{kj}^{+} \le b_{kj}^{-} - s_{jk}^{D} + M\left(1 - y_{ij}^{k} + \sum_{l\in\delta^{-}(j)} x_{lj}\right) \quad (i,j)\in A, k\in K, j \ne e \tag{25}$$

$$b_{kj}^{+} \ge Q_k - Q_k\left(2 - y_{ij}^{k} + x_{ij} - \sum_{l\in\delta^{-}(j),l\ne i} x_{lj}\right) \quad (i,j)\in A, k\in K, j \ne e \tag{26}$$

$$b_{kj}^{-} \ge t_j^{L} - t_i^{L} - t_{ijk}^{D} - s_{jk}^{H} - M\left(2 - y_{ij}^{k} + x_{ij} - z_j^{H}\right) \quad (i,j)\in A, \forall k\in K \tag{27}$$

$$b_{kj}^{-} \ge s_{jk}^{D} + max\left\{0, t_j^{L} - t_i^{L} - t_{ijk}^{D} - s_{jk}^{D}\right\} - M\left(3 - y_{ij}^{k} + x_{ij} - \sum_{l\in\delta^{-}(j),l\ne i} x_{lj} - z_j^{D}\right) \quad (i,j)\in A, \forall k \in K \tag{28}$$

$$b_{ko}^{+} = Q_k \quad \forall k\in K \tag{29}$$

(F) Constraints on variable relationships:

$$z_j^{D} \ge \sum_{i\in\delta^{-}(j),k\in K} y_{ij}^{k} - \sum_{i\in\delta^{-}(j)} x_{ij} \quad \forall j\in C \tag{30}$$

$$z_j^{D} \le 1 - x_{ij} + \sum_{l\in\delta^{-}(j),k\in K,l\ne i} y_{lj}^{k} \quad (i,j)\in A, j \ne e \tag{31}$$

$$\sum_{i\in\delta^{-}(j),k\in K} y_{ij}^{k} \ge z_j^{D} \quad \forall j\in C \tag{32}$$

$$z_j^{H} \ge x_{ij} - \sum_{l\in\delta^{-}(j),k\in K,l\ne i} y_{lj}^{k} \quad (i,j)\in A, j \ne e \tag{33}$$

$$\sum_{i\in\delta^{-}(j)} x_{ij} \ge z_j^{H} \quad \forall j\in C \tag{34}$$

$$z_j^{H} + z_j^{D} \le 1 + \left(2 - y_{ij}^{k} + x_{ij} - \sum_{l\in\delta^{-}(j),l\ne i} x_{lj}\right) \quad (i,j)\in A, j \ne e \tag{35}$$

$$z_j^{H} + z_j^{D} \ge 1 - \left(2 - y_{ij}^{k} + x_{ij} - \sum_{l\in\delta^{-}(j),l\ne i} x_{lj}\right) (i,j)\in A, j \ne e \tag{36}$$

Constraints (2) ensure that a node is visited by the helicopter at most once and correspond to the balance between incoming and outgoing links for the helicopter route. Constraints (3) and (4) ensure that the number of outgoing links from node *o* and incoming links at node *e* equal 1. Constraints (5)–(7) are equivalent to Constraints (2)–(4) but apply to drones instead of the helicopter. Constraints (8) ensure that each retrieval node only retrieves one drone. Constraints (9) restrict reconnaissance of each mission node to at most one drone. Constraints (10) –(18) calculate the departure time of the helicopter and drones at each mission node. Constraints (19)–(20) guarantee that the helicopter returns to the depot within the flight endurance limit. Constraints (21) enforce the departure time at the origin node. The methodology proposed by Gonzalez et al. (2020) is utilized to compute the remaining battery capacity of drones when they arrive at and depart from nodes based on Constraints (22)–(26). Constraints (27) and (28) ensure the drones have sufficient battery capacity to be retrieved at the retrieval nodes. Constraints (29) set the initial battery capacity of drones in the depot to their maximum level. Constraints (30)–(36) indicate the relationship between binary variables.

4. Heuristic algorithm

This section presents a heuristic algorithm for solving the HDHOP. The simulated annealing (SA) and A-SA algorithms are introduced in Section 4.1. The solution's coding scheme is presented in Section 4.2, and the algorithm for constructing the initial solution is presented in Section 4.3. The operators used in the A-SA are defined in Section 4.4. Finally, Section 4.5 describes the efficient method of evaluating the feasibility of any solution.

To ensure consistency, we use the term "helicopter route" to describe the sequence of mission nodes visited by the helicopter. Additionally, we differentiate between a single drone trip and a drone schedule. A single drone trip includes a launch node, a sequence of visited mission nodes, and a retrieval node, while a drone schedule comprises a series of non-overlapping single trips assigned to the same drone. These definitions align with those proposed by Luo et al. (2021).

4.1 SA and A-SA algorithms

SA, a widely recognized probabilistic search technique developed by Kirkpatrick et al. (1983), can escape from local optimal solutions by probabilistically accepting suboptimal solutions. SA has been successfully applied to a variety of truck–drone routing problems, including those studied by Gonzalez et al. (2020), Moshref-Javadi et al. (2020), Masmoudi et al. (2022), and Salama and Srinivas (2022).

The A-SA algorithm, described as Algorithm 1, combines the ideas of the adaptive large neighborhood search (ALNS) and SA algorithms. Recent research by Masmoudi et al. (2022) also adopted this idea. The method is based on the simulated

annealing algorithm, which generates a series of solutions iteratively and utilizes the Metropolis criterion to accept some suboptimal solutions, enabling escape from local optima (Line 14). A-SA adjusts the operator weight based on performance during the search process (Line 17), dynamically selecting operators using roulette rules to enhance the efficiency and search space of the algorithm (Line 6). At each annealing, the new solution needs to undergo feasibility evaluation, and the objective value for infeasible solutions is 0 (Line 8). The initial temperature is set to T_{st} and the internal loop (Line 5) stops when the temperature is below T_0. The maximum number of iterations for the outer loop (Line 4) is $iterMax$, and the cooling coefficient is α.

Secondly, if there is no improvement with multiple consecutive iterations, it may indicate that the current operator weights are difficult to optimize for the current solution and need to be reset. Thus, if "noImpv2" consecutive iterations do not yield any improvements, the operator weights and scores will be reset (Line 27).

Finally, the algorithm will be restarted from other initial solutions to explore different search regions if consecutive "noImpv1" iterations are not improving (Line 24). This helps to enhance search region diversity and avoid excessive iteration around local optimal solutions. This technique is called the multi-start method in the literature. It has been successfully applied to various heuristic algorithms, such as multi-start Tabu search (Luo et al., 2021) and adaptive multi-start simulated annealing (Masmoudi et al., 2022).

Algorithm 1. A-SA

1: **Input:** $T_{st}, T_0, noImpvMax1, noImpvMax2, iterMax, \alpha$
2: $S \leftarrow$ Construct a new initial solution
3: $S_{best} \leftarrow S$, $noImpv1 \leftarrow 0$, $noImpv2 \leftarrow 0$, $T_c \leftarrow T_{st}$, $iter \leftarrow 0$, Initialize operator weights and scores
4: **While** *iter* < $iterMax$ **do**
5: **While** $T_c > T_0$ **do**
6: Select an operator q from all operators using a roulette wheel rule
7: $S_{new} \leftarrow q(S)$
8: Feasibility evaluation of S_{new}, if infeasible, $f(S_{new}) = 0$
9: **If** $f(S_{new}) \geq f(S)$ **then**
10: $S \leftarrow S_{new}$
11: **If** $f(S_{new}) \geq f(S_{best})$ **then**
12 $S_{best} \leftarrow S_{new}$
13: **End if**
14: **Else if** random (0,1) < exp ($\frac{f(S_{new})-f(S)}{T_c}$) **then**
15: $S \leftarrow S_{new}$
16: **End if**
17 Update the weight and score of operator q
18: $T_c \leftarrow \alpha T_c$
19: *noImpv1* $\leftarrow noImpv1 + 1$, *noImpv2* $\leftarrow noImpv2 + 1$
20: **If** S_{best} is improved **then**
21: *noImpv1* $\leftarrow 0$, *noImpv2* $\leftarrow 0$
22: **End if**
23: **If** *noImpv1* > $noImpvMax1$ **then**
24: *noImpv1* $\leftarrow 0$, $S \leftarrow$ construct a new initial solution
25: **End if**
26: **If** *noImpv2* > $noImpvMax2$ **then**
27: *noImpv2* $\leftarrow 0$, reset operator weights and scores
28: **End if**
29: *iter* ← *iter* + *1*, $T_c \leftarrow T_{st}$
30: **End while**
31: **End while**
32: **Output:** S_{best}

4.2 Solution coding scheme

A solution to the problem includes information about which mission nodes are assigned to the helicopter or the drones, the sequence of mission nodes visited by the helicopter and the drones, and the launch and retrieval nodes of all drone trips. The solution in Fig. 2 is represented in Fig. 3, where green squares indicate unvisited nodes and red squares indicate the drone reconnoiters retrieval node in a single trip when its value is 1. Otherwise, the red square value is 0.

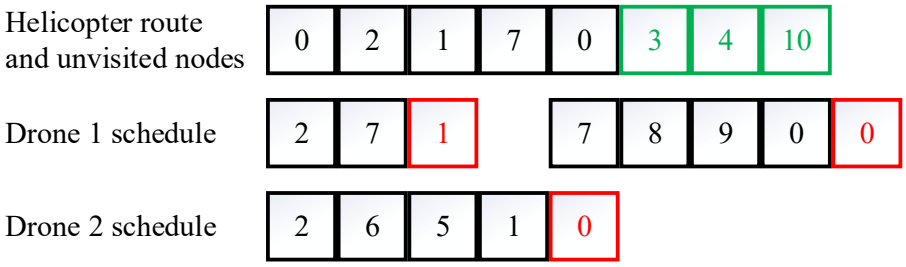

Fig. 3. Fig. 2 solution coding scheme

4.3 Initial solution construction

Algorithm 2. Construction algorithm

1: **Input:** N
2: $C \leftarrow N \setminus \{o, e\}$ is the unprocessed mission node sequence
3: Randomly select one node n from C and insert it in the middle of the route (o, e)
4: Remove n from C
5: Helicopter route $h \leftarrow (o, n, e)$
6: **While** not exceeding helicopter endurance **do**
7: **For** c in C **do**
8: Attempt to insert c at the optimal insertion position in h, and calculate information increase rate
9: **End for**
10: Update $h \leftarrow$ Select the route with the highest information increase rate after insertion
11: Remove c in h from C
12: **End while**
13: Let L_k be a set of launch nodes that drone k can use
14: Let R_k be a set of retrieval nodes that drone k can use
15: **While true do**
16: **For** each drone k **do**
17: $c \leftarrow$ randomly select a node from C
18: **For** each feasible pair of l_k and r_k where $l_k \in L_k$ and $r_k \in R_k$ **do**
19: Attempt to generate a single drone trip (l_k, c, r_k), and calculate information increase rate
20: **End for**
21: Retain $u \leftarrow (l_k, c, r_k)$ as the single drone trip with the highest rate of information increase for drone k
22: Remove c in u from C, remove l_k in u from L_k and remove r_k in u from R_k
23: **End for**
24: Until feasible single drone trips cannot be generated
25: **End while**
26: **While true do**
27: Randomly select a single drone trip for any drone k
28: **For** each node in C **do**
29: Attempt to insert at the optimal position in the single drone trip and calculate information increase rate
30: **End for**
31: Retain $u \leftarrow u$ as the single drone trip with the highest rate of information increase after insertion
32: Remove the selected insertion node from C
33: Until there is no feasible insertion
34: **End while**
35: **Output:** helicopter's initial route, drones' initial schedule

Based on the characteristics of the problem, we develop a construction algorithm (Algorithm 2) that guarantees the generation of feasible solutions. The helicopter route is created by randomly selecting the first insertion node to ensure solution diversity (Line 3). Then, we choose the insertion node with the highest rate of information increase (information gain per unit resource consumption) at the optimal insertion position (minimal change in travel time caused by insertion) (Line 10) sequentially until we reach the maximum endurance limit of the helicopter.

Secondly, each drone is selected by looping (Line 16) and one node is randomly chosen from the unvisited nodes (Line 17). Then, a set of single drone trips is generated using this selected node and each feasible pair of launch and retrieval nodes. Only the single drone trip with the highest information increase rate in the set is retained (Line 21). The above operation is repeated until no more feasible single drone trips can be generated.

Finally, a single drone trip is randomly selected (Line 27), and each unvisited node attempts to insert itself into the optimal

position of the single drone trip (Line 29), generating a set of single drone trips visiting an additional node. Only the single drone trip with the highest information increase rate in the set is retained (Line 31). The above operation is repeated until there is no feasible insertion.

To make the generated solutions feasible, we ensure there will be no situation where the helicopter is waiting for the drones. Specifically, the above sets of single drone trips must meet the following conditions, and the travel time does not include waiting time:

- The drones' travel time must not exceed their maximum endurance.
- The drones' travel time must not exceed the corresponding helicopter's travel time (from the launch node of the single drone trip to the retrieval node).
- The corresponding helicopter's travel time must not exceed the drones' maximum endurance.

4.4 Operators for A-SA

Heuristic methods that utilize insertion and swap operators are commonly used to solve the orienteering problem (Kim et al., 2020). Thus, these operators were extended for use in the HDHOP, considering the problem's unique characteristics. A total of eight operators are proposed. Note that the nodes eligible for swap or insertion in a single drone trip exclude the launch and retrieval nodes. The helicopter route nodes, including the launch and retrieval nodes, are eligible for swap or insertion. Furthermore, the launch and retrieval nodes of the affected single drone trip must be updated after swap or insertion.

4.4.1 Insertion operators

- For all single drone trips, when the drones arrive at a retrieval node, if any drone's remaining endurance is sufficient for the reconnaissance mission, then the retrieval node reconnaissance mission is transferred from the helicopter to the drone(s). Then, randomly select a drone to generate a trip from the launch node to the retrieval node and conduct reconnaissance. Finally, select the unvisited node with the highest rate of information increase after insertion at the optimal location on the helicopter route, and insert the node into the helicopter route.
- Select the single drone trip with the highest remaining endurance upon arrival at the retrieval node. If the drone reconnoiters the retrieval node on this trip, change the retrieval node to be reconnoitered by the helicopter. Among the unvisited nodes, select the node with the highest information increase rate after being inserted at the optimal position on this single drone trip, and insert the node in the trip.
- Randomly select a single drone trip. If this trip goes directly from the launch node to the retrieval node, select the unvisited node with the highest information increase rate after being inserted at the optimal position in this trip, and insert it into this trip. If this trip does not go directly from the launch node to the retrieval node, randomly remove a node from this trip, select the unvisited node with the highest information increase rate after insertion at the optimal position in this trip, and insert it into this trip.
- Randomly select a single drone trip. If the drone reconnoiters the retrieval node, change it to be reconnoitered by the helicopter. Combine the helicopter route nodes and unvisited nodes into a single set, then randomly select a node from this set to insert at the optimal position of this drone trip. If the insertion node is a launch or retrieval node in any drone trip, set the subsequent node of the insertion node in the helicopter route as the launch node for the affected single drone trip and the previous node as the retrieval node for the affected single drone trip (Luo et al., 2021).
- Randomly select a node from the unvisited nodes, then randomly select a drone with a feasible launch and retrieval node pair to create a single drone trip for this selected node.

4.4.2 Swap operators

- Combine nodes of the helicopter route and unvisited nodes into a set, then randomly select two nodes to swap positions; at least one of them should be a node on the helicopter route. If these swapped nodes are launch or retrieval nodes in any drone trip, update the launch and retrieval nodes of the affected drone trip to the node after the swap.
- Combine nodes of the helicopter route and unvisited nodes into a set and randomly select a node. Then randomly select a single drone trip, which must not be a direct trip from the launch node to the retrieval node. Swap the two selected nodes. If the swapped node is a launch or retrieval node in the other drone trip, update the launch and retrieval nodes of the affected drone trip to the node after the swap.
- Randomly select a single drone trip, which must not be a direct trip from the launch node to the retrieval node. Swap nodes between any two positions in this trip, or swap any node in this trip with an unvisited node. If swapping with an unvisited node, select the swapped node with the highest rate of information increase after the swap among the unvisited nodes.

4.5 Solution feasibility evaluation

The feasibility of any solution depends on its structure and whether energy consumption is within the maximum endurance level. Since the drones' launch and retrieval operations can be done simultaneously with the helicopter's reconnaissance operations, the launch and retrieval timing in single drone trips is not fixed. When evaluating the feasibility of the endurance level, a solution is feasible when the minimum waiting time for the helicopter and the drones is feasible. Algorithm 3 is used to evaluate the feasibility of the solutions. The objective value is set to 0 if the solution is not feasible. Section 4.5.1 explains the feasibility evaluation of the solution's structure, Section 4.5.2 discusses the calculation of the minimum waiting time, and Section 4.5.3 describes the feasibility evaluation of the solution's endurance level.

Algorithm 3. Solution feasibility-evaluation

Input: S_{new}
If S_{new} has infeasible structure **then**
    $f(S_{new}) = 0$
Else if S^{new} is infeasible at endurance-level **then**
    $f(S_{new}) = 0$
End if
Output: $f(S_{new})$

4.5.1 Structural feasibility evaluation

A solution is feasible at the structure level only if two conditions are met: (1) there is no scheduling for each drone, as shown in Fig. 4; and (2) the helicopter can retrieve only one drone at each node except for the depot.

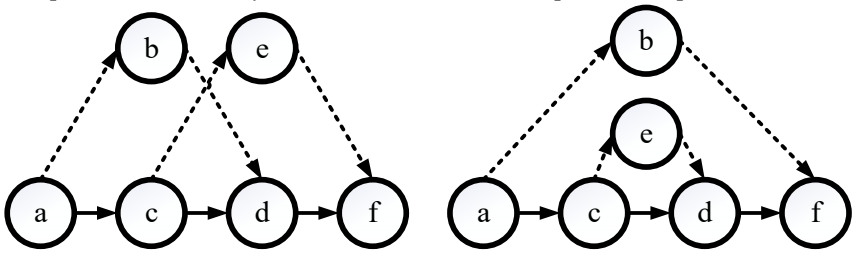

Fig. 4. Two prohibited drone schedules (Ponza, 2016)

4.5.2 Calculate the minimum waiting time

The combined drone and helicopter routes between the same drone launch nodes and their corresponding retrieval nodes is defined as a route segment. *S* is defined as the length of the drones' available launch time window at the mission nodes. The earliest launch times are when the drones are retrieved by the helicopter or when the helicopter and the drones arrive simultaneously at the nodes. The drones' latest launch times are when the helicopter leaves the nodes' timing. In Fig. 5, *a* indicates the time when the helicopter arrives at the node, and *d* is the time when the helicopter leaves the node. The total time span is the helicopter's reconnaissance time plus its waiting time at this node. When a node is reconnoitered by a drone, the reconnaissance time of the helicopter at the node is equal to 0. When there is no helicopter waiting at the node, the waiting time of the helicopter is also 0. Three scenarios for the launch time window of a drone at any node are as follows:

- When the helicopter and the drones arrive at the node simultaneously, or when a drone arrives first (shown in Fig. 5(a);
- When the helicopter arrives first, and the drones arrive before the helicopter completes the reconnaissance of the node (shown in Fig. 5(b);
- When the helicopter is waiting for a drone at a node (shown in Fig. 5(c).

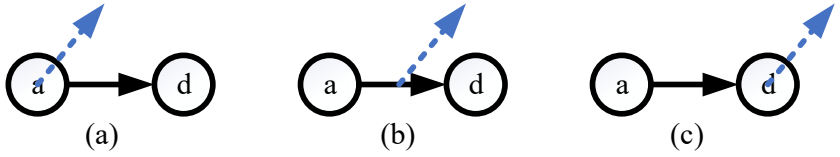

Fig. 5. Launch time window

Table 4
Route segment related parameters

TH	The travel time for the helicopter from leaving the launch node to reach the retrieval node.
TD	The travel time for the drone from leaving the launch node to reach the retrieval node.
SD	The time required for the drone to conduct reconnaissance for the retrieval node.
SH	The time required for the helicopter to conduct reconnaissance for the retrieval node.
WH	The minimum waiting time for the helicopter at the retrieval node.
WD	The minimum waiting time for the drone at the retrieval node.

In a route segment, if a drone is launched at the earliest time at the launch node and the helicopter still needs to wait at the retrieval node, then this waiting time for the helicopter is the minimum waiting time, which is calculated by Formulas (37) and (38). This also indicates that if the drone is launched any time before the helicopter leaves the launch node, the helicopter must wait at the retrieval node.

If the retrieval node is reconnoitered by the helicopter:

$$WH = TD - TH - SH - S \tag{37}$$

If the retrieval node is reconnoitered by the drone:

$$WH = TD + SD - TH - S \tag{38}$$

In a route segment, if the drone is launched at the launch node at the latest time and still needs to wait at the retrieval node, then this waiting time is the minimum waiting time, which is calculated by Formulas (39) and (40). This also indicates that if the drone is launched any time before the helicopter leaves the launch node, the drone must wait at the retrieval node.

If the retrieval node is reconnoitered by the helicopter:

$$WD = TH - TD \tag{39}$$

If the retrieval node is reconnoitered by the drone:

$$WD = TH - TD - SD \tag{40}$$

4.5.3 Endurance-level feasibility evaluation

The solutions are feasible at the endurance level only if the total energy consumption of each drone and the helicopter during flight, reconnaissance, and waiting does not exceed the maximum endurance level. The algorithm is detailed as Algorithm 4.

Algorithm 4. Endurance-level feasibility evaluation

Input: S_{new}
Sort the route segments according to the order of retrieval nodes on the helicopter route
For each route segment after sorting **do**
    Calculate the waiting time of the helicopter or drones at the retrieval node and update the launch time window accordingly
    If the single drone trip energy consumption exceeds its maximum endurance level **then**
        $f(S_{new}) = 0$
    End if
End for
Calculate helicopter route energy consumption
If the energy consumption of the helicopter route exceeds its maximum endurance level **then**
    $f(S_{new}) = 0$
End if
Output: $f(S_{new})$

Helicopter and drone energy consumption affect each other between route segments, and this influence is transmitted according to the order of the retrieval nodes on the helicopter route. Therefore, we can evaluate each route segment according to the order of the retrieval nodes on the helicopter route and update the launch time window and the helicopter or drone waiting time of each route segment retrieval node in turn. We call this a segment-sorting evaluation (SSE). An example is shown in Fig. 6, where route segment 1 impacts the earliest drone launch time at node 3 and the helicopter travel time of route segments 2 and 4. Route segment 2 affects the helicopter travel time of route segments 3 and 4, and route segment 3 affects the helicopter travel time of route segment 4. Thus, the sequence of evaluation should be 1–2–3–4.

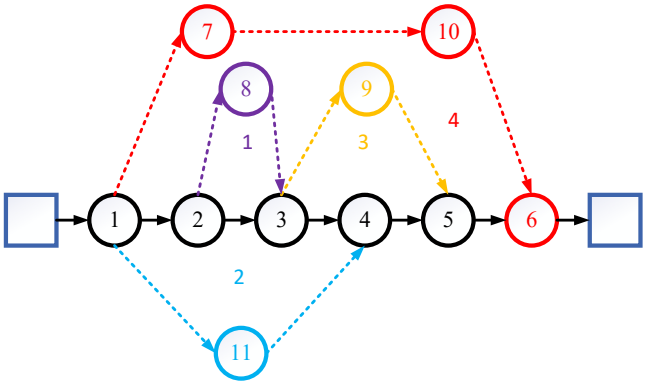

Fig. 6. Example of evaluation sequence

The calculation time of the SSE is directly proportional to the number of route segments with an upper limit of n/2. Thus, the time complexity of SSE is $O(n)$. Furthermore, this method can be extended to address the problem of retrieving multiple drones at the same retrieval nodes. However, it needs to determine the time window of the retrieval nodes.

5. Numerical examples

This section presents the calculation results of the HDHOP test instances. First, the test instances are described in Section 5.1. Second, the algorithm's parameters are determined through parameter tuning (Section 5.2). Next, an experiment is conducted to verify that the proposed SSE has a faster evaluation speed than the method based on mathematical models (Section 5.3). Fourth, the performance of the A-SA algorithm is evaluated on small-scale instances by comparing it with the Gurobi solver (Section 5.4). Fifth, large-scale instance experiments are conducted (Section 5.5), including a comparison with the SA algorithm. Finally, the effects of allowing the drones to reconnaissance at retrieval nodes are analyzed (Section 5.6). The experimental method in this section refers to the numerical experimental method proposed by Gu et al. (2022) and Matijević (2023).

The MILP model was solved using Gurobi 10.0.0, and the A-SA algorithm was implemented using Python 3.9.13. All computational work was conducted on a Lenovo PC with a 4-core Intel i5-1135G7 processor and 16 GB of RAM, running Microsoft Windows 10 in 64-bit mode. A stopping criterion was defined based on the maximum number of iterations. The maximum number of iterations for each instance, denoted as iterMax, was determined according to the number of nodes N: 10 iterations when $N < 100$; 20 iterations when $100 \leq N$.

5.1 HDHOP test instances

These instances were modified based on benchmark instances (kroA100, kroA150, kroA200, kroB100, kroB150, kroB200, lin105, lin318 in gen2, gen3, gen4) proposed by Kobeaga et al. (2018) for the orienteering problem. The reconnaissance time for each mission node by the helicopter is limited to less than 5 minutes and is generated using random number seeds. The reconnaissance time for the drones at each mission node is proportional to the helicopter's time, and the proportional coefficient is the ratio of the helicopter's speed to the drone's speed. All instances cannot obtain feasible solutions to visit all nodes through A-SA, to ensure consistent difficulty.

5.2 Parameter setting

In our numerical experiment, the helicopter carried three distinct models of drones. For detailed data, please consult Table 5.

Table 5
Helicopter and drone flight parameters

Vehicle	Speed	Endurance
Helicopter	350 km/h	6 h
Drone1	100 km/h	2 h
Drone2	90 km/h	4 h
Drone3	80 km/h	5 h

In the A-SA algorithm, the initial temperature (T_{st}) is set to 100 and the termination temperature (T_0) is set to 10. The operators' initial weights and scores are set to 1. The number of consecutive unimproved times (*noImpvMax2*) is set to one-fifth of *iterMax*, and *noImpvMax1* is set to one-tenth of *iterMax*. The cooling rate (α) is determined to be 0.99975, as suggested by Demir et al. (2012) and Masmoudi et al. (2016). Operator scores are updated according to the rules outlined in Table 6.

Table 6
Operators score updates criteria

State	Score
$f(S_{new}) > f(S)$ and $f(S_{new}) > f(S_{best})$	1.5
$f(S_{new}) > f(S)$ and $f(S_{new}) \leq f(S_{best})$	1.2
$f(S_{new}) = f(S)$ and $f(S_{new}) = f(S_{best})$	0.7
$f(S_{new}) = f(S)$ and $f(S_{new}) \neq f(S_{best})$	0.5
$f(S_{new}) < f(S)$ and meets Metropolis	0.3
$f(S_{new}) < f(S)$ and does not meet Metropolis	0.1

We adopted the weight update method described by Ropke and Pisinger (2006), using a reaction factor β. To examine the influence of varying β values, we chose the lin105 instance from gen2, gen3, and gen4. Then, we selected N nodes sequentially, starting from the Gth node in lin105, to generate 18 new instances, where N took values of 10, 25, and 50. The newly generated test instances were named *lin105-genX-N-G*. The results are presented in Table 7, with column S_a indicating the average objective value and column T_a indicating the average calculation time in seconds. Our observations suggest that reaction

factor β significantly influences the solutions. For $\beta = 0.3$, the calculation time was relatively short, but the obtained objective value was lower. The best solution was obtained using $\beta = 0.9$, with the highest objective value and a moderate increase in calculation time. Hence, we set reaction factor β to 0.9 for subsequent experiments. The detailed results for each instance are provided in Appendix B Table B1.

Table 7
Calibration experiment results for strength of reaction factor

N	β =0.1		β =0.3		β =0.5		β =0.7		β =0.9	
	S_a	T_a	S_a	T_a	S_a	T_a	S_a	T_a	S_a	T_a
10	187.17	26.05	187.17	27.45	187.17	26.61	187.17	28.22	187.17	28.16
25	496.83	20.59	502.00	19.85	510.17	20.38	512.17	21.83	512.17	21.48
50	635.67	66.18	626.67	56.72	633.17	62.67	635.83	60.56	639.33	60.36
Average	439.89	37.61	438.61	34.67	443.50	36.55	445.06	36.87	446.22	36.67

5.3 Acceleration analysis of SSE

This section presents an experiment that was conducted to evaluate the performance of SSE. The critical path method (CPM) proposed by Luo et al. (2021) assumes that vehicles and drones must depart simultaneously, making it unsuitable to evaluate the feasibility of HDHOP solutions. We customized a linear programming (LP) model for HDHOP to evaluate the feasibility of the solutions. The constraint conditions for LP are from Constraints (10)–(29) in Section 3.2.2. We compared the evaluation efficiency of SSE and LP. Please note that the time complexity of SSE is $O(n)$, which is a linear relationship with the instance size.

Using the A-SA algorithm, we applied these two evaluation methods and recorded the computational time required to evaluate the feasibility of a single solution. The test instances were modified from kroA200 in gen2. The size of N ranged from 10 to 160, resulting in a total of 16 instances. We conducted 50 runs for each instance and calculated the average evaluation time for each run. The recorded time was measured in milliseconds.

We plotted the relationship between the SEE and LP average calculation time and the number of nodes (N) (Fig. 7). The results reveal that LP required a longer calculation time than SEE for feasibility evaluation. With an increasing number of nodes, LP exhibited a strong growth trend for calculation time, increasing from 3.2504 to 49953.6652 milliseconds. In contrast, SEE's calculation time showed a slight increase from 0 to 0.8360 milliseconds. These findings demonstrate the superiority of SEE over LP, particularly for large-scale test instances. Fig. 8 shows that the calculation time of SEE increased approximately linearly, supporting the statement about the time complexity of SEE. The results are summarized in Appendix B Table B2.

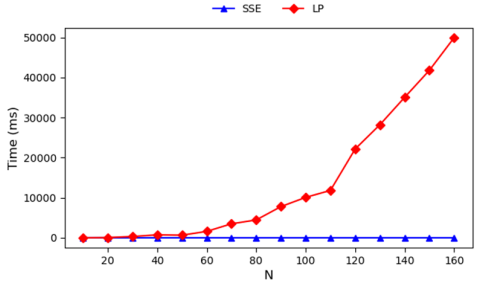

Fig. 7. Comparison of LP and SSE calculation time

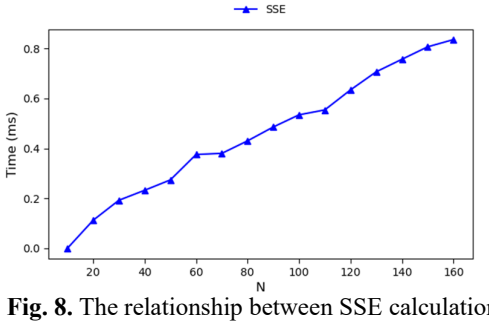

Fig. 8. The relationship between SSE calculation time and N

5.4 Comparison of HDHOP small-scale instances

To evaluate the performance of the A-SA, we chose lin105 instances from gen2, gen3, and gen4 and sequentially selected N nodes beginning from the Gth node in lin105 to create 15 small-scale instances. These instances were solved using Gurobi 10.0.0 based on the mathematical models proposed in Section 3.2.2. Some acceleration constraints were applied to make the model more compact and improve the solving speed of Gurobi, summarized in Appendix A. Based on a running time limit of 1800 seconds, the number of nodes in the optimal solution for the scenario of three drones was nine ($N = 9$). We selected eight- and nine-node instances for the experiment based on this result.

The summarized results are presented in Table 8, which includes the following notations: # *Opt* represents the instance numbers in optimal solutions found by Gurobi, and $\#S_b$ indicates the number of instances where A-SA achieved the same objective value as Gurobi. T_{tot} denotes the average time the Gurobi solver takes, while T_b represents the average time required by the A-SA algorithm to achieve the same objective value.

For the instances involving eight nodes, A-SA successfully obtained the optimal solutions for all test instances within approximately 2 seconds, whereas Gurobi took a minimum of 147 seconds to achieve the same result. Regarding the instances with nine nodes, Gurobi found the optimal solutions in only three instances. In contrast, A-SA found solutions with equivalent objective values to Gurobi and accomplished this within a significantly shorter time. Thus, A-SA demonstrates superior performance. Please refer to Appendix B Table B3 for comprehensive details on each test instance.

Table 8
Comparison of Gurobi and A-SA for small Instances

		N=8				*N*=9			
		Gurobi		A-SA		Gurobi		A-SA	
Inst. series	Inst. num	*#Opt*	T_{tot}	$\#S_b$	T_b	*#Opt*	T_{tot}	$\#S_b$	T_b
lin105-gen2	5	5	193.02	5	1.75	1	1674.09	5	1.67
lin105-gen3	5	5	147.38	5	1.33	1	1720.99	5	1.78
lin105-gen4	5	5	178.97	5	1.21	1	1690.79	5	1.30

5.5 Comparison of HDHOP large-scale instances

To evaluate the effectiveness of the proposed A-SA algorithm in large-scale scenarios, we compared its performance with the simulated annealing (SA) algorithm, employing the same parameter settings for both algorithms. We generated 54 large-scale instances based on the kroA100, kroB100, kroA150, kroB150, kroA200, kroB200, and lin318 instances from gen2, gen3, and gen4. The detailed results of the experiments conducted on 100, 150, and 200 nodes, including from 10 runs, the best (S_b) and average (S_{avg}) objective values, best (T_b) and average (T_{avg}) operation times, and average gap (Gap_{avg}) between the best and average objective values, are summarized in Table 9. The results demonstrate that the A-SA algorithm outperformed the SA algorithm regarding the objective value, solution speed, and stability. Additionally, the A-SA algorithm exhibited remarkable computational efficiency. For the three-drone scenario, the average time to obtain the superior solution was approximately 151.41 seconds for $N = 100$, less than 170 seconds for $N = 150$, and around 335.81 seconds for $N = 200$. Detailed information on each test instance is given in Appendix B Table B4.

Table 9
Comparing A-SA and SA: summarized results for large instances

	N=100					*N*=150					*N*=200				
	S_b	T_b	S_{avg}	T_{avg}	Gap_{avg}	S_b	T_b	S_{avg}	T_{avg}	Gap_{avg}	S_b	T_b	S_{avg}	T_{avg}	Gap_{avg}
A-SA	4495.44	127.42	4295.31	151.41	0.04	5513.72	136.81	5173.96	168.66	0.06	6014.56	256.75	5687.32	335.81	0.06
SA	4336.22	159.63	3946.41	216.88	0.09	5198.78	305.24	4740.06	414.04	0.09	5641.50	512.06	5142.12	663.00	0.09

5.6 Prohibiting drones from reconnoitering retrieval nodes

To investigate the impact of drone reconnoiter retrieval nodes on objective value, we conducted a comparative analysis between two scenarios: when retrieval nodes can be reconnoitered by drones, and when they cannot. We generated 72 instances using the kroA100, kroB100, kroA150, kroB150, kroA200, kroB200, and lin318 instances from the gen2, gen3, and gen4 datasets. The instance sizes were 25, 50, 100, and 200, and each size had *Nu* instances. Each instance was run 10 times while the average and maximum objective values were recorded. Appendix B Table B5 provides more details on each test instance. *RD* and *NRD* denote the situations where drones can and cannot reconnoiter retrieval nodes, respectively. Table 10 summarizes the experimental results. In the table, $\#RD_{max}$ represents the number of instances where the maximum objective value of *RD* is greater than or equal to that of *NRD*, and $\#RD_{avg}$ represents the number of instances where the average objective value of *RD* is greater than or equal to that of *NRD*. The results of the analysis show that 68.06% of the maximum objective value at *RD* is not lower than at *NRD*, and 70.83% of the average objective value at *RD* is not lower than at *NRD*. This indicates that allowing the drones to reconnoiter retrieval nodes very likely helps to obtain a better objective value. This result aligns with the actual situation, allowing expanded solution numbers for drone reconnoiter retrieval nodes, and thereby increasing the likelihood of obtaining better objective value.

Table 10
Comparing RD and NRD

	N=25	*N*=50	*N*=100	*N*=200	Avg
$\#RD_{max}$	18.00	12.00	10.00	9.00	12.25
$\#RD_{max}/Nu$	100.00%	66.67%	55.56%	50.00%	68.06%
$\#RD_{avg}$	10.00	16.00	11.00	14.00	12.75
$\#RD_{avg}/Nu$	55.56%	88.87%	61.11%	77.78%	70.83%

6. Conclusion

This paper has proposed the heterogeneous drones–helicopter orienteering problem, which involves the synchronous routing

of a helicopter with multiple heterogeneous drones for reconnaissance. The drones can visit multiple mission nodes in each trip. To solve this problem, we developed a mixed-integer linear programming (MILP) model. However, commercial solvers are limited to handling small-scale instances. To address medium-sized and large-scale instances, which better reflect real-world scenarios, the A-SA algorithm and a novel evaluation method based on segment sorting were designed. We randomly generated some new instances based on benchmark instances from the orienteering problem and evaluated the performance of the proposed algorithm. Experiments showed that the new evaluation method can accelerate the evaluation of the synchronized routing problem for helicopters with multiple drones. Subsequently, the influence of whether or not drones can or cannot reconnoiter retrieval nodes was analyzed, and approximately 70% of the results of the examples show that allowing the drones to reconnoiter the retrieval nodes improves the objective value.

Based on this work, future research could incorporate more real-world factors into the model, such as mission node dynamic information, terrain, and wind speed, and address the limitation of communication distance between drones and helicopters. In addition, we could explore the use of helicopters and drones in post-disaster transportation of emergency goods (such as blood and drugs) and study how to employ helicopters in collaboration with drones to perform different missions effectively.

Acknowledgment

This work was supported by the National Natural Science Foundation of China [grant numbers 72071122,72134004]; the Natural Science Foundation of Shandong Province [grant number ZR2020MG002]; and the Social Science Planning Research Project of Shandong Province [grant number 20CGLJ11].

References

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization Approaches for the Traveling Salesman Problem with Drone. *Transportation Science*, 52(4), 965-981. https://doi.org/10.1287/trsc.2017.0791

Avi, A., Frisco, N., Giurato, M., Lovera, M., Masarati, P., Panza, S., ... & Quaranta, G. (2022). Scout Drone: a Drone-Helicopter Collaboration to Support HEMS Missions. In *48th European Rotorcraft Forum (ERF 2022)* (pp. 1-8).

Cavani, S., Iori, M., & Roberti, R. (2021). Exact Methods for the Traveling Salesman Problem with Multiple Drones. *Transportation Research Part C: Emerging Technologies*, 130, 103280. https://doi.org/10.1016/j.trc.2021.103280

Chung, S. H., Sah, B., & Lee, J. (2020). Optimization for Drone and Drone-Truck Combined Operations: A Review of the State of the Art and Future Directions. *Computers & Operations Research*, 123, 105004. https://doi.org/10.1016/j.cor.2020.105004

Demir, E., Bektas, T., & Laporte, G. (2012). An Adaptive Large Neighborhood Search Heuristic for the Pollution-Routing Problem. *European Journal of Operational Research*, 223(2), 346-359. https://doi.org/10.1016/j.ejor.2012.06.044

Golden, B. L., Levy, L., & Vohra, R. (1987). The Orienteering Problem. *Naval Research Logistics*, 34(3), 307-318. https://doi.org/10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D

Gonzalez-R, P. L., Canca, D., Andrade-Pineda, J. L., Calle, M., & Leon-Blanco, J. M. (2020). Truck-Drone Team Logistics: A Heuristic Approach to Multi-Drop Route Planning. *Transportation Research Part C: Emerging Technologies*, 114, 657-680. https://doi.org/10.1016/j.trc.2020.02.030

Gu, R., Poon, M., Luo, Z., Liu, Y., & Liu, Z. (2022). A Hierarchical Solution Evaluation Method and a Hybrid Algorithm for the Vehicle Routing Problem with Drones and Multiple Visits. *Transportation Research Part C: Emerging Technologies*, 141, 103733. https://doi.org/10.1016/j.trc.2022.103733

Gunawan, A., Lau, H. C., & Vansteenwegen, P. (2016). Orienteering Problem: A Survey of Recent Variants, Solution Approaches and Applications. *European Journal of Operational Research,* 255(2), 315-332. https://doi.org/10.1016/j.ejor.2016.04.059.

Optimization, G. (2020). Gurobi Optimizer Reference Manual. < http://www.gurobi.com >

Kim, H., Kim, B. I., & Noh, D. J. (2020). The Multi-Profit Orienteering Problem. *Computers & Industrial Engineering*, 149, 106808. https://doi.org/10.1016/j.cie.2020.106808

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science* 220(4598), 671-680. https://doi.org/10.1126/science.220.4598.671

Kitjacharoenchai, P., Min, B.-C., & Lee, S. (2020). Two Echelon Vehicle Routing Problem with Drones in Last Mile Delivery. *International Journal of Production Economics*, 225, 107598. https://doi.org/10.1016/j.ijpe.2019.107598

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M. A., & Brunese, P. A. (2019). Multiple Traveling Salesman Problem with Drones: Mathematical Model and Heuristic Approach. *Computers & Industrial Engineering*, 129, 14-30. https://doi.org/10.1016/j.cie.2019.01.020

Kobeaga, G., Merino, M., & Lozano, J. A. (2018). An Efficient Evolutionary Algorithm for the Orienteering Problem. *Computers & Operations Research*, 90, 42-59. https://doi.org/10.1016/j.cor.2017.09.003

Liu, Y., Qi, N., Yao, W., Zhao, J., & Xu, S. (2020). Cooperative Path Planning for Aerial Recovery of a Uav Swarm Using Genetic Algorithm and Homotopic Approach. *Applied Sciences*, 10(12), 4154. https://doi.org/10.3390/app10124154

Luo, Q., Wu, G., Ji, B., Wang, L., & Suganthan, P. N. (2022). Hybrid Multi-Objective Optimization Approach with Pareto Local Search for Collaborative Truck-Drone Routing Problems Considering Flexible Time Windows. *IEEE Transactions on Intelligent Transportation Systems*, 23(8), 13011-13025. https://doi.org/10.1109/tits.2021.3119080

Luo, Z., Poon, M., Zhang, Z., Liu, Z., & Lim, A. (2021). The Multi-Visit Traveling Salesman Problem with Multi-Drones. *Transportation Research Part C: Emerging Technologies*, 128, 103172. https://doi.org/10.1016/j.trc.2021.103172

Losey, S. (2021). DARPA nabs Gremlin drone in midair for first time. <https://www.defensenews.com/unmanned/2021/11/05/darpa-nabs-gremlin-drone-in-midair-for-first-time/ >

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., & Laporte, G. (2020). Drone-Aided Routing: A Literature Review. *Transportation Research Part C: Emerging Technologies*, 120, 102762. https://doi.org/10.1016/j.trc.2020.102762

Mahmoudi, B., & Eshghi, K. (2022). Energy-Constrained Multi-Visit Tsp with Multiple Drones Considering Non-Customer Rendezvous Locations. *Expert Systems with Applications*, 210, 118479. https://doi.org/10.1016/j.eswa.2022.118479

Masmoudi, M. A., Hosny, M., Braekers, K., & Dammak, A. (2016). Three Effective Metaheuristics to Solve the Multi-Depot Multi-Trip Heterogeneous Dial-a-Ride Problem. *Transportation Research Part E: Logistics and Transportation Review*, 96, 60-80. https://doi.org/10.1016/j.tre.2016.10.002

Moshref-Javadi, M., Hemmati, A., & Winkenbach, M. (2020). A Truck and Drones Model for Last-Mile Delivery: A Mathematical Model and Heuristic Approach. *Applied Mathematical Modelling*, 80, 290-318. https://doi.org/10.1016/j.apm.2019.11.020

Moshref-Javadi, M., & Winkenbach, M. (2021). Applications and Research Avenues for Drone-Based Models in Logistics: A Classification and Review. *Expert Systems with Applications*, 177, 114854. https://doi.org/10.1016/j.eswa.2021.114854

Murray, C. C., & Chu, A. G. (2015). The Flying Sidekick Traveling Salesman Problem: Optimization of Drone-Assisted Parcel Delivery. *Transportation Research Part C: Emerging Technologies*, 54, 86-109. https://doi.org/10.1016/j.trc.2015.03.005

Murray, C. C., & Raj, R. (2020). The Multiple Flying Sidekicks Traveling Salesman Problem: Parcel Delivery with Multiple Drones. *Transportation Research Part C: Emerging Technologies*, 110, 368-398. https://doi.org/10.1016/j.trc.2019.11.003

Mizokami, K. (2020). Black Hawk Helicopters Can Now Launch Drones From Midair.<https://www.popularmechanics.com/military/weapons/a32617628/black-hawk-drones/ >

Masmoudi, M. A., Mancini, S., Baldacci, R., & Kuo, Y. H. (2022). Vehicle Routing Problems with Drones Equipped with Multi-Package Payload Compartments. *Transportation Research Part E: Logistics and Transportation Review*, 164, 102757. https://doi.org/10.1016/j.tre.2022.102757

Matijević, L. (2023). General variable neighborhood search for electric vehicle routing problem with time-dependent speeds and soft time windows.*International Journal of Industrial Engineering Computations*, 14(2), 275-275. https://doi.org/10.5267/j.ijiec.2023.2.001

Morandi, N., Leus, R., Matuschke, J., & Yaman, H. (2023). The Traveling Salesman Problem with Drones: The Benefits of Retraversing the Arcs. *Transportation Science.* https://doi.org/10.1287/trsc.2022.0230

Poikonen, S., & Golden, B. (2020). The Mothership and Drone Routing Problem. *INFORMS Journal on Computing*, 32(2), 249-262. https://doi.org/10.1287/ijoc.2018.0879

Ponza, A. (2016). Optimization of drone-assisted parcel delivery. https://doi.org/10.13140/RG.2.2.24444.56962

Ropke, S., & Pisinger, D. (2006). An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(4), 455-472. https://doi.org/10.1287/trsc.1050.0135

Salama, M. R., & Srinivas, S. (2022). Collaborative Truck Multi-Drone Routing and Scheduling Problem: Package Delivery with Flexible Launch and Recovery Sites. *Transportation Research Part E: Logistics and Transportation Review*, 164, 102788. https://doi.org/10.1016/j.tre.2022.102788

Schermer, D., Moeini, M., & Wendt, O. (2019). A Matheuristic for the Vehicle Routing Problem with Drones and Its Variants. *Transportation Research Part C: Emerging Technologies,* 106, 166-204. https://doi.org/10.1016/j.trc.2019.06.016

Tamke, F., & Buscher, U. (2021). A Branch-and-Cut Algorithm for the Vehicle Routing Problem with Drones. *Transportation Research Part B Methodological,* 144, 174-203. https://doi.org/10.1016/j.trb.2020.11.011

Tu, P. A., Dat, N. T., & Dung, P. Q. (2018, December). Traveling salesman problem with multiple drones. In *Proceedings of the 9th International Symposium on Information and Communication Technology* (pp. 46-53).

Vansteenwegen, P., Souffriau, W., & Van Oudheusden, D. (2011). The Orienteering Problem: A Survey. *European Journal of Operational Research,* 209(1), 1-10. https://doi.org/10.1016/j.ejor.2010.03.045

Wang, K., Pesch, E., Kress, D., Fridman, I., & Boysen, N. (2022). The Piggyback Transportation Problem: Transporting Drones Launched from a Flying Warehouse. *European Journal of Operational Research,* 296(2), 504-519. https://doi.org/10.1016/j.ejor.2021.03.064

Wang, X., Poikonen, S., & Golden, B. (2016). The Vehicle Routing Problem with Drones: Several Worst-Case Results. *Optimization Letters*, 11(4), 679-697. https://doi.org/10.1007/s11590-016-1035-3

Wang, Z., & Sheu, J.-B. (2019). Vehicle Routing Problem with Drones. *Transportation Research Part B Methodological,* 122, 350-364. https://doi.org/10.1016/j.trb.2019.03.005.

Wen, X., & Wu, G. (2022). Heterogeneous Multi-Drone Routing Problem for Parcel Delivery. *Transportation Research Part C: Emerging Technologies,* 141, 103763. https://doi.org/10.1016/j.trc.2022.103763

Yoon, J. J. (2018). *The traveling salesman problem with multiple drones: an optimization model for last-mile delivery* (Doctoral dissertation, Massachusetts Institute of Technology).

Zhao, L., Bi, X., Li, G., Dong, Z., Xiao, N., & Zhao, A. (2022). Robust Traveling Salesman Problem with Multiple Drones: Parcel Delivery under Uncertain Navigation Environments. *Transportation Research Part E: Logistics and Transportation Review*, 168, 102967. https://doi.org/10.1016/j.tre.2022.102967

Appendix A.

Supplementary constraints

$$t_j^L \geq t_i^L + t_{ij}^H + s_j^H - M(1 - x_{ij} + \sum_{m\in\delta^+(i),k\in K,j\neq m} y_{im}^k + \sum_{l\in\delta^-(j),k\in K,l\neq i} y_{lj}^k) \quad (i,j)\in A,\ j\neq e \tag{A.1}$$

$$t_j^L \leq t_i^L + t_{ij}^H + s_j^H + M(1 - x_{ij} + \sum_{m\in\delta^+(i),k\in K,j\neq m} y_{im}^k + \sum_{l\in\delta^-(j),k\in K,l\neq i} y_{lj}^k) \quad (i,j)\in A,\ j\neq e \tag{A.2}$$

$$t_e^L \leq t_i^L + t_{i,e}^H + M(1 - x_{i,e}) \quad i\in N\backslash e \tag{A.3}$$

$$b_{kj}^- \geq b_{ki}^+ - t_{ijk}^D - M(1 - y_{ij}^k + x_{ij}) \quad (i,j)\in A, \forall k\in K \tag{A.4}$$

$$b_{kj}^+ \geq b_{kj}^- - s_{jk}^D - M(1 - y_{ij}^k + x_{ij} + \sum_{l\in\delta^-(j),l\neq i} x_{lj}) \quad (i,j)\in A, k\in K, j\neq e \tag{A.5}$$

$$(T_{max}^H + \max\{Q_k\})(\sum_{i\in\delta^-(j)} x_{ij} + \sum_{i\in\delta^-(j),k\in K} y_{ij}^k) \geq t_j^L \quad \forall j\in C \tag{A.6}$$

$$Q_k(\sum_{i\in\delta^-(j)} y_{ij}^k) \geq b_{kj}^- \quad \forall j\in C, k\in K \tag{A.7}$$

$$Q_k(\sum_{i\in\delta^-(j)} y_{ij}^k) \geq b_{kj}^+ \quad \forall j\in C, k\in K \tag{A.8}$$

$$t_j^L - t_i^L \geq b_{ki}^+ - b_{kj}^+ - M(1 - y_{ij}^k + x_{ij} + \sum_{m\in\delta^+(i),j\neq m} x_{im} + \sum_{l\in\delta^-(j),l\neq i} x_{lj}) \quad (i,j)\in A, \forall k\in K, j\neq e \tag{A.9}$$

$$t_j^L - t_i^L \leq b_{ki}^+ - b_{kj}^+ + M(1 - y_{ij}^k + x_{ij} + \sum_{m\in\delta^+(i),j\neq m} x_{im} + \sum_{l\in\delta^-(j),l\neq i} x_{lj}) \quad (i,j)\in A, \forall k\in K, j\neq e \tag{A.10}$$

$$z_j^H + z_j^D \leq 1 \quad \forall j\in C \tag{A.11}$$

Constraints (A1) and (A2) determine the time relationship for the helicopter non-launch and non-retrieval routes. Constraints (A3) and Constraints (19) in the model calculate the time for the helicopter to return to the depot. Constraints (A4) and Constraints (24) in the model calculate the power consumption between two nodes among drone routes. Constraints (A5) and Constraints (25) in the model calculate the power consumption of the drones to reconnoiter mission nodes. Constraints (A6) ensure that unvisited nodes' t_j^L is equal to 0. Constraints (A7) and (A8) ensure that unvisited nodes' b_{kj}^- and b_{kj}^+ are equal to 0. Constraints (A9) and (A10) enforce that the power consumption of the drones during a trip corresponds to the time difference between its consecutive nodes. Constraints (A11) guarantee that any mission node can be reconnoitered by the drones or the helicopter at most once.

Appendix B.

Numerical experimental data

Table B1

Calibration experiment results of reaction factor β

N=10	0.1		0.3		0.5		0.7		0.9	
Inst.	S_b	T_b	S_b	T_b	S_b	T_b	S_b	T_b	S_b	T_b
lin105-gen2-10-1	233	27.19	233	26.73	233	25.15	233	25.30	233	27.82
lin105-gen2-10-2	261	26.19	261	25.70	261	24.78	261	24.89	261	24.45
lin105-gen3-10-1	63	25.31	63	28.63	63	26.27	63	30.18	63	27.38
lin105-gen3-10-2	72	25.34	72	28.83	72	28.78	72	29.86	72	28.59
lin105-gen4-10-1	233	25.98	233	26.46	233	27.76	233	29.28	233	36.27
lin105-gen4-10-2	261	26.26	261	28.27	261	26.92	261	29.81	261	24.46
Average	187.17	26.05	187.17	27.45	187.17	26.61	187.17	28.22	187.17	28.16
N=25	0.1		0.3		0.5		0.7		0.9	
Inst.	S_b	T_b	S_b	T_b	S_b	T_b	S_b	T_b	S_b	T_b
lin105-gen2-25-1	628	22.51	657	15.85	657	17.95	657	17.83	657	19.82
lin105-gen2-25-2	613	18.17	613	17.49	675	18.76	674	18.33	674	21.62
lin105-gen3-25-1	199	28.87	199	24.26	199	25.16	199	25.87	199	26.67
lin105-gen3-25-2	209	20.06	212	17.78	198	23.45	212	22.71	212	21.60
lin105-gen4-25-1	657	17.53	657	23.88	657	21.19	657	25.37	657	25.59
lin105-gen4-25-2	675	16.39	674	19.81	675	15.78	674	20.84	674	13.56
Average	496.83	20.59	502.00	19.85	510.17	20.38	512.17	21.83	512.17	21.48
N=50	0.1		0.3		0.5		0.7		0.9	
Inst.	S_b	T_b	S_b	T_b	S_b	T_b	S_b	T_b	S_b	T_b
lin105-gen2-50-1	803	31.08	769	22.6	803	35.12	803	40.86	824	42.88
lin105-gen2-50-2	833	60.29	833	46.97	833	46.00	833	38.16	833	47.89
lin105-gen3-50-1	255	76.05	256	75.32	240	83.28	256	102.72	256	85.31
lin105-gen3-50-2	266	86.92	266	104.60	266	92.27	266	87.82	266	86.33
lin105-gen4-50-1	824	77.45	803	45.26	824	56.55	824	57.40	824	48.15
lin105-gen4-50-2	833	65.26	833	45.55	833	62.79	833	36.40	833	51.57
Average	635.67	66.18	626.67	56.72	633.17	62.67	635.83	60.56	639.33	60.36

Table B2
Summary of average computational time per method (in milliseconds)

	N=10	*N*=20	*N*=30	*N*=40	*N*=50	*N*=60	*N*=70	*N*=80
LP	3.2504	52.7976	315.8955	732.5333	645.1020	1591.1075	3487.2586	4449.0487
SSE	0.0000	0.1123	0.1925	0.2325	0.2740	0.3759	0.3803	0.4305
	N=90	*N*=100	*N*=110	*N*=120	*N*=130	*N*=140	*N*=150	*N*=160
LP	7783.3895	10081.7623	11820.9198	22159.5686	28218.0217	35054.5429	41800.4544	49953.6652
SSE	0.4860	0.5347	0.5543	0.6345	0.7071	0.7573	0.8071	0.8360

Table B3
Comparison of Gurobi and A-SA for small Instances

N=8		Gurobi		A-SA	
Inst.	Opt?	S_b	T_{tot}	S_b	T_b
lin105-gen2-8-1	Y	170	94.34	170	1.83
lin105-gen2-8-2	Y	216	125.75	216	1.26
lin105-gen2-8-3	Y	198	319.83	198	1.16
lin105-gen2-8-4	Y	101	207.82	101	2.72
lin105-gen2-8-5	Y	63	217.37	63	1.77
lin105-gen3-8-1	Y	22	116.68	22	1.90
lin105-gen3-8-2	Y	40	102.17	40	1.09
lin105-gen3-8-3	Y	115	125.58	115	1.55
lin105-gen3-8-4	Y	77	189.76	77	0.95
lin105-gen3-8-5	Y	41	202.73	41	1.16
lin105-gen4-8-1	Y	170	96.52	170	1.20
lin105-gen4-8-2	Y	216	139.12	216	1.70
lin105-gen4-8-3	Y	198	315.53	198	1.19
lin105-gen4-8-4	Y	101	151.47	101	1.19
lin105-gen4-8-5	Y	63	192.23	63	0.76
N=9		Gurobi		A-SA	
Inst.	Opt?	S_b	T_{tot}	S_b	T_b
lin105-gen2-9-1	Y	231	1170.25	231	1.78
lin105-gen2-9-2	N	218	1800.06	218	2.04
lin105-gen2-9-3	N	205	1800.06	205	1.55
lin105-gen2-9-4	N	101	1800.02	101	1.59
lin105-gen2-9-5	N	88	1800.05	88	1.41
lin105-gen3-9-1	Y	41	1404.76	41	2.38
lin105-gen3-9-2	N	62	1800.03	62	2.02
lin105-gen3-9-3	N	115	1800.05	115	2.00
lin105-gen3-9-4	N	77	1800.07	77	1.09
lin105-gen3-9-5	N	57	1800.02	57	1.41
lin105-gen4-9-1	Y	170	1253.73	170	1.38
lin105-gen4-9-2	N	231	1800.02	231	1.60
lin105-gen4-9-3	N	218	1800.08	218	1.33
lin105-gen4-9-4	N	101	1800.06	101	1.09
lin105-gen4-9-5	N	88	1800.05	88	1.09

Table B4
Comparing A-SA and SA for large Instances

	N=100									
	A-SA					SA				
Inst.	S_b	T_b	S_{avg}	T_{avg}	Gap_{avg}	S_b	T_b	S_{avg}	T_{avg}	Gap_{avg}
kroA100-gen2	4783.00	116.46	4303.60	141.27	0.10	4107.00	150.74	3844.80	229.57	0.06
kroB100-gen2	4645.00	141.68	4480.40	164.47	0.04	4497.00	155.86	4222.80	187.66	0.06
kroA100-gen3	4721.00	123.10	4549.2	148.69	0.04	4311.00	141.58	3750.20	180.98	0.13
kroB100-gen3	4029.00	118.82	3909.30	141.07	0.03	3913.00	155.95	3285.00	172.08	0.16
kroA100-gen4	4583.00	118.35	4269.10	143.96	0.07	4092.00	156.75	3933.20	179.24	0.04
kroB100-gen4	4568.00	128.39	4410.10	145.09	0.03	4528.00	176.11	4194.40	230.51	0.07
kroA150-gen2-100-1	4668.00	140.88	4288.30	154.18	0.08	4247.00	159.27	4074.40	194.21	0.04
kroB150-gen2-100-1	4270.00	129.09	4162.00	144.45	0.03	4116.00	160.10	4000.80	218.67	0.03
kroA150-gen3-100-1	4619.00	138.03	4451.10	163.64	0.04	4560.00	137.51	3731.80	248.68	0.18
kroB150-gen3-100-1	4589.00	124.83	4529.20	155.01	0.01	4572.00	158.87	4386.20	243.86	0.04
kroA150-gen4-100-1	4271.00	123.81	4171.60	168.11	0.02	4663.00	147.89	4112.60	185.46	0.12
kroB150-gen4-100-1	4646.00	129.83	4497.80	167.53	0.03	4339.00	187.61	3940.80	253.90	0.09
KroA200-gen2-100-1	4373.00	143.38	4179.60	170.24	0.04	4083.00	194.84	3852.40	225.82	0.06
KroB200-gen2-100-1	4675.00	141.71	4501.40	165.13	0.04	4697.00	172.15	4173.80	206.78	0.11
KroA200-gen3-100-1	4317.00	131.46	4114.60	148.79	0.05	4310.00	144.19	3737.00	398.14	0.13
KroB200-gen3-100-1	4027.00	133.83	3902.00	151.33	0.03	4021.00	166.61	3453.80	178.34	0.14
KroA200-gen4-100-1	4529.00	104.68	4167.80	129.06	0.08	4440.00	164.48	4002.60	209.80	0.10
KroB200-gen4-100-1	4605.00	105.24	4428.50	123.42	0.04	4556.00	142.79	4338.80	160.11	0.05

Table B4
Comparing A-SA and SA for large Instances (Continued)

	N=150									
	A-SA					SA				
Inst.	S_b	T_b	S_{avg}	T_{avg}	Gap_{avg}	S_b	T_b	S_{avg}	T_{avg}	Gap_{avg}
kroA150-gen2	5540.00	132.15	5271.80	148.78	0.05	5258.00	310.79	4988.40	360.06	0.05
kroA150-gen3	6156.00	138.74	5467.40	173.44	0.11	5993.00	303.22	5147.60	373.28	0.14
kroA150-gen4	5595.00	153.17	5300.00	162.54	0.05	5444.00	286.90	4987.60	353.31	0.08
kroB150-gen2	5370.00	133.53	5093.30	177.18	0.05	4984.00	302.87	4736.40	417.25	0.05
kroB150-gen3	6126.00	133.55	5639.70	199.43	0.08	5279.00	311.36	4805.00	395.10	0.09
kroB150-gen4	5322.00	124.20	5013.90	147.07	0.06	5026.00	318.51	4622.80	365.42	0.08
kroA200-gen2-150-1	5386.00	140.13	5081.80	165.44	0.06	5085.00	357.89	4678.00	419.57	0.08
kroA200-gen3-150-1	5362.00	121.88	5007.60	192.39	0.07	5069.00	284.26	4500.60	557.84	0.11
kroA200-gen4-150-1	5133.00	133.66	5033.50	150.94	0.02	5735.00	286.39	5128.80	447.32	0.11
kroB200-gen2-150-1	5580.00	148.13	5377.40	160.01	0.04	5395.00	316.29	4896.40	328.28	0.09
kroB200-gen3-150-1	5738.00	133.14	5448.20	174.25	0.05	5291.00	321.37	4994.00	364.29	0.06
kroB200-gen4-150-1	5571.00	148.46	5343.80	168.50	0.04	5124.00	251.20	4595.80	315.61	0.10
kroA200-gen2-150-10	5136.00	132.67	4806.70	155.74	0.06	4704.00	359.69	4405.60	568.20	0.06
kroA200-gen3-150-10	5413.00	140.85	4854.30	166.35	0.10	4249.00	351.38	3750.80	434.25	0.12
kroA200-gen4-150-10	5068.00	114.61	4876.10	178.18	0.04	5156.00	281.52	4726.00	587.36	0.08
kroB200-gen2-150-10	5693.00	140.93	5317.80	161.12	0.07	5208.00	257.70	4968.00	354.39	0.05
kroB200-gen3-150-10	5357.00	151.47	4870.00	175.99	0.09	5210.00	261.21	4642.40	385.70	0.11
kroB200-gen4-150-10	5701.00	141.27	5327.90	178.55	0.07	5368.00	331.75	4746.80	425.42	0.12
	N=200									
	A-SA					SA				
Inst.	S_b	T_b	S_{avg}	T_{avg}	Gap_{avg}	S_b	T_b	S_{avg}	T_{avg}	Gap_{avg}
kroA200-gen2	6396.00	204.16	5960.50	274.52	0.07	5505.00	592.87	5078.20	721.21	0.08
kroB200-gen2	6573.00	220.27	6186.50	272.33	0.06	6224.00	477.01	5720.00	611.11	0.08
kroA200-gen3	5957.00	256.64	5797.90	327.44	0.03	5766.00	547.99	4929.40	653.35	0.15
kroB200-gen3	6133.00	235.86	5827.00	317.38	0.05	5329.00	521.10	4675.80	620.41	0.12
kroA200-gen4	6285.00	289.70	5988.90	371.21	0.05	6023.00	434.66	5690.60	553.70	0.06
kroB200-gen4	6522.00	188.87	6122.90	312.35	0.06	6207.00	476.45	5831.00	536.48	0.06
lin318-gen2-200-1	6392.00	280.17	6155.50	366.70	0.04	5375.00	551.41	5248.00	765.29	0.02
lin318-gen3-200-1	4641.00	292.07	4051.70	382.99	0.13	4305.00	536.38	3569.00	746.23	0.17
lin318-gen4-200-1	6251.00	258.30	5947.50	319.34	0.05	6205.00	530.27	5439.40	683.09	0.12
lin318-gen2-200-10	6269.00	281.52	5909.90	376.89	0.06	5647.00	491.37	5427.40	611.39	0.04
lin318-gen3-200-10	4884.00	271.68	4512.50	360.53	0.08	4575.00	466.98	4139.80	615.07	0.10
lin318-gen4-200-10	6387.00	299.64	6184.10	387.25	0.03	6179.00	503.21	5667.80	628.65	0.08
lin318-gen2-200-20	6307.00	284.63	6060.40	368.23	0.04	5782.00	600.48	5276.00	716.37	0.09
lin318-gen3-200-20	4935.00	278.72	4603.00	324.56	0.07	4386.00	493.25	4143.60	771.90	0.06
lin318-gen4-200-20	6369.00	302.23	6011.20	380.76	0.06	6441.00	593.97	5882.60	667.76	0.09
lin318-gen2-200-30	6349.00	199.87	6082.00	281.74	0.04	6584.00	511.95	5663.80	658.08	0.14
lin318-gen3-200-30	5184.00	270.59	4834.60	342.42	0.07	4652.00	426.31	4297.00	719.58	0.08
lin318-gen4-200-30	6428.00	206.59	6135.60	277.97	0.05	6362.00	461.42	5878.80	654.35	0.08

Table B5
Comparing RD and NRD

	N=25					*N*=50			
	RD		*NRD*			*RD*		*NRD*	
Inst.	S_b	S_{avg}	S_b	S_{avg}	Inst.	S_b	S_{avg}	S_b	S_{avg}
kroA100-gen2-25-1	1113.00	1034.40	1113.00	1050.70	kroA100-gen2-50-1	1825.00	1699.70	1820.00	1646.00
kroA100-gen3-25-1	1114.00	1040.90	1114.00	1023.50	kroA100-gen3-50-1	1865.00	1711.70	1751.00	1658.50
kroA100-gen4-25-1	1113.00	1059.40	1113.00	1073.70	kroA100-gen4-50-1	1796.00	1718.40	1837.00	1651.10
kroA150-gen2-25-1	1113.00	1058.00	1113.00	1048.70	kroA150-gen2-50-1	1774.00	1659.10	1747.00	1655.40
kroA150-gen3-25-1	1130.00	1024.90	1130.00	1039.70	kroA150-gen3-50-1	1792.00	1736.10	1771.00	1704.40
kroA150-gen4-25-1	1113.00	1069.10	1113.00	1041.00	kroA150-gen4-50-1	1814.00	1671.90	1841.00	1633.90
kroA200-gen2-25-1	1118.00	997.70	1106.00	997.80	kroA200-gen2-50-1	1779.00	1652.00	1875.00	1685.80
kroA200-gen3-25-1	955.00	909.40	955.00	913.70	kroA200-gen3-50-1	1562.00	1424.00	1594.00	1490.40
kroA200-gen4-25-1	1118.00	991.60	1060.00	1030.50	kroA200-gen4-50-1	1807.00	1656.90	1803.00	1622.00
kroB100-gen2-25-1	1176.00	1155.50	1176.00	1151.30	kroB100-gen2-50-1	1924.00	1830.40	1985.00	1790.20
kroB100-gen3-25-1	1169.00	1152.60	1169.00	1150.80	kroB100-gen3-50-1	1933.00	1779.80	1824.00	1722.20
kroB100-gen4-25-1	1176.00	1157.40	1169.00	1134.10	kroB100-gen4-50-1	2060.00	1924.80	2023.00	1795.20
kroB150-gen2-25-1	1138.00	1063.20	1060.00	995.50	kroB150-gen2-50-1	1793.00	1666.50	1853.00	1637.40
kroB150-gen3-25-1	1034.00	968.00	1034.00	964.40	kroB150-gen3-50-1	1773.00	1584.10	1761.00	1554.20
kroB150-gen4-25-1	1118.00	1067.60	1060.00	1019.40	kroB150-gen4-50-1	1903.00	1674.10	1864.00	1613.20
kroB200-gen2-25-1	1176.00	1154.60	1176.00	1147.30	kroB200-gen2-50-1	2060.00	1873.40	1945.00	1823.20
kroB200-gen3-25-1	1169.00	1145.10	1169.00	1146.60	kroB200-gen3-50-1	1813.00	1739.20	1774.00	1678.20
kroB200-gen4-25-1	1176.00	1117.10	1164.00	1140.20	kroB200-gen4-50-1	2028.00	1881.70	1952.00	1856.30

Table B5
Comparing RD and NRD (continued)

	N=100					*N*=200			
	RD		*NRD*			*RD*		*NRD*	
Inst.	S_b	S_{avg}	S_b	S_{avg}	Inst.	S_b	S_{avg}	S_b	S_{avg}
kroA100-gen2	4783.00	4303.60	4508.00	4226.90	kroA200-gen2	6396.00	5960.50	6249.00	5950.80
kroB100-gen2	4645.00	4480.40	4495.00	4340.70	kroB200-gen2	6573.00	6186.50	6450.00	6131.40
kroA100-gen3	4721.00	4549.2	4679.00	4373.80	kroA200-gen3	5957.00	5797.90	6086.00	5727.80
kroB100-gen3	4029.00	3909.30	4011.00	3929.70	kroB200-gen3	6133.00	5827.00	6392.00	5866.60
kroA100-gen4	4583.00	4269.10	4499.00	4246.60	kroA200-gen4	6285.00	5988.90	6236.00	5892.30
kroB100-gen4	4568.00	4410.10	4690.00	4365.40	kroB200-gen4	6522.00	6122.90	6353.00	5986.60
kroA150-gen2-100-1	4668.00	4288.30	4414.00	4147.00	lin318-gen2-200-1	6392.00	6155.50	6305.00	6008.00
kroB150-gen2-100-1	4270.00	4162.00	4363.00	4031.50	lin318-gen3-200-1	4641.00	4051.70	4574.00	4049.10
kroA150-gen3-100-1	4619.00	4451.10	4687.00	4510.80	lin318-gen4-200-1	6251.00	5947.50	6321.00	5971.40
kroB150-gen3-100-1	4589.00	4529.20	4699.00	4532.70	lin318-gen2-200-10	6269.00	5909.90	6347.00	5997.20
kroA150-gen4-100-1	4271.00	4171.60	4588.00	4275.00	lin318-gen3-200-10	4884.00	4512.50	4622.00	4317.50
kroB150-gen4-100-1	4646.00	4497.80	4314.00	4113.40	lin318-gen4-200-10	6387.00	6184.10	6418.00	6061.50
KroA200-gen2-100-1	4373.00	4179.60	4398.00	4195.20	lin318-gen2-200-20	6307.00	6060.40	6660.00	6130.80
KroB200-gen2-100-1	4675.00	4501.40	4630.00	4432.40	lin318-gen3-200-20	4935.00	4603.00	5110.00	4560.30
KroA200-gen3-100-1	4317.00	4114.60	4362.00	4185.00	lin318-gen4-200-20	6369.00	6011.20	6530.00	5914.10
KroB200-gen3-100-1	4027.00	3902.00	3972.00	3875.20	lin318-gen2-200-30	6349.00	6082.00	6791.00	6019.20
KroA200-gen4-100-1	4529.00	4167.80	4347.00	4181.80	lin318-gen3-200-30	5184.00	4834.60	5165.00	4815.20
KroB200-gen4-100-1	4605.00	4428.50	4613.00	4347.10	lin318-gen4-200-30	6428.00	6135.60	6336.00	5997.60