The Multi-visit Drone-assisted Routing Problem with Soft Time Windows and Stochastic Truck Travel Times

Shanshan Meng [a,b*], **Dong Li** [c], **Jiyin Liu** [b], **Yanru Chen** [a,d]

[a] *School of Economics and Management, Southwest Jiaotong University, Chengdu, China*
[b] *Loughborough Business School, Loughborough University, Leicestershire LE11 3TU, United Kingdom*
[c] *Department of Management Science, Lancaster University, Lancaster, LA1 4YX, United Kingdom*
[d] *Service Science and Innovation Key Laboratory of Sichuan Province, Chengdu, China*

Abstract

We consider a combined truck-drone delivery problem with stochastic truck travel times and soft time windows. A fleet of homogeneous trucks and drones are deployed in pairs to provide delivery services to customers. Each drone can be launched from and retrieved to its truck multiple times, and in each flight, a drone can serve one or more customers. Our objective is to determine the truck routes and drone flights that minimise the total cost, including time window violation penalties. We formulate this problem into a two-stage stochastic model with recourse action in the second stage to optimise the truck waiting time at each node. We approximate the stochastic model with a large-scale mixed-integer program using the sample average approximation (SAA) framework, which is computationally intractable. To this end, we propose a hybrid metaheuristic approach that incorporates SAA. The waiting times of each truck obtained in the planning phase are optimal against the sampled or estimated travel times along the entire route, but the actual values are known only once the truck has returned to the depot. To this end, we reformulate the second-stage model in a rolling-horizon manner, which can be easily implemented and efficiently solved in the execution phase. Extensive numerical experiments demonstrate the strong performance of the proposed metaheuristic approach and rolling-horizon model. The results also highlight the clear benefits of the stochastic modelling approach over its deterministic counterpart, with a pronounced reduction in the total cost in various scenarios.

Keywords: Multi-visit drone routing; Two-stage stochastic model; Soft time windows; Stochastic truck travel times; Hybrid metaheuristic

1. Introduction

Unmanned aerial vehicles (UAVs), commonly known as drones, have shown great potential in delivery services and have attracted considerable attention in the last decade. For example, in 2017, Aha, Iceland's largest online marketplace, partnered with the drone company Flytrex and together they expanded on-demand drone delivery service for fast delivery (Edwards, 2018). In 2019, Hangzhou (China) started drone delivery in partnership with Antwork Technology, a company that has been testing drone logistics applications in urban settings since 2018 (Hu, 2020). Until now, however, drone deliveries have remained on a rather small scale and are only available in a few places. One of the major hurdles for mass adoption is a regulation that requires commercial drones to have visual observers along their routes, which is very costly, if not impractical. However, this is about to change after the recent relaxation of this requirement by the Federal Aviation Administration, which began authorising drones to fly beyond the visual line of sight (FAA., 2023). In light of this relaxation, some major operators announced expansion plans. Amazon will add a third operational site in the US and two more sites in Europe by 2024 (Jones, 2023). Zipline plans to deploy next-generation delivery drones for precise delivery to customers, including restaurants, retailers, and medical centres. They will also start operating in the UK in 2024 (Crumley, 2023). No wonder some analysts acclaim that 'drone deliveries are finally going mainstream' (Muller, 2024).

Compared to conventional vehicles, drones have many attractive benefits, such as faster delivery times, less environmental impact, and increased accessibility. Drones are highly flexible and not tied to road networks. However, owing to technological limitations, drones have a restricted cargo capacity and short flying ranges. To overcome these limitations, a truck-drone combined mode, in which trucks and drones work together to serve customers, has been proposed to utilise the advantages of both vehicles (Murray & Chu, 2015). More precisely, while a truck travels along the route to perform delivery tasks, each onboard drone can take off from the truck, detour to serve one or several customers, and then rejoin the truck along

*Corresponding author.
Email addresses: mengss2020@outlook.com (Shanshan Meng*), dong.li@lancaster.ac.uk (Dong Li), j.y.liu@lboro.ac.uk (Jiyin Liu), chenyanru@swjtu. edu.cn (Yanru Chen).

the route (we refer to such a detour as a drone flight). Drones can unload/load goods and swap batteries on trucks, allowing them to perform multiple flights along the route.

In the academic community, a pioneering study on the combined truck-drone mode is the *flying sidekick* travelling salesman problem (FSTSP), in which a truck and drone work in tandem (a vehicle pair) in parcel delivery. The drone can deliver only one parcel per flight (i.e., a single visit)(Murray & Chu, 2015). The travelling salesman problem with drone (TSP-D), a similar variant, was introduced soon after (Agatz et al., 2018), allowing each drone to revisit the location where it was launched. The vehicle routing problem with drones (VRP-D) was first introduced by Wang et al. (2017), in which multiple trucks and drones work collaboratively to serve customers to achieve the minimal makespan. Since then, many studies examine variants of combined truck-drone routing problems (Kitjacharoenchai et al., 2020; Meng et al., 2024).

However, the vast majority of existing studies on combined modes only consider deterministic problems, such as constant travel time over a certain distance. In reality, however, trucks are prone to road network conditions such as traffic congestion and road work. Actual travel times are often uncertain. This simplification is particularly detrimental in the presence of service time windows, which are often offered to and preferred by customers. Ignoring travel time uncertainties can result in poor routing decisions, leading to service misses (for hard time windows) and high violation penalties (for soft time windows). Moreover, in soft time windows, it is not always the best option for trucks to set off to the next customer immediately after service completion at the current one. It may be beneficial to wait for some time so that the arrival times of subsequent customers fall within their time windows. Therefore, once the truck/drone routes are determined, another decision is the optimal waiting time at each node to minimise time window violations. Considering stochastic truck travel times, waiting time optimisation is not a trivial task.

To see this in Figure 1, we show the route and flight of a single truck-drone pair for a network of six customers and a depot, where the travel times are shown next to each arc and the time windows are shown in square brackets next to each node. We assume that the service times for the truck and drone are 3 and 2 min, respectively, and that the drone launch and retrieval times are both 1 min. The arrival and departure times of the truck/drone are denoted by 't_1' and 't_2', respectively. Without waiting Figure 1(i) shows total 5 time window violations, which are reduced to just 2 if the truck waits at some nodes, as Figure 1(ii) shows. However, if the truck travel times are different, as in Figure 1(iii), the same waiting times would lead to all customers being served outside their time windows and a late return of the truck to the depot.

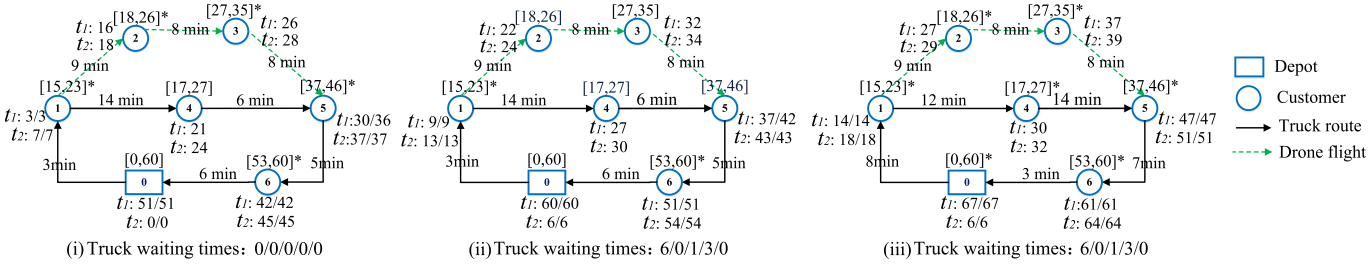

Figure 1: One route/flight with different truck travel times and waiting times.

In classical VRP problems, stochastic truck travel times have been extensively studied (Gendreau et al., 1996), as have waiting times with soft time windows (Figliozzi, 2010). Few studies considered both, and tend to focus on the optimal departure time from the depot (Taş et al., 2013). In the VRP-D literature, however, neither stochastic truck travel times nor optimal waiting times have been adequately studied. In fact, as far as we know, the closest study to our work is due to Yang et al. (2023), who consider a robust truck-drone delivery problem with road traffic uncertainty and hard time windows. No waiting time optimisation is required because the vehicles have to wait until the time windows open. They consider only a single truck and a single drone, that is, TSP-D, and the drone could only serve one customer per flight. In their model, the truck remains at the launch node while the drone is serving a customer and waits until the drone returns before moving to the next location. Such a so-called 'dispatch-wait-collect' mode is much less efficient than the synchronised mode that we consider. They also assume that it is not mandatory to serve all customers, and that the remaining customers can be served by third-party logistics or crowd-sourced drivers.

In this study, we consider a drone-assisted delivery problem with stochastic truck travel times and soft time windows. Multiple trucks and drones are deployed in pairs to work in tandem to serve all customers. Each drone serves multiple customers per flight. Our objective is to minimise the total cost, including time

window violation penalties. The cost includes the fixed cost for the deployed truck-drone pairs, travelling cost for trucks, and energy consumption for drones. The penalties include both earliness and lateness (with respect to the time windows) at all customer nodes, and lateness back to the depot. We formulate this problem into a two-stage stochastic model, with the recourse action in the second stage being the optimal waiting time (or departure time) at each node. Because of the complexity of the model, we propose a hybrid metaheuristic solution approach, which incorporates the sample average approximation (SAA) method. First, we apply a greedy-based heuristic to generate the initial solution, which is then improved using a hybrid metaheuristic approach (HMSA) with the mean truck speeds to obtain the best solution. Subsequently, the SAA framework is incorporated to further improve the solution in a stochastic setting. For simplicity, drone travel times are assumed to be deterministic, which is a reasonable assumption for drones and is common in the literature. Nevertheless, stochastic drone travel times due to weather conditions, for example, will be studied in the future.

In summary, we make the following contributions. 1) To the best of our knowledge, this work is the first to consider stochastic truck travel times with soft time windows in a multi-visit VRP-D context. This is also the first study to consider the optimal waiting times for trucks. We formulate the problem as a two-stage stochastic model in which the first stage determines the route/flights for each truck/drone pair, whereas the second stage determines the waiting time at each node, including the depot. 2) We approximate the stochastic model with a large-scale mixed-integer program (MIP) using the SAA framework, which is computationally prohibitive. Therefore, we propose a hybrid metaheuristic which incorporates SAA. 3) The waiting times obtained in the planning phase via either the second-stage model or the heuristic approach are only optimal against the realisation of the entire route of each truck, which is known only once the truck has returned to the depot. To this end, we reformulate the second-stage model in a rolling-horizon manner, which is easy to implement and efficient to solve in the execution phase. This model is re-optimised whenever a truck arrives at a customer node using an up-to-date estimate of the truck speeds in the remaining route at that time. Only the current waiting time decision is implemented, while the rest are discarded. 4) We conduct extensive numerical studies to assess the performance and efficiency of the proposed metaheuristic and rolling-horizon waiting time optimisation model in various problem settings. The results also demonstrate the benefit of the stochastic model over its deterministic counterpart, showing a pronounced reduction in total costs in various scenarios.

The remainder of the paper is organised as follows. We review the relevant literature in Section 2. Section 3 formally presents the problem and formulates it into a two-stage stochastic model. Section 4 describes our proposed solution approach. In Section 5 we present the rolling-horizon model to dynamically optimise waiting times of trucks. Section 6 presents the computational results along with managerial insights and corresponding analyses. Finally, Section 7 provides concluding remarks and future directions.

2. Literature review

For the potential benefits of the truck-drone combined delivery service, the VRP-D and other variants received considerable attention. In this section, we first review existing works on deterministic VRP-D and their extensions to consider time windows, followed by those that consider stochastic truck travel times.

2.1. Single/multi -visit VRP-D

As mentioned previously, VRP-D was first proposed by Wang et al. (2017). Two years later, Sacramento et al. (2019) investigated the operational cost of VRP-D, and Schermer et al. (2019) extended VRP-D with en route operations to reduce the makespan. Kitjacharoenchai et al. (2019) studied multiple TSP-Ds in which the dispatched drones could return to any nearby truck to minimise completion time. All of them provide heuristic approaches to solve problems, whereas other studies, such as Tamke & Buscher (2021), Zhen et al. (2023), and Zhou et al. (2023), develop exact algorithms. Some studies consider more complex cases for practical applications. For example, Rave et al. (2023) study a variant of VRP-D in which both customer locations and dedicated micro-depots could be employed for drone launch/retrieval to minimise the total cost. They develop a customised adaptive large neighbourhood search (ALNS) algorithm to solve instances with up to 200 customers. Kloster et al. (2023) consider a multiple travelling salesman problem in which drones can only be launched and retrieved at selected stations. Multiple drones can be dispatched

from each station to perform round-trip deliveries before they return to the same station. They propose three metaheuristic algorithms to minimise the makespan of problems containing 120 customers and 6 stations.

In recent years, many studies considered multiple visits during one drone flight in the VRP-D. Kitjacharoenchai et al. (2020) extend the FSTSP with multiple visits to further minimise the completion time. Kyriakakis et al. (2022) investigate a different variant, the electric VRP-D, in which electric vehicles (EVs) work as a mobile depot travelling between pre-designated stations to dispatch drones for final deliveries. In their work, the EVs follow the 'dispatch-wait-collect' procedure at stations and drones can deliver multiple parcels per flight. Four variants of the ant colony optimisation framework are employed to address this problem. Both pickup and delivery services are considered by Luo et al. (2022), who consider paired services in which the parcels picked up from one customer must be delivered to another on the same route, and propose an iterated local search algorithm for larger instances. Meng et al. (2023) introduce a multi-visit VRP-D with pickup and delivery that allows a truck to decide whether to wait at the launch node or move to the next location for drone retrieval, allowing for more flexible operations to reduce the energy consumption of drones. They propose a two-stage approach for solving problems involving 100 customers.

2.2. VRP-D with time windows

Adding time windows to the VRP-D (VRPDTW) is a natural extension. Di Puglia Pugliese & Guerriero (2017) study this problem with hard time windows. Their model could be solved using the CPLEX solver with up to 10 customers. Subsequently, Coindreau et al. (2021) and Kuo et al. (2022) develop heuristic approaches to address similar problems. Wang et al. (2022) analyse a VRP-D with hard time windows and time-dependent road travel times. They integrate the congestion cost into the cost calculation for trucks and develop an iterated local search heuristic to minimise the total cost.

Some researchers consider multiple visits during each drone flight. For example, Li et al. (2020) extend the VRPDTW with multiple visits in a drone trip and assume that each truck can deliver and dispatch drones at customer locations; however, the truck must wait there until all its drones return. Masmoudi et al. (2022) allow drones to return to any truck to maximise profits. They develop an adaptive multi-start simulated annealing algorithm to address instances with up to 200 customers. Then, Yin et al. (2023) develop an enhanced branch-and-price-and-cut (BPC) algorithm to solve the multi-visit VRPDTW optimally. The BPC integrates a bounded bidirectional labelling algorithm and efficient enhancement strategies, which can solve instances with up to 45 customers within 3 hours. Meng et al. (2024) study a combined routing problem in which drones are launched and retrieved only from a pool of designated stations. In addition to normal pickup and delivery services, they also consider paired services and solve the problem using a customised ALNS heuristic.

2.3. VRP-D with stochastic truck travel times

Despite the extensive VRP-D literature highlighted above, most studies concentrate on deterministic problems. To the best of our knowledge, only 2 studies considered stochastic truck travel times in combined truck-drone delivery problems. Apart from Yang et al. (2023), discussed above, Liu et al. (2022) extend the basic FSTSP to stochastic travel times, where a single vehicle pair is employed to serve all customers to minimise the total delivery time. They model the problem as a multi-agent Markov decision process to capture the uncertainty and develop a reinforcement learning algorithm to solve it.

A brief summary of the most relevant literature is listed in Table 1.

3. The model

3.1. Problem description

A set of homogeneous truck-drone vehicle pairs, starting at and returning to a single depot, are deployed to make deliveries to customers, each of whom must be served exactly once by either a truck or drone. Some customers, called truck-only customers, must be served by trucks because of practical requirements, such as overweight parcels or personal preferences. The others are drone-eligible customers, who can be served by trucks or drones. Each customer has a soft time window; that is, starting service before or after the time window is allowed but incurs a penalty for either earliness or lateness. Moreover, all vehicles must return

Table 1: Summary of the most relevant studies.

Reference	K	TW	ST	TC	MV/SV	Syn
Wang et al. (2017); Sacramento et al. (2019); Zhen et al. (2023)	m	×	×	√	1	√
Schermer et al. (2019); Kitjacharoenchai et al. (2019); Tamke & Buscher (2021)	m	×	×	×	1	√
Rave et al. (2023)	m	×	×	√	1	√
Zhou et al. (2023)	m	×	×	√	1	×
Kloster et al. (2023)	m	×	×	×	1	×
Wang & Sheu (2019); Luo et al. (2022); Kitjacharoenchai et al. (2020)	m	×	×	√	m	√
Kyriakakis et al. (2022)	m	×	×	√	m	×
Meng et al. (2023)	m	×	×	√	m	√
Di Puglia Pugliese & Guerriero (2017); Kuo et al. (2022); Wang et al. (2022)	m	hard	×	√	1	√
Coindreau et al. (2021)	m	hard	×	×	1	√
Li et al. (2020)	m	hard	×	√	m	×
Masmoudi et al. (2022); Yin et al. (2023)	m	hard	×	√	m	√
Liu et al. (2022)	1	×	√	×	1	√
Yang et al. (2023)	1	hard	√	×	1	×
Meng et al. (2024)	m	hard	×	√	m	√
This work	**m**	**soft**	√	√	**m**	√

K: number of trucks used. m: multiple. TW: time windows (hard: hard time windows; soft: soft time windows). ST: stochastic travel times. TC: truck capacity. MV/SV: multiple visits/ single visit of drones. Syn: whether trucks and drones are synchronously serving customers, i.e., trucks can move forward to serve other customers after launching drones.

to the depot before a pre-specific return time, which could be the maximum working hours; otherwise, a penalty (e.g., overtime pay) is incurred for any lateness. A drone can serve multiple customers in a flight as long as its payload and battery capacities permit, and it can be launched from its truck and retrieved at different locations multiple times along the truck route. When a truck and its drone reconvene at a node, the one arriving first must wait for the other. We assume that drone aprons are available for landing at customer nodes; thus, the energy consumption of the drones while waiting is negligible. A drone can be launched and retrieved at most once at each customer location. Once a drone is retrieved, its battery is swapped with a fully charged one. Trucks/drones start service immediately upon arrival at a customer node. However, trucks are allowed to wait at each node (including the depot) before departing for the next node. The operational procedure for a node visited by a truck is illustrated in Figure 2. *Note* that when the truck travel time is deterministic along an arc, waiting after service at the outbound node is equivalent to waiting before service at the inbound node.

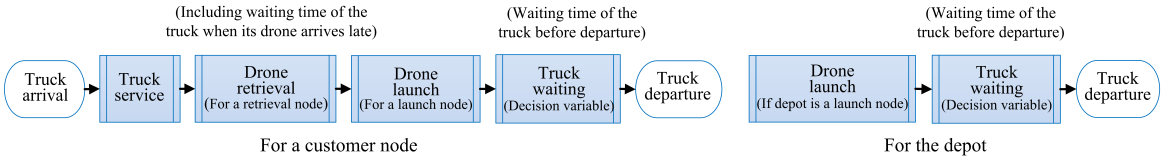

Figure 2: The operational procedure at a node visited by trucks.

We consider stochastic truck travel times and deterministic drone travel times. Our objective is to determine the optimal routes/flights and truck waiting times that minimise the total cost, which includes the operational cost (travelling cost of trucks and energy cost of drones), fixed cost for deployed vehicle pairs, and expected penalties for the violation of time windows. We refer to this problem as the 'multi-visit drone-assisted routing problem with soft time windows and stochastic truck travel times' (mDRP-TS) henceforth. Figure 3 illustrates an example of an mDRP-TS in which two vehicle pairs serve 20 customers.

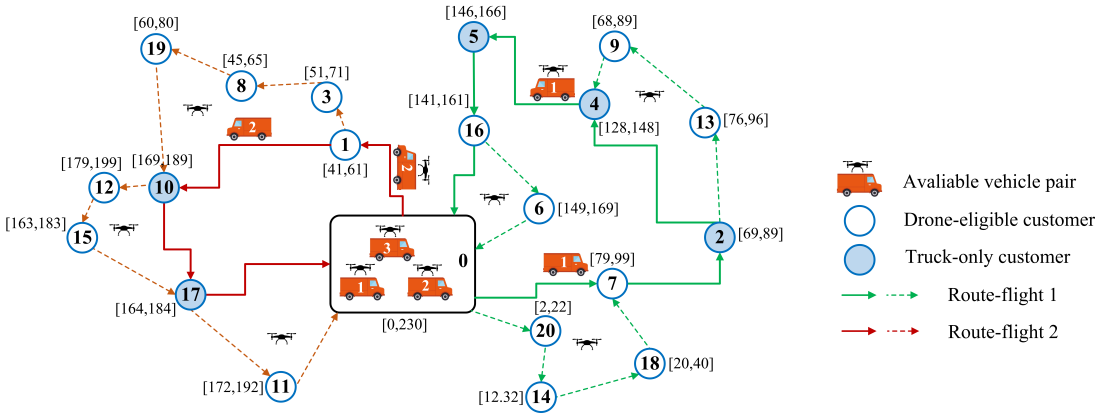

Figure 3: An illustrative mDRP-TS example.

3.2. The stochastic model

The mDRP-TS problem can be formulated as a two-stage stochastic model. The first stage determines the number of deployed truck-drone pairs, their routes, and flights. These decisions are made before vehicles depart from the depot. The second stage concerns the optimal waiting time at each node to minimise the time window violations along each itinerary, which takes place after the vehicles have left the depot and more information on the truck travel times becomes available. Before presenting the model, we introduce important notation.

A fleet of K truck-drone pairs (indexed by k) is available to serve customers located on a directed graph $G(N, A)$, where N is the set of nodes and A is the set of arcs. $N = \{0, 1, ..., n+1\}$ includes customer set $N_c = \{1, ..., n\}$, starting depot (0), and ending depot $(n+1)$. Let N_d be the set of drone-eligible customers; then, $N_c \setminus N_d$ is the set of truck-only customers. Moreover, the sets of outbound and inbound nodes are denoted by $N_s = N \setminus \{n+1\}$ and $N_e = N \setminus \{0\}$, respectively. Thus, $A = \{(i, j) | i \in N_s, j \in N_e, i \neq j\}$. Denote by r_{ij} and $\hat{r}_{ij}$ the travel distances for trucks and drones on arc (i, j) using Manhattan and Euclidean metrics, respectively (Yang et al., 2023). Each arc (i, j) has a stochastic truck travel time induced by stochastic truck speed $\mathbf{v}_{ij}$. We have $\mathbf{v}_{ij} \sim F_{ij}$, where F_{ij} is the corresponding distribution function that is known or can be estimated at the time of planning. We assume that all $\mathbf{v}_{ij}$ values are independent. The constant drone speed is denoted as $\hat{v}$. Each node i has a known demand d_i and a soft time window $[e_i, l_i]$. A time window $[0, l_{max}]$ is associated with the depot, where l_{max} is the time at which all vehicles must return to the depot. For brevity, we list in Table 2 the notation used in the first-stage problem.

Table 2: Notation for the first-stage problem.

Sets	
N_c	Set of customers, $N_c = \{1, 2, ..., n\}$
N	Set of nodes, $N = \{0, 1, ..., n+1\}$
N_s	Set of nodes for the start of an arc, $N_s = \{0\} \cup N_c = \{0, 1, ..., n\}$
N_e	Set of nodes for the end of an arc, $N_e = N_c \cup \{n+1\} = \{1, ..., n+1\}$
N_d	Subset of drone-eligible customers
A	Set of arcs, $A = \{(i, j) \| i \in N_s, j \in N_e, i \neq j\}$
K	Set of vehicle pairs, each indexed by $k \in K$
Parameters	
$\mathbf{v}_{ij}$	Stochastic truck speed on arc (i, j) (unit: km/h)
F_{ij}	Distribution function of truck speed $\mathbf{v}_{ij}$
$\hat{v}$	Constant speed of each drone (unit: km/h)
c^0	Fixed cost of deploying a vehicle pair (unit: \$)
c	Cost for operating truck k per unit distance (unit: \$/km)
$\hat{c}$	Cost for operating drone k per unit time (unit: \$/min)
r_{ij}	Manhattan distance of arc (i, j) (unit: km)
$\hat{r}_{ij}$	Euclidean distance of arc (i, j) (unit: km)
W	Available capacity of each truck (unit: kg)
$\hat{W}$	Capacity of each drone (unit: kg)
d_i	Demand of customer $i \in N_c$ (unit: kg)
B	Battery capacity of each drone (unit: min)
s_i	Service time at customer $i \in N_c$ by trucks (unit: min)
$\hat{s}_i$	Service time at customer $i \in N_d$ by drones (unit: min)
M	A sufficiently large number
Decision variables	
$x_{ij}^k \in \{0, 1\}$	1 if truck k traverses arc (i, j) with its drone on board and 0 otherwise
$y_{ij}^k \in \{0, 1\}$	1 if truck k traverses arc (i, j) independently and 0 otherwise
$\hat{y}_{ij}^k \in \{0, 1\}$	1 if drone k traverses arc (i, j) independently and 0 otherwise
Auxiliary variables	
$z_i^k \in \{0, 1\}$	1 if customer i is served by drone k and 0 otherwise
$u^k \in \{0, 1\}$	1 if vehicle pair k is deployed and 0 otherwise
$e_i^k \in [0, B]$	Cumulative energy consumed by drone k upon leaving from node i in a flight
$w_i^k \in [0, \hat{W}]$	Payload of drone k after visiting node i
$\lambda_i^k \in \mathbb{Z}^+$	Relative position of customer i in the visit sequence of truck/drone k, $1 \leq \lambda_i^k \leq n$

For convenience, we denote the vector of routing decision variables as $\boldsymbol{\theta} = \{x_{ij}^k, y_{ij}^k, \hat{y}_{ij}^k, \forall k \in K, (i, j) \in A\}$ and the vector of stochastic truck speeds for all arcs by $\mathbf{v}$. We now present the first-stage formulation:

$$min \quad c \sum_{k \in K} \sum_{(i,j) \in A} r_{ij}(x_{ij}^k + y_{ij}^k) + \hat{c} \left(\sum_{k \in K} \sum_{(i,j) \in A} \hat{y}_{ij}^k \frac{\hat{r}_{ij}}{\hat{v}} + \sum_{k \in K} \sum_{i \in N_d} \hat{s}_i z_i^k \right) + c^0 \sum_{k \in K} u^k + E[Q(\boldsymbol{\theta}, \mathbf{v})], \tag{1}$$

subject to:

Routing constraints of trucks and drones:

$$\sum_{i \in N_e} (x_{0i}^k + y_{0i}^k) = \sum_{i \in N_s} (x_{i,n+1}^k + y_{i,n+1}^k) = 1, k \in K \tag{2a}$$

$$\sum_{i\in N_e}(x_{0i}^k+\hat{y}_{0i}^k)=\sum_{i\in N_s}(x_{i,n+1}^k+\hat{y}_{i,n+1}^k)=1, k\in K \tag{2b}$$

$$\sum_{i\in N_s}(x_{ij}^k+y_{ij}^k)=\sum_{i\in N_e}(x_{ji}^k+y_{ji}^k), j\in N_c, k\in K \tag{3a}$$

$$\sum_{i\in N_s}(x_{ij}^k+\hat{y}_{ij}^k)=\sum_{i\in N_e}(x_{ji}^k+\hat{y}_{ji}^k), j\in N_c, k\in K \tag{3b}$$

$$2*z_i^k\le\sum_{j\in N_s}\hat{y}_{ji}^k+\sum_{j\in N_e}\hat{y}_{ij}^k, i\in N_d, k\in K \tag{4}$$

$$\sum_{k\in K}\sum_{i\in N_s}(x_{ij}^k+y_{ij}^k)+\sum_{k\in K}z_j^k=1, j\in N_c \tag{5a}$$

$$\sum_{k\in K}\sum_{i\in N_s}(x_{ij}^k+y_{ij}^k)=1, j\in N_c\setminus N_d \tag{5b}$$

$$\sum_{j\in N_e}\hat{y}_{ij}^k-z_i^k\le\sum_{j\in N_e}y_{ij}^k, i\in N_c, k\in K \tag{6a}$$

$$\sum_{j\in N_s}\hat{y}_{ji}^k-z_i^k\le\sum_{j\in N_s}y_{ji}^k, i\in N_c, k\in K \tag{6b}$$

$$\sum_{k\in K}\sum_{j\in N_e}\hat{y}_{ij}^k\le 1, i\in N_c \tag{7a}$$

$$\sum_{k\in K}\sum_{j\in N_s}\hat{y}_{ji}^k\le 1, j\in N_c \tag{7b}$$

$$\sum_{j\in N_s}\sum_{i\in N_c}(x_{ji}^k+y_{ji}^k)+\sum_{i\in N_d}z_i^k\le n\cdot u^k, k\in K \tag{8}$$

$$\lambda_i^k-\lambda_j^k+n(x_{ij}^k+y_{ij}^k)\le n-1, i,j\in N_c, k\in K \tag{9a}$$

$$\lambda_i^k-\lambda_j^k+n(x_{ij}^k+\hat{y}_{ij}^k)\le n-1, i,j\in N_c, k\in K \tag{9b}$$

Load capacity constraint of trucks:

$$\sum_{i\in N_s}\sum_{j\in N_c}(x_{ij}^k+y_{ij}^k)d_j+\sum_{i\in N_d}z_i^k d_i\le W, k\in K \tag{10}$$

Demand flow constraints of drones:

$$w_i^k-d_j-M(2-\hat{y}_{ij}^k-z_j^k)\le w_j^k\le w_i^k-d_j+M(2-\hat{y}_{ij}^k-z_j^k), i\in N_s, j\in N_d, k\in K \tag{11a}$$

$$w_j^k\le M(2-\sum_{j'\in N_s}y_{j'i}^k-\hat{y}_{ji}^k), j\in N_d, i\in N_e, k\in K \tag{11b}$$

Energy consumption constraints of drones:

$$e_i^k\le B(2-\sum_{j\in N_e}y_{ij}^k-\sum_{j\in N_d}\hat{y}_{ij}^k), i\in N_s, k\in K \tag{12a}$$

$$e_i^k+\frac{\hat{r}_{ij}}{\hat{v}}+\hat{s}_j-M(2-\hat{y}_{ij}^k-z_j^k)\le e_j^k\le e_i^k+\frac{\hat{r}_{ij}}{\hat{v}}+\hat{s}_j+M(2-\hat{y}_{ij}^k-z_j^k), i\in N_s, j\in N_d, k\in K \tag{12b}$$

$$e_i^k+\frac{\hat{r}_{ij}}{\hat{v}}-M(2-\hat{y}_{ij}^k-\sum_{i'\in N_s}y_{i'j}^k)\le B, i\in N_d, j\in N_e, k\in K. \tag{12c}$$

Objective (1) is to minimise the total cost, including expected time window violation penalties. The first term denotes the transportation cost of trucks, the second term denotes the energy consumption cost of drones, the third term captures the fixed cost of deploying vehicle pairs, and the final term $E[Q(\boldsymbol{\theta},\mathbf{v})]$ denotes the expected value of the objective function of the second-stage problem. Constraints $(2a)$ and $(2b)$ ensure that each vehicle pair departs and returns to a single depot exactly once. Constraints $(3a)$ and $(3b)$ ensure the flow balance of all the trucks and drones at each node. Constraint (4) ensures that if a customer is served by drone k, it must visit this node. Constraint $(5a)$ states that each customer can be served only once by either a truck or drone. Constraint $(5b)$ ensures that truck-only customers are served by trucks. Constraints $(6a)$ and $(6b)$ ensure that the nodes used for drone launch and retrieval must be

visited by paired trucks. Constraints (7a) and (7b) ensure that drones can only depart from and arrive at a customer node once at most. Constraint (8) states that if a vehicle pair k serves a customer, it must be deployed. Constraints (9a) and (9b) are subtour elimination constraints. If customer j is visited directly after customer i on the route of vehicle pair k, then $\lambda_i^k + 1 \leq \lambda_j^k$. Constraint (10) ensures that all the parcels assigned to each vehicle pair cannot exceed the available capacity of trucks. Note that the capacity constraint of drones is imposed implicitly by the definition of the variable w_i^k. Constraint (11a) tracks the payload of drones upon leaving each node they serve. Constraint (11b) ensures that the payload of drones upon retrieval is zero. Constraint (12a) ensures that the energy consumption of drones upon launch is zero; that is, their batteries are fully charged. Constraint (12b) tracks the energy consumption of drones when leaving the customer they serve. Constraint (12c) ensures that the total energy consumption of a drone during a flight does not exceed battery capacity.

For a given routing decision $\boldsymbol{\theta}$ and a realisation of truck speeds $\mathbf{v}^\xi$, the second-stage problem optimises the waiting times of trucks to minimise time window violation penalties. Table 3 presents all the notation used in the second-stage formulation.

Table 3: Notation for the second-stage problem.

Parameters	
ξ	A scenario or realisation of truck speeds on all arcs
v_{ij}^ξ	Truck speed on arc (i,j) in Scenario ξ (unit: km/h)
c^e	Unit penalty cost of earliness when services start earlier than the lower bounds of time windows (unit: \$/min)
c^l	Unit penalty cost for lateness when services start later than the upper bounds of time windows (unit: \$/min)
c^k	Unit penalty cost for lateness when vehicle pair k returning to the depot late (unit: \$/min). We assume $c^k > c^l > c^e$
$[e_i, l_i]$	Time window for node i (unit: min)
s^l	Required time for drone launch (unit: min)
s^r	Required time for drone retrieval (unit: min)
Decision variable	
$\omega_i^{k,\xi} \geq 0$	Waiting time of truck k at node i in Scenario ξ
Auxiliary variables	
$\delta_i^{-,\xi} \geq 0$	Earliness of the service start time at node i before the lower bound of its time window in Scenario ξ
$\delta_i^{+,\xi} \geq 0$	Lateness of the service start time at node i after the upper bound of its time window in Scenario ξ
$\delta^{k,\xi} \geq 0$	Lateness of vehicle pair k returning to the depot in Scenario ξ
$\tau_i^{k,\xi} \geq 0$	Arrival time of truck k at node i in Scenario ξ
$\hat{\tau}_i^{k,\xi} \geq 0$	Arrival time of drone k at node i in Scenario ξ
$\varphi_i^\xi \geq 0$	Service start time at node i in Scenario ξ
$\bar{\tau}_i^{k,\xi} \geq 0$	The time at which both vehicles in pair k are ready for the retrieval at node i in Scenario ξ; i.e., $max\{\tau_i^{k,\xi} + s_i, \hat{\tau}_i^{k,\xi}\}$
$\alpha_i^\xi \in \{0,1\}$	1 if node i is served before its lower bound time window in Scenario ξ and 0 otherwise
$\beta_i^\xi \in \{0,1\}$	1 if node i is served after its upper bound time window in Scenario ξ and 0 otherwise
$\gamma_i^{k,\xi} \in \{0,1\}$	1 if truck k arrives at retrieval node i later than its drone k in Scenario ξ and 0 otherwise

The mathematical model of the second-stage problem is presented below:

$$Q(\boldsymbol{\theta}, \mathbf{v}^\xi) = min \sum_{i \in N_c} (c^e \delta_i^{-,\xi} + c^l \delta_i^{+,\xi}) + \sum_{k \in K} c^k \delta^{k,\xi}, \tag{13}$$

subject to:

Arrival time calculation of drones:

$$\varphi_i^\xi + \hat{s}_i + \frac{\hat{r}_{ij}}{\hat{v}} - M(2 - \hat{y}_{ij}^k - z_i^k) \leq \hat{\tau}_j^{k,\xi} \leq \varphi_i^\xi + \hat{s}_i + \frac{\hat{r}_{ij}}{\hat{v}} + M(2 - \hat{y}_{ij}^k - z_i^k), i \in N_d, j \in N_e, k \in K \tag{14}$$

$$\bar{\tau}_i^{k,\xi} + s^r + s^l + \frac{\hat{r}_{ij}}{\hat{v}} - M(3 - \sum_{j \in N_e} y_{ij}^k - \hat{y}_{ij}^k - \sum_{j \in N_d} \hat{y}_{ji}^k) \leq \hat{\tau}_j^{k,\xi}$$

$$\leq \bar{\tau}_i^{k,\xi} + s^r + s^l + \frac{\hat{r}_{ij}}{\hat{v}} + M(3 - \sum_{j \in N_e} y_{ij}^k - \hat{y}_{ij}^k - \sum_{j \in N_d} \hat{y}_{ji}^k), i \in N_s, j \in N_d, k \in K \tag{15a}$$

$$\varphi_i^\xi + s_i + s^l + \frac{\hat{r}_{ij}}{\hat{v}} - M(2 - \sum_{j \in N_e} y_{ij}^k - \hat{y}_{ij}^k + \sum_{j \in N_d} \hat{y}_{ji}^k) \leq \hat{\tau}_j^{k,\xi}$$

$$\leq \varphi_i^\xi + s_i + s^l + \frac{\hat{r}_{ij}}{\hat{v}} + M(2 - \sum_{j \in N_e} y_{ij}^k - \hat{y}_{ij}^k + \sum_{j \in N_d} \hat{y}_{ji}^k), i \in N_s, j \in N_d, k \in K \tag{15b}$$

Arrival time calculation of trucks:

$$\bar{\tau}_i^{k,\xi} + s^r + s^l \sum_{j \in N_d} \hat{y}_{ij}^k + \omega_i^{k,\xi} + \frac{r_{ij}}{v_{ij}^\xi} - M(2 - x_{ij}^k - y_{ij}^k - \sum_{j \in N_d} \hat{y}_{ji}^k) \leq \tau_j^{k,\xi}$$

$$\leq \bar{\tau}_i^{k,\xi} + s^r + s^l \sum_{j \in N_d} \hat{y}_{ij}^k + \omega_i^{k,\xi} + \frac{r_{ij}}{v_{ij}^\xi} + M(2 - x_{ij}^k - y_{ij}^k - \sum_{j \in N_d} \hat{y}_{ji}^k), (i,j) \in A, k \in K \quad (16a)$$

$$\varphi_i^\xi + s_i + s^l \sum_{j \in N_d} \hat{y}_{ij}^k + \omega_i^{k,\xi} + \frac{r_{ij}}{v_{ij}^\xi} - M(1 - x_{ij}^k - y_{ij}^k + \sum_{j \in N_d} \hat{y}_{ji}^k) \leq \tau_j^{k,\xi}$$

$$\leq \varphi_i^\xi + s_i + s^l \sum_{j \in N_d} \hat{y}_{ij}^k + \omega_i^{k,\xi} + \frac{r_{ij}}{v_{ij}^\xi} + M(1 - x_{ij}^k - y_{ij}^k + \sum_{j \in N_d} \hat{y}_{ji}^k), (i,j) \in A, k \in K \quad (16b)$$

Service start time constraints:

$$\hat{\tau}_i^{k,\xi} - M(1 - z_i^k) \leq \varphi_i^\xi \leq \hat{\tau}_i^{k,\xi} + M(1 - z_i^k), i \in N_d, k \in K \quad (17a)$$

$$\tau_i^{k,\xi} - M(1 - \sum_{j \in N_s} x_{ji}^k - \sum_{j \in N_s} y_{ji}^k) \leq \varphi_i^\xi \leq \tau_i^{k,\xi} + M(1 - \sum_{j \in N_s} x_{ji}^k - \sum_{j \in N_s} y_{ji}^k), i \in N_c, k \in K \quad (17b)$$

Constraints of the lateness of returning to the depot:

$$\delta^{k,\xi} \geq \hat{\tau}_{n+1}^{k,\xi} - l_{max} - M(1 - \sum_{i \in N_d} \hat{y}_{i,n+1}^k), k \in K \quad (18a)$$

$$\delta^{k,\xi} \geq \tau_{n+1}^{k,\xi} - l_{max} - M(1 - \sum_{i \in N_c} x_{i,n+1}^k - \sum_{i \in N_c} y_{i,n+1}^k), k \in K \quad (18b)$$

Earliness and lateness calculation:

$$\varphi_i^\xi - e_i \leq M(1 - \alpha_i^\xi), i \in N_c \quad (19a)$$

$$\delta_i^{-,\xi} \leq M\alpha_i^\xi, i \in N_c \quad (19b)$$

$$e_i - \varphi_i^\xi \leq \delta_i^{-,\xi} \leq e_i - \varphi_i^\xi + M(1 - \alpha_i^\xi), i \in N_c \quad (19c)$$

$$l_i - \varphi_i^\xi \leq M(1 - \beta_i^\xi), i \in N_c \quad (19d)$$

$$\delta_i^{+,\xi} \leq M\beta_i^\xi, i \in N_c \quad (19e)$$

$$\varphi_i^\xi - l_i \leq \delta_i^{+,\xi} \leq \varphi_i^\xi - l_i + M(1 - \beta_i^\xi), i \in N_c \quad (19f)$$

$$\alpha_i^\xi + \beta_i^\xi \leq 1, i \in N_c \quad (19g)$$

Calculation of the time ready for drone retrieval:

$$\hat{\tau}_i^{k,\xi} - \tau_i^{k,\xi} - s_i \leq M(1 - \gamma_i^{k,\xi}), i \in N_c, k \in K \quad (20a)$$

$$\hat{\tau}_i^{k,\xi} \leq \bar{\tau}_i^{k,\xi} \leq \hat{\tau}_i^{k,\xi} + M\gamma_i^{k,\xi}, i \in N_c, k \in K \quad (20b)$$

$$\tau_i^{k,\xi} + s_i \leq \bar{\tau}_i^{k,\xi} \leq \tau_i^{k,\xi} + s_i + M(1 - \gamma_i^{k,\xi}), i \in N_c, k \in K. \quad (20c)$$

The objective function $Q(\boldsymbol{\theta}, \mathbf{v}^\xi)$ minimises the penalties for violating the time windows in Scenario ξ given the routing decisions $\boldsymbol{\theta}$. Constraint (14) calculates the arrival time of each drone at a node if the preceding node is served by it. Otherwise, the arrival time is calculated using the constraints (15a) and (15b), depending on whether the preceding node is a retrieval node. Similarly, Constraints (16a) and (16b) calculate the arrival time of each truck at a node depending on whether the preceding node visited by the truck is a retrieval node. Constraints (17a) and (17b) ensure that the service starts immediately upon the vehicle arrival at each node. Constraints (18a) and (18b) calculate the lateness of each vehicle pair that returns to the depot. Constraint (19a)–(19c) calculates the earliness of each customer node, whereas constraint (19d)–(19f) calculates the lateness of each customer node. Constraint (19g) ensures a mutually exclusive relationship between the earliness and lateness at each node. Constraints (20a)–(20c) are the linearised formulae to calculate the time when each vehicle pair is ready for retrieval, that is, $\bar{\tau}_i^{k,\xi} = max\{\tau_i^{k,\xi} + s_i, \hat{\tau}_i^{k,\xi}\}$.

3.3. Sample average approximation

The main challenge in solving the two-stage stochastic model is evaluating $E[Q(\boldsymbol{\theta}, \mathbf{v})]$. The distribution function F_{ij} can be either continuous or discrete. Even for the latter case, which is simpler, the exact evaluation of $E[Q(\boldsymbol{\theta}, \mathbf{v})]$ is practically impossible when the number of realisations of $\mathbf{v}$ is large. This is the case in this study, even for a small network. A common technique for addressing this computational

challenge is a sample average approximation framework. A sample of $|\Xi|$ realisations of $\mathbf{v}$ is firstly generated, and then the sample averaged function $\frac{1}{|\Xi|}\sum_{\xi\in\Xi} Q(\boldsymbol{\theta}, \mathbf{v}^{\xi})$ is used to approximate $E[Q(\boldsymbol{\theta}, \mathbf{v})]$. By adding the second-stage constraints (14)–(20) for each realisation $\mathbf{v}^{\xi}$ to the first-stage formulation, we obtain the following large-scale MIP. Note that subtour elimination constraints (9a) and (9b) in the first-stage model are redundant.

$$\begin{aligned} min \quad & c\sum_{k\in K}\sum_{(i,j)\in A} r_{ij}(x_{ij}^k + y_{ij}^k) + \hat{c}\left(\sum_{k\in K}\sum_{(i,j)\in A}\hat{y}_{ij}^k\frac{\hat{r}_{ij}}{\hat{v}} + \sum_{k\in K}\sum_{i\in N_d}\hat{s}_i z_i^k\right) + c^0\sum_{k\in K}u^k \\ & + \frac{1}{|\Xi|}\sum_{\xi\in\Xi}\left(\sum_{i\in N_c}(c^e\delta_i^{-,\xi} + c^l\delta_i^{+,\xi}) + \sum_{k\in K}c^k\delta^{k,\xi}\right) \end{aligned} \tag{21}$$

subject to:
(2)-(8), (10)-(12), and (14)-(20), $\forall\xi\in\Xi$.

To reduce the approximation error and produce solutions converging to the true optimality a large sample needs to be generated in the above model, adding extra computational complexity to the already NP-hard combinatorial optimisation problem. To this end, we develop an efficient heuristic approach that incorporates the SAA framework, as described in the next section.

4. A hybrid metaheuristic with SAA

Considering that the model will become too large to solve when the sample size grows, we develop a hybrid metaheuristic (called hybrid iterated greedy metaheuristic based on simulated annealing, HMSA) with SAA (HMSA-S) for the mDRP-TS. The framework is shown in Algorithm 1. Specifically, a greedy-based heuristic is first applied to generate a feasible initial solution (line 1), which is then improved by the HMSA with the mean truck speeds to obtain the best solution S in the deterministic setting (line 2). Finally, the HMSA integrating SAA (HMSA-SAA) is applied to further optimise solution S in the stochastic setting (line 3).

Algorithm 1. *Outline of the hybrid metaheuristic integrating SAA(HMSA-S)*

Input: M, number of batches; H, sample size in each batch; H', a large sample size for candidate solution estimation; $T_0 = (T_{01}, T_{02})$, initial temperature; $\mu = (\mu_1, \mu_2)$, cooling rate; $iterMax = (iterMax_1, iterMax_2)$, number of iterations; $T_f = (T_{f1}, T_{f2})$, floor temperature
Output: the best solution s^{best}
1: $s^{initial} \leftarrow$ *Heuristic for initial solution construction()*
2: $S \leftarrow$ HMSA($s^{initial}$, T_{01}, μ_1, $iterMax_1$, T_{f1}, 1) using the mean truck speeds
3: $s^{best} \leftarrow$ HMSA-SAA(S, M, H, H', T_{02}, μ_2, $iterMax_2$, T_{f2})
4: **return** s^{best}

4.1. Heuristic for initial solution construction

The heuristic constructs an initial solution in two phases. It starts with a *GetTruckonly* procedure to construct a truck-only solution to serve all customers. Then, another procedure *SeqConstFlight* is applied to reassign some customers to drones. Note that the mean truck speeds are used and all procedures below are performed only if the solution is feasible.

The *GetTruckonly* procedure starts with a greedy construction of an initial truck-only solution with the constraint of truck load capacity. Specifically, starting from the depot as the currently visited node s_{cur}, the unassigned customer i that has the minimum cost $r_{s_{cur},i} * c + c^e * max\{0, e_i - \tau_i^k\} + c^l * max\{0, \tau_i^k - l_i\}$ is added to the route of the current truck. After that the newly added customer node becomes the new s_{cur} and the process repeats to add more customers, one-by-one as long as the truck capacity allows. Otherwise, a new route is started from the depot. The above steps are repeated until all customers are assigned. Then the obtained routes are improved with a basic SA algorithm. Note that in the initial solution we assume that there is no additional waiting time at each node. The pseudocode for the *GetTruckonly* procedure is presented in Supplementary Materials A (as Algorithm I). The *SeqConstFlight* procedure is the same as the split and insertion operations proposed in Meng et al. (2023).

4.2. HMSA

The HMSA is implemented with a set of perturbation procedures to achieve diversification and a set of local searches embedded in simulated annealing to iteratively obtain a better solution. The local searches include intra-route and inter-route procedures that work within one route and on different routes, respectively. In each iteration, the 'intra' and then 'inter' local searches are applied to generate candidate solutions. For each of the local searches, each of the operators is invoked to generate one new solution, and the best-feasible one is chosen as the candidate solution. The pseudocode is shown in Supplementary Materials B (as Algorithm II).

Perturbation procedure

Based on the problem features, we design a set of customised perturbation operators to shake the current solution. For the removed nodes in the first four operators, we apply a greedy insertion to each of them, i.e., relocate each randomly-selected node from the removal list to the place that gives the best objective value until a new complete solution is constructed. Moreover, if all attempts fail to obtain a feasible relocation for a node, a new truck or flight route, whichever is better, is created to serve it. Note that if a launch/retrieval node is removed, the affected flights also need to be removed.

- **Random removal** randomly takes out $\lceil 10\% * n \rceil$ customers from the solution.
- **Worst removal** iteratively removes the node whose removal produces the greatest cost reduction from the current solution, until $\lceil 10\% * n \rceil$ customers are removed.
- **Sweep removal** removes from the current solution a randomly-selected node i and the first $max\{1, \lceil 10\% * n - 1 \rceil\}$ nodes with the smallest acute angles from it. The angle between two nodes are defined by the two rays from the depot to each node.
- **Related removal** removes customers with similarities. A randomly-selected seed i is first deleted, then the node j with the smallest relatedness $R(i, j)$ becomes the new seed and is deleted as well. We iteratively repeat above steps until $\lceil 10\% * n \rceil$ nodes are removed. The relatedness is defined by $R(i, j) = w_1|d_i - d_j| + w_2(|e_i - e_j| + |l_i - l_j|) + w_3(r_{ij} + \hat{r}_{ij}) + w_4 * h_{i,j}$, $w_1, w_2, w_3, w_4 \in [0, 1]$, where $h_{i,j} \in \{0, 1\}$. $h_{i,j} = 1$ indicates that nodes i and j are served by the same vehicle pair and 0 otherwise. The values of (w_1, w_2, w_3, w_4) are tuned to (0.32, 0.51, 0.46, 0.24) via preliminary experiments.
- **String relocation** first randomly removes 1, 2 or 3 strings from the solution, then randomly relocates each into the original or other routes. Note that if a split node hosts drone launch or retrieval, it will be retained in the route and the affected flights in the deleted strings need to find new launch and/or retrieval node(s) around the new location, as shown in Figure 4. Moreover, a removed string that contains flights can only be inserted into arcs traversed by both trucks and drones.

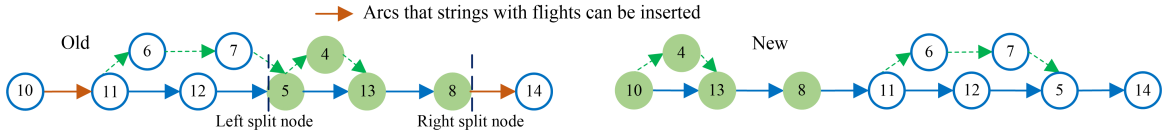

Figure 4: String relocation operator.

- **Multi-exchange** conducts $max\{2, \lceil 15\% * n \rceil\}$ random exchanges between two nodes from the same or different routes. The exchanged nodes can be any customers served by trucks or drones.

Intra-route local search

Seven neighborhood operators are used to generate neighbouring solutions within each route.

- **Drone node exchange** randomly exchanges two nodes served by the drone.
- **Truck node exchange** randomly exchanges two nodes served by the truck. Note that if a launch/retrieval node is involved, the locations in the affected flights also need to be exchanged.
- **General exchange** is similar to truck node exchange, but randomly exchanging two drone-eligible nodes that are served by a truck and a drone respectively.
- **Truck route insertion** randomly selects a node served in a drone flight and greedily inserts it into a location i on the truck route. Note that the launch and retrieval nodes of the flights scheduled after

location i need to be updated accordingly (i.e., update procedure), as shown in Figure 5, which is also applied to the next operator.

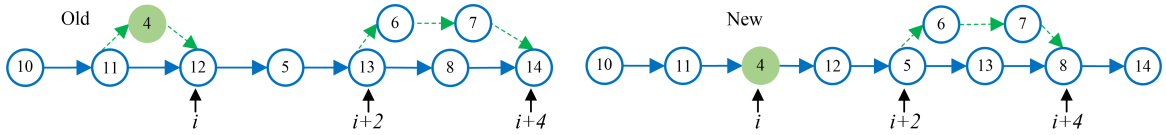

Figure 5: Truck route insertion operator.

- **Same route insertion** randomly selects a truck-served node and greedily inserts it to another position on the truck route. Note that before performing the update procedure, if the moved node serves as a launch/retrieval node, the affected flights need to (randomly) identify new launch/retrieval nodes.
- **Flight change** randomly selects a drone-served node and greedily reassigns it to the original or another flight, then performs a local search for the affected flights for better launch and retrieval positions.
- **Flight construction** greedily constructs a new flight for a truck-served node. Note that if a launch/retrieval node is involved, the affected flights need to identify new launch/retrieval nodes.

Inter-route local search

If there are more than one route in the solution, inter-route local search is performed using the operators below.

- **Node insertion** first randomly selects two routes, then randomly chooses a node from one route and greedily relocates it to the other. The relocation can be done in one of following randomly chosen ways: inserting to the truck route, inserting to an existing flight or creating a new flight.
- **Route cut** randomly intercepts a string from the longest route to construct a new one, staring from and ending at the depot.

4.3. Truck waiting time optimisation

The HMSA involves optimising the waiting time at each node. In principle, the waiting times can be exactly optimised by the second-stage formulation in Section 3.2. However, the total computational time would be intractable as the model has to be solved for each sampled scenario in each iteration. To accelerate the solution process, we ignore the impact of waiting times on the penalty cost of drones and develop the following heuristic to optimise waiting times. For each truck, define the truck route as an ordered set $\{0, i_1, i_2, ..., i_g, ..., i_G, n+1\}$, where G is the number of customers served by the truck. Let s_{ij}^{ready} be the ready time for serving node j after travelling there from node i on the truck route. That is, if node i is a retrieval node, $s_{ij}^{ready} = \bar{\tau}_i + s^r + s^l \sum_{j' \in N_d} \hat{y}_{ij'} + \frac{r_{ij}}{v_{ij}}$. Otherwise, $s_{ij}^{ready} = \tau_i + s_i + s^l \sum_{j' \in N_d} \hat{y}_{ij'} + \frac{r_{ij}}{v_{ij}}$. We have the following results:

Lemma 1. *The optimal waiting time at the last visited customer (i.e., ω_{i_G}) on the truck route is zero, allowing the earliest arrival time at the depot.*

The proofs of Lemma 1 and the Propositions in the paper are all given in Supplementary Materials C. In view of Lemma 1, we set $\omega_{i_G} = 0$.

Proposition 1. *If $s_{ij}^{ready} \geq l_j$, $\omega_i = 0$ gives the minimum penalty cost at node j and the earliest ready times for the subsequent services, i.e., the minimum latency.*

In view of Proposition 1, if $s_{ij}^{ready} \geq l_j$, we set $\omega_i = 0$. Otherwise, we consider the following two cases.
Case 1: The truck travels from node i to a non-retrieval node j.

Proposition 2. *If $s_{ij}^{ready} \geq e_j$, $\omega_i = 0$ gives the minimum penalty cost at node j and the minimum latency. Otherwise, the optimal waiting time must take values from $[0, e_j - s_{ij}^{ready}]$, allowing the minimum latency.*

In view of Proposition 2, if $s_{ij}^{ready} \geq e_j$, we set $\omega_i = 0$. Otherwise, we find the ω_i from the range $[0, e_j - s_{ij}^{ready}]$ by the following algorithm, which includes six scenarios as shown in Figure 6. We use p to denote the next node after j in the route.

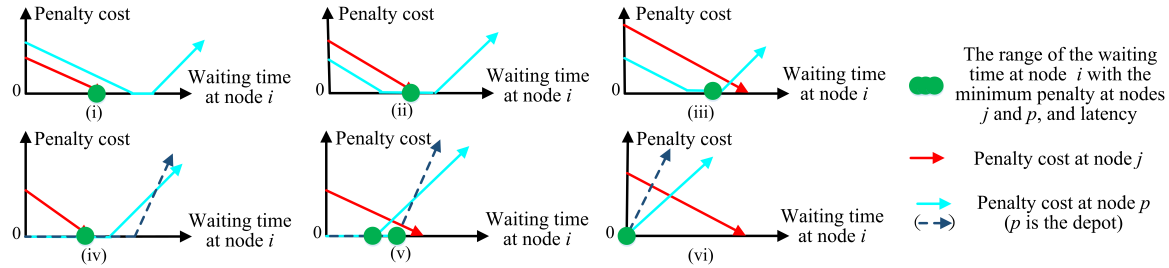

Figure 6: The penalty costs at nodes j and p in the case of $s_{ij}^{ready} < e_j$.

Algorithm 2. *Waiting time optimisation at node i (when $0 < s_{ij}^{ready} \leq e_j$, $G \geq 2$)*

Input: s_{ij}^{ready}, the ready time for serving node j
Output: the best waiting time ω_i

$s_{jp}^{ready,l} = s_{ij}^{ready} + s_j + s^l \sum_{i' \in N_d} \hat{y}_{ji'} + \frac{r_{jp}}{v_{jp}}$ # Calculate the earliest ready time for node p
$s_{jp}^{ready,u} = s_{jp}^{ready,l} - s_{ij}^{ready} + e_j$ # Calculate the latest ready time for node p
if $p \neq n+1$ **then**
  if $s_{jp}^{ready,l} > l_p$ **then**
    return $\omega_i = 0 \leftarrow$ Figure 6(vi)
  else
    if $e_p \leq s_{jp}^{ready,l} \leq l_p$ **then**
      if $s_{jp}^{ready,u} \leq l_p$ **then**
        return $\omega_i = e_j - s_{ij}^{ready} \leftarrow$ Figure 6(iv)
      else
        return $\omega_i = l_p - s_{jp}^{ready,l} \leftarrow$ Figure 6(v)
    else
      if $s_{jp}^{ready,u} \leq l_p$ **then**
        return $\omega_i = e_j - s_{ij}^{ready} \leftarrow$Figures 6(i) and 6(ii)
      else
        return $\omega_i = l_p - s_{jp}^{ready,l} \leftarrow$ Figure 6(iii)
else # $p = n+1$, node p is the depot
  $\bar{\Phi} = l_{max} - \frac{r_{jp}}{v_{jp}} - s_j$
  if $s_{ij}^{ready} \geq e_j$ **then**
    return $\omega_i = 0 \leftarrow$ Proposition 2
  else
    if $\bar{\Phi} \geq e_j$ **then**
      return $\omega_i = e_j - s_{ij}^{ready} \leftarrow$ Figure 6(iv)
    else
      if $\bar{\Phi} > s_{ij}^{ready}$ **then**
        return $\omega_i = \bar{\Phi} - s_{ij}^{ready} \leftarrow$ Figure 6(v)
      else
        return $\omega_i = 0 \leftarrow$ Figure 6(vi)

Case 2: The truck travels from node i to a retrieval node j. (Note that if $s_{ij}^{ready} + s_j \geq \hat{\tau}_j$, the result is the same as that in Case 1.)

Proposition 3. *When $e_j \leq s_{ij}^{ready} \leq l_j$, the penalty cost at node j is zero for any $\omega_i \in [0, l_j - s_{ij}^{ready}]$ if $\hat{\tau}_j - s_j \geq l_j$ (Figure 7i), and for any $\omega_i \in [0, \hat{\tau}_j - s_{ij}^{ready} - s_j]$ if otherwise (Figure 7ii).*

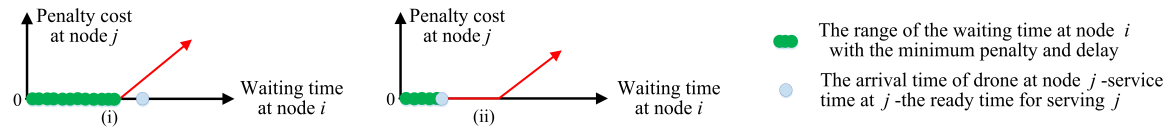

Figure 7: The penalty cost at node j in the case of $e_j \leq s_{ij}^{ready} \leq l_j$.

In view of Proposition 3, for $e_j \leq s_{ij}^{ready} \leq l_j$, if $\hat{\tau}_j - s_j \geq l_j$, we set $\omega_i = l_j - s_{ij}^{ready}$. Otherwise, $\omega_i = \hat{\tau}_j - s_{ij}^{ready} - s_j$.

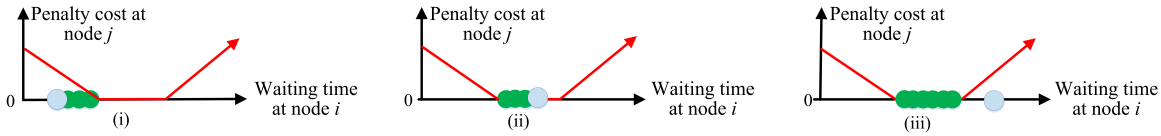

Figure 8: The penalty cost at node j in the case of $0 < s_{ij}^{ready} < e_j$.

Proposition 4. *When $0 < s_{ij}^{ready} < e_j$, we have: i) if $0 < \hat{\tau}_j - s_j < e_j$, the optimal waiting time at node i must take values from $[\hat{\tau}_j - s_{ij}^{ready} - s_j, e_j - s_{ij}^{ready}]$, allowing the minimum latency (Figure 8i); ii) otherwise, the penalty cost at node j is zero for any $\omega_i \in [e_j - s_{ij}^{ready}, \hat{\tau}_j - s_{ij}^{ready} - s_j]$ if $e_j \leq \hat{\tau}_j - s_j \leq l_j$ (Figure 8ii), and for any $\omega_i \in [e_j - s_{ij}^{ready}, l_j - s_{ij}^{ready}]$ if otherwise (Figure 8iii).*

In view of Proposition 4, we obtain the waiting time at node i when $0 < s_{ij}^{ready} < e_j$ as follows. If $e_j \leq \hat{\tau}_j - s_j \leq l_j$, we set $\omega_i = \hat{\tau}_j - s_{ij}^{ready} - s_j$; if $\hat{\tau}_j - s_j > l_j$, we set $\omega_i = l_j - s_{ij}^{ready}$; otherwise, ω_i can be determined by Algorithm III, which is similar to Algorithm 2 and included in Supplementary Materials D.

To sum up, Algorithm 3 outlines the waiting time optimisation at node i_g for all considered cases.

Algorithm 3. *Waiting time optimisation at node i_g for all considered cases*

Input: $s_{i_g,i_{g+1}}^{ready}$, the ready time for serving node i_{g+1}; $\hat{\tau}_{i_{g+1}}$, the drone arrival time at node i_{g+1}
Output: the best waiting time ω_{i_g}

if $i_g = i_G$ or $s_{i_g,i_{g+1}}^{ready} \geq l_{i_{g+1}}$ **then**
    return $\omega_{i_g} = 0 \leftarrow$ Lemma 1 & Proposition 1
else
    if i_{g+1} is not a retrieval node **then** # Case1
        if $s_{i_g,i_{g+1}}^{ready} \geq e_{i_{g+1}}$ **then**
            return $\omega_{i_g} = 0 \leftarrow$ Proposition 2
        else
            return $\omega_{i_g} \leftarrow$ Algorithm 2
    else # Case2
        if $s_{i_g,i_{g+1}}^{ready} + s_{i_{g+1}} \geq \hat{\tau}_{i_{g+1}}$ **then**
            if $s_{i_g,i_{g+1}}^{ready} \geq e_{i_{g+1}}$ **then**
                return $\omega_{i_g} = 0 \leftarrow$ Proposition 2
            else
                return $\omega_{i_g} \leftarrow$ Algorithm 2
        else
            if $s_{i_g,i_{g+1}}^{ready} \geq e_{i_{g+1}}$ **then**
                return $\omega_{i_g} \leftarrow$ Proposition 3
            else
                return $\omega_{i_g} \leftarrow$ Proposition 4

4.4. SAA mechanism

The pseudocode of the HMSA-SAA is presented in Algorithm 4, which repeatedly solves the stochastic problem M batches. In each batch m, a random sample of size H is generated and used to obtain the best solution s_H^m with the expected objective value of c_H^m (lines 5-6). Then the lower bound of the stochastic problem $\bar{c}_H^M$, defined as the average objective value of the M best solutions, is compared against the upper bound that is obtained in the same manner but with a much larger sample size H'. If the relative gap between the two bounds is sufficiently small ($gap < 5\%$), the algorithm returns the final best solution (line 16). Otherwise, it increases the value of H and repeats the above procedure (lines 12-13).

Algorithm 4. *HMSA-SAA*

Input: S, M, H, H', T_{02}, μ_2, $iterMax_2$, T_{f2}
Output: the best solution s^{M*}

1: Generate a sample of size H'
2: **while** True **do**
3: Initialisation: solution set $S_H^M \leftarrow \emptyset$
4: **for** m in M **do**
5: Generate a sample of size H

The best solution and cost s_H^m, $c_H^m \leftarrow$ HMSA(S, T_{02}, μ_2, $iterMax_2$, T_{f2}, H)
S_H^M.add(s_H^m)
Calculate the lower bound $\bar{c}_H^M = \frac{1}{M}\sum_{m\in M} c_H^m$ of the stochastic problem
for s_H^m in S_H^M **do**
Using larger sample size H' to obtain the upper bound $c_{H'}^m = \frac{1}{H'}\sum_{\xi\in H'} F(s_H^m, v^\xi)$
$c_{H'}^{m*} \leftarrow min\{c_{H'}^m\}$
if $gap = (c_{H'}^{m*} - \bar{c}_H^M)/\bar{c}_H^M >= 5\%$ **then**
$H+ = 10$
else
break
return $s^{M*} \leftarrow$ the solution s_H^m with cost of $c_{H'}^{m*}$ from the M solutions

Note: $F(s_H^m, v^\xi)$ is the cost of solution s_H^m with scenario v^ξ.

5. A rolling-horizon model for dynamic waiting time optimisation in the execution phase

Once the optimal routing decisions are identified in the planning phase, the optimal waiting time decisions for each truck can be derived using either a second-stage model or the heuristic Algorithm 3. However, both approaches require an input of the actual truck travel speeds on all arcs along the entire route, which are not available *a priori* and only become known sequentially as the truck traverses each arc. Thus, the estimated speeds can be used. Although satellite navigation services, such as Google Maps, are becoming more accurate, the weather, road conditions and traffic are dynamic and volatile, leading to unavoidable estimation errors, especially further down the route. Therefore, the obtained waiting time solutions may become suboptimal, given the inaccurate truck speeds used. To this end, we reformulate the second-stage model in a rolling-horizon manner that can be easily implemented in the executive phase. This model allows re-optimisation whenever a truck arrives at a customer node using an up-to-date estimate of the truck speeds in the remaining route at that time. Only the current waiting time decision is implemented, while the rest are discarded and re-optimised upon the truck's arrival at the next node.

For any vehicle pair, suppose that the truck arrives at node i_m — the mth customer on its route — at time τ_{i_m}. Let $N_S(m) = \{i_m, i_{m+1}, ..., i_G\}$ be the set of customers yet to be served by the truck. Recall that G is the number of customers served by the truck. The remaining truck route can be represented by ordered set $S^m = \{(i_m, i_{m+1}), (i_{m+1}, i_{m+2}), ..., (i_G, n+1)\}$. For the drone, denote the sequence of uncompleted flights by $R^m = \{f_m, ..., F\}$, where F is the total number of flights, and f_m is the index of the first uncompleted flight. Note that the flight currently in the air, if any, is not included. For each flight $f \in R^m$, let the set of customers covered by this flight be $N_f = \{i_1^f, ..., i_{|N_f|}^f\}$, and i_0^f and $i_{|N_f|+1}^f$ be the launch and retrieval nodes, respectively. At time τ_{i_m} the drone could be at one of three locations (as shown in Figure 9): i) already on the truck, ii) having just completed flight $f_m - 1$ and waiting to be retrieved at node i_m, and iii) still up in the air in flight $f_m - 1$, to be retrieved at another node further down the route. To track the drone's where-about, we use $\tilde{\tau}_{i_m}$ for the time when the drone would arrive at its next retrieval node, which takes the values of τ_{i_m}, $\hat{\tau}_{i_m}$ and $\hat{\tau}_{i^{f_m-1}_{|N_{f_m-1}|+1}}$, respectively, in the above 3 scenarios.

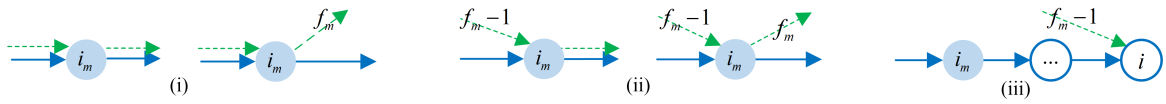

Figure 9: Various scenarios for the drone's location at time τ_{i_m}.

The set $N_S(m)$ can be partitioned into the following subsets: the nodes to be visited by the truck independently $N_S^1(m)$, the nodes to be visited by the vehicle pair together (excluding the launch/retrieval nodes) $N_S^2(m)$, the launch-only nodes $N_S^3(m)$, the retrieval-only nodes $N_S^4(m)$, and the retrieval-launch nodes $N_S^5(m)$. Moreover, let $N_R(m) = \bigcup_{f\in R^m} N_f$ be the set of all the customers yet to be served by the drone. All other variables are the same as those in the second-stage model, but without index k or ξ. For completeness, for $m = 0$ we let $i_0 = 0$ and $\tau_{i_0} = 0$ denote that the truck is still at the depot; in this case, we also have $f_0 = 1$ and $\tilde{\tau}_{i_0} = 0$ and the model becomes equivalent to the second-stage model (13)-(20c).

We are now ready to present the rolling-horizon waiting time optimisation model for the current truck visiting node (i_m) and the corresponding time stamps for the truck (τ_{i_m}) and drone ($\tilde{\tau}_{i_m}$):

$$min \sum_{i \in N_S(m+1)} (c^e \delta_i^- + c^l \delta_i^+) + \sum_{f \in R^m} \sum_{i_j^f \in N_f} (c^e \delta_{i_j^f}^- + c^l \delta_{i_j^f}^+) + c^k \delta, \tag{22}$$

subject to:

$$\tau_{i'} = \begin{cases} \tau_i + s_i + \frac{r_{i,i'}}{v_{i,i'}} + \omega_i, (i \in N_S^1(m), i \neq i_m) \bigvee (i \in N_S^2(m)), (i,i') \in S^m \\ \tau_i + s_i + s^l + \frac{r_{i,i'}}{v_{i,i'}} + \omega_i, i \in N_S^3(m), (i,i') \in S^m \end{cases} \tag{23a}$$

$$\tau_{i_{m+1}} = \begin{cases} max\{\tau_{i_m} + s_{i_m}, \tilde{\tau}_{i_m}\} + s^r + \frac{r_{i_m,i_{m+1}}}{v_{i_m,i_{m+1}}} + \omega_{i_m}, i_m \in N_S^4(m) \\ max\{\tau_{i_m} + s_{i_m}, \tilde{\tau}_{i_m}\} + s^r + s^l + \frac{r_{i_m,i_{m+1}}}{v_{i_m,i_{m+1}}} + \omega_{i_m}, i_m \in N_S^5(m) \end{cases} \tag{23b}$$

$$\tau_{i'} = \begin{cases} max\{\tau_i + s_i, \tilde{\tau}_{i_m}\} + s^r + \frac{r_{i,i'}}{v_{i,i'}} + \omega_i, i_m \in N_S^1(m), i = i^{f_m - 1}_{|N_{f_m-1}|} \in N_S^4(m) \\ max\{\tau_i + s_i, \tilde{\tau}_{i_m}\} + s^r + s^l + \frac{r_{i,i'}}{v_{i,i'}} + \omega_i, i_m \in N_S^1(m), i = i^{f_m - 1}_{|N_{f_m-1}|} \in N_S^5(m) \end{cases} \tag{23c}$$

$$\tau_{i'} = \begin{cases} \bar{\tau}_i + s^r + \frac{r_{i,i'}}{v_{i,i'}} + \omega_i, i \in N_S^4(m), i \neq i^{f_m - 1}_{|N_{f_m-1}|}, i \in N_S(m) \setminus i_m \\ \bar{\tau}_i + s^r + s^l + \frac{r_{i,i'}}{v_{i,i'}} + \omega_i, i \in N_S^5(m), i \neq i^{f_m - 1}_{|N_{f_m-1}|}, i \in N_S(m) \setminus i_m \end{cases} \tag{23d}$$

$$\hat{\tau}_{i_1^{f_m}} = \begin{cases} \tau_{i_0^{f_m}} + s_{i_0^{f_m}} + s^l + \frac{\hat{r}_{i_0^{f_m}, i_1^{f_m}}}{\hat{v}}, (i_m \in N_S^2(m) \cup N_S^4(m)) \bigvee (i_m \in N_S^1(m), i_0^{f_m} \in N_S^3(m)) \\ \tau_{i_m} + s_{i_m} + s^l + \frac{\hat{r}_{i_m, i_1^{f_m}}}{\hat{v}}, i_m \in N_S^3(m) \\ max\{\tau_{i_m} + s_{i_m}, \tilde{\tau}_{i_m}\} + s^r + s^l + \frac{\hat{r}_{i_m, i_1^{f_m}}}{\hat{v}}, i_m \in N_S^5(m) \\ max\{\tau_{i_0^{f_m}} + s_{i_0^{f_m}}, \tilde{\tau}_{i_m}\} + s^r + s^l + \frac{\hat{r}_{i_0^{f_m}, i_1^{f_m}}}{\hat{v}}, i_m \in N_S^1(m), i_0^{f_m} \in N_S^5(m) \end{cases} \tag{24a}$$

$$\tau_{i_0^f} + s_{i_0^f} + s^l + \frac{\hat{r}_{i_0^f, i_1^f}}{\hat{v}} \leq \hat{\tau}_{i_1^f} \leq \tau_{i_0^f} + s_{i_0^f} + s^l + \frac{\hat{r}_{i_0^f, i_1^f}}{\hat{v}}, i_0^f \in N_S^3(m), f \in R^m \setminus f_m \tag{24b}$$

$$\bar{\tau}_{i_0^f} + s^r + s^l + \frac{\hat{r}_{i_0^f, i_1^f}}{\hat{v}} \leq \hat{\tau}_{i_1^f} \leq \bar{\tau}_{i_0^f} + s^r + s^l + \frac{\hat{r}_{i_0^f, i_1^f}}{\hat{v}}, i_0^f \in N_S^5(m), f \in R^m \setminus f_m \tag{24c}$$

$$\hat{\tau}_{i_j^f} + \hat{s}_{i_j^f} + \frac{\hat{r}_{i_j^f, i_{j+1}^f}}{\hat{v}} \leq \hat{\tau}_{i_{j+1}^f} \leq \hat{\tau}_{i_j^f} + \hat{s}_{i_j^f} + \frac{\hat{r}_{i_j^f, i_{j+1}^f}}{\hat{v}}, i_j^f \in N_f, f \in R^m \tag{25}$$

$$\delta \geq \begin{cases} \tau_{n+1} - l_{max} \\ \hat{\tau}_{n+1} - l_{max}, n+1 \in N_S^4(m), \end{cases} \tag{26}$$

(19a)-(19g), $\forall i \in N_S(m) \cup N_R(m)$, and (20$a$)-(20$c$), $\forall i \in N_S^4(m) \cup N_S^5(m)$.

The objective function (22) minimises the total violation penalty of the remaining route. Constraint (23) calculates the truck arrival times in different situations (as shown in Figure 10), depending on the type and location of departure node i. Constraint (23a) considers the three situations where i is a non-retrieval node, as shown in Figure 10i. Constraints (23b)–(23d) consider the following situations where i is a retrieval node: 1) node i is i_m and the drone has just completed flight $f_m - 1$ (Figure 10ii), 2) the same as the previous one except that node i is further down the route (Figure 10iii), and 3) the drone has just completed flight $f \in R^m$ (Figure 10iv).

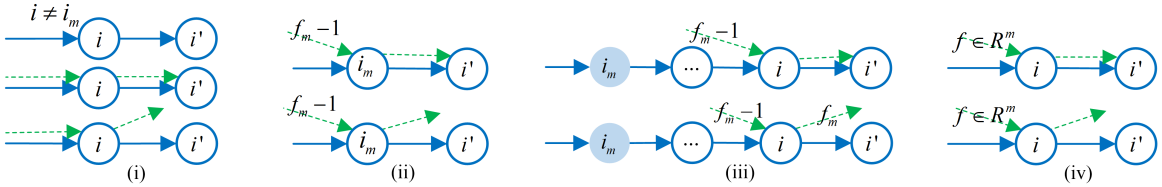

Figure 10: Cases in which the truck traverses arc (i, i').

Constraint (24a) calculates the drone arrival time at the *first* customer node in flight f_m under different situations (as shown in Figure 11i-iv). Constraints (24b) and (24c) calculate the drone arrival time at the *first* customer node in subsequent flights. The drone arrival times at all the remaining nodes in each flight

are calculated using constraint (25). Constraint (26) is equivalent to (18a)-(18b).

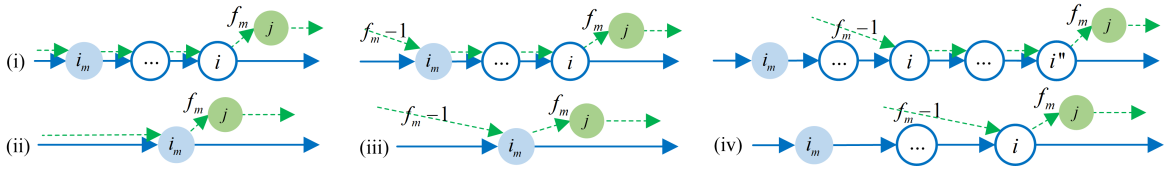

Figure 11: Cases for the arrival at the *first* drone-served node in flight f_m.

The above model can be easily solved using off-the-shelf solvers, and the problem size gradually decreases as the number of customers visited increases.

6. Computational experiments

In this section, a number of computational experiments are conducted to assess the benefits of stochastic modelling and the performance of the proposed hybrid metaheuristic approach, HMSA-S. Section 6.1 introduces the test instances and parameters used in the experiments. The solutions produced by HMSA-S for all instances are presented in Section 6.2. In Section 6.3, we conduct a simulation study to demonstrate the benefits of the stochastic approach and the effectiveness of dynamic waiting time optimisation. Finally, a sensitivity analysis is conducted on the three key parameters in Section 6.4. The metaheuristic was coded in Python 3.9.5, the rolling-horizon model was solved using Gurobi, and all computations were executed on a Windows 10 operating system equipped with an Intel (R) Core (TM) i7-10750H processor (2.60 GHz) and 32 GB of RAM.

6.1. Test instances

We randomly generate 36 test instances, grouped into four sets. Each set has a different number of customers $n \in \{30, 50, 70, 100\}$. The node locations are uniformly distributed in a square area of 10 km×10 km/12 km×12 km/12 km×12 km/15 km×15 km for the four sets (Liu et al., 2020). Note that our model and solution algorithms are not restricted to any specific graph of locations. Each set includes 3 time window scenarios (A, B and C), with 3 instances generated randomly in each scenario (Dellaert et al., 2019). Specifically, for each customer i we have A) $e_i \in [20, 260]$ and $l_i = e_i + 20$, B) $e_i \in [60, 360]$ and $l_i = e_i + 20$ and C) $e_i \in [60, 360]$ and $l_i = e_i + 90$. The depot has a time window of [0,450]. In each instance, the demand for 90% of the customers are randomly generated in the range (0, 2.3] (Liu et al., 2020). The remaining customers are designated as truck-only nodes, whose demands are sampled in the range (3, 50](Zhen et al., 2023). For convenience, each instance is called 'A/B/C.#.n'. All instances are available at 10.17632/d97vkm32w4.1.

The values of all the other model parameters are listed in Table 4a, while the parameters used in the metaheuristic approach are listed in Table 4b.

Table 4: List of parameters.

Parameter	**Value**	**Reference**
Distribution of the truck speed on each arc (F)	Truncated normal distribution $F(\mu, \sigma^2)$ over the support [20, 40] km/h, μ=30, σ^2=9 (i.e., the Coefficient of Variation CV=σ/μ=0.1)	Chen et al. (2014)
Load capacity of drones ($\hat{W}$)	3.0 kg	Liu et al. (2020)
Constant speed of drones ($\hat{v}$)	40 km/h	Nguyen et al. (2022)
Battery duration of drones (B)	30 min	Murray & Chu (2015)
Available capacity of trucks (W)	650 kg	Meng et al. (2024)
Setup times for drone launch and retrieval (s^l, s^r)	1 min, 1 min	Murray & Chu (2015)
Service times of trucks and drones (s, $\hat{s}$)	3 min, 2 min	Coindreau et al. (2021)
Travelling cost per unit distance for trucks (c)	\$0.78/km	Salama & Srinivas (2020)
Drone operating cost per unit time ($\hat{c}$)	\$0.06/min(=0.78/$\varepsilon * \hat{v}$/60), where the truck-to-drone unit cost ratio ε =8.3	Salama & Srinivas (2020)
Unit time penalty costs (c^e, c^l, c^k)	\$0.5/min, \$1.5/min, \$3.0/min	–
Fixed cost for each deployed vehicle pair (c^0)	\$30	–

(a) Model parameters.

Description	Value	Description	Value
Parameters used in the SAA		**Parameters used in the hybrid metaheuristic**	
M- number of batches	5	$T_0 = (T_{01}, T_{02})$- initial temperatures	(100,100)
N- sample size in each batch	15	$\beta = (\beta_1, \beta_2)$- cooling rates	(0.95,0.92)
N'- sample size used for solution evaluation	500	$T_f = (T_{f1}, T_{f2})$- floor temperatures	(10,10)
		$iterMax = (iterMax_1, iterMax_2)$- number of iterations	(20,20)

(b) Metaheuristic parameters.

6.2. HMSA-S solutions

In this section, we present in Table 5 the best solution obtained by the proposed metaheuristic HMSA-S for each of the 36 instances. The results include the operational cost (C_v), penalty cost (C_p), fixed cost(C_f), total cost (TC), number of deployed vehicle pairs (U), number of flights (F), number of drone-served customers (R), and number of drone-eligible customers (D). We also report the proportions of the penalty cost to the total cost ($\Delta_{p/c}\% = 100 * \frac{C_p}{TC}$) and drone-served customers to drone-eligible customers ($\Delta_{r/d}\% = 100 * \frac{R}{D}$), and the computational time (cpu).

Table 5: HMSA-S solutions.

Instance	C_v	C_p	C_f	TC	$\Delta_{p/c}$	U	F	R	D	$\Delta_{r/d}$	cpu(s)	Instance	C_v	C_p	C_f	TC	$\Delta_{p/c}$	U	F	R	D	$\Delta_{r/d}$	cpu(s)
A.1.30	53.92	0.00	60.00	113.92	0.00	2	10	15	27	55.56	769.80	A.1.70	102.47	0.36	90.00	192.83	0.19	3	20	36	63	57.14	2179.76
A.2.30	48.90	0.00	60.00	108.90	0.00	2	10	18	27	66.67	687.22	A.2.70	103.70	1.67	90.00	195.37	0.85	3	18	36	63	57.14	2203.76
A.3.30	54.37	0.00	60.00	114.37	0.00	2	9	16	27	59.26	759.17	A.3.70	106.86	0.44	90.00	197.30	0.22	3	18	33	63	52.38	2328.33
B.1.30	62.06	0.27	30.00	92.33	0.29	1	9	17	27	62.96	1058.66	B.1.70	112.83	1.56	90.00	204.39	0.76	3	18	33	63	52.38	3523.60
B.2.30	82.53	1.46	30.00	113.99	1.28	1	8	15	27	55.56	850.15	B.2.70	120.08	0.00	90.00	210.08	0.00	3	17	31	63	49.21	2351.24
B.3.30	55.85	0.00	60.00	115.85	0.00	2	11	15	27	55.56	976.39	B.3.70	112.83	10.22	90.00	213.05	4.80	4	22	36	63	57.14	2198.35
C.1.30	47.70	0.00	30.00	77.70	0.00	1	7	15	27	55.56	606.07	C.1.70	75.82	0.00	60.00	135.82	0.00	2	19	38	63	60.32	1743.34
C.2.30	49.22	0.01	30.00	79.23	0.01	1	10	15	27	55.56	716.38	C.2.70	84.72	0.00	60.00	144.72	0.00	2	20	40	63	63.49	2454.97
C.3.30	39.97	0.00	30.00	69.97	0.00	1	8	19	27	70.37	660.68	C.3.70	79.34	0.24	60.00	139.58	0.17	2	20	41	63	65.08	2269.07
Avg.					**0.18**					**59.67**		**Avg.**					**0.78**					**57.14**	
A.1.50	96.73	0.80	90.00	187.53	0.43	3	16	26	45	57.78	1830.18	A.1.100	184.18	7.88	150.00	342.06	2.30	5	25	47	90	52.22	4486.55
A.2.50	106.82	2.50	60.00	169.32	1.48	2	14	24	45	53.33	1761.63	A.2.100	209.83	6.87	120.00	336.70	2.04	4	29	52	90	57.78	3412.94
A.3.50	89.41	4.48	90.00	183.89	2.44	3	17	26	45	57.78	1347.87	A.3.100	190.72	12.71	120.00	323.43	3.93	4	24	43	90	47.78	3353.65
B.1.50	87.00	1.53	60.00	148.53	1.03	2	14	25	45	55.56	2032.34	B.1.100	206.82	3.51	120.00	330.33	1.06	4	30	49	90	54.44	3409.28
B.2.50	109.39	3.34	60.00	172.73	1.93	2	14	23	45	51.11	3521.11	B.2.100	182.96	1.14	150.00	334.10	0.34	5	27	49	90	54.44	5743.05
B.3.50	86.09	3.66	60.00	149.75	2.44	2	17	27	45	60.00	2345.88	B.3.100	187.50	0.22	120.00	307.72	0.07	4	27	49	90	54.44	5609.95
C.1.50	74.00	0.00	60.00	134.00	0.00	2	14	28	45	62.22	1985.31	C.1.100	136.14	0.15	90.00	226.29	0.07	3	27	55	90	61.11	3623.93
C.2.50	72.48	0.34	60.00	132.82	0.26	2	16	29	45	64.44	971.11	C.2.100	150.87	0.22	90.00	241.09	0.09	3	26	52	90	57.78	2981.98
C.3.50	69.08	0.00	60.00	129.08	0.00	2	15	29	45	64.44	1939.53	C.3.100	133.85	1.85	120.00	255.70	0.72	4	28	60	90	66.67	3693.09
Avg.					**1.11**					**58.52**		**Avg.**					**1.18**					**56.30**	

It is shown that the instances in Scenario C have the lowest total costs compared with their counterparts in the other two scenarios, and they usually require the least number of vehicle pairs owing to their widest time windows. In most instances, more than 50% of drone-eligible customers are served by drones. Although the ratio is slightly higher in Scenario C than in the other scenarios, it is relatively stable over different instance sizes within each scenario. The number of flights is comparable across the scenarios. In addition, the computational time increases with the instance size. For scenarios of the same size, the computational time in Scenario C is the shortest, followed by those in Scenarios A and B. In all scenarios, the time window violation penalties represent a small proportion of the total costs, always less than 4.80%. A detailed analysis of waiting time is presented in Section 6.3.2. As we shall see, this is not always the case, and the time window violation penalties could be significantly higher if trucks are not allowed to wait.

6.3. Benefits of the stochastic approach and dynamic waiting time optimisation

To evaluate the benefits of the stochastic approach and effectiveness of dynamic waiting time optimisation, we conduct the following simulation study. For each problem instance, the routing decisions are obtained using the HMSA-S. Subsequently, for every truck-drone pair, we first generate a sample of truck travel speeds along the entire route and solve the rolling-horizon model to determine the waiting time at the depot. Then, the truck waits at the depot as required before setting off to the first customer node. We assume that the sampled travel speed for the next arc is always the true speed. Upon arrival at the first customer node, another sample of truck speeds is generated for the remaining route, and the rolling-horizon model is solved again to determine the waiting time at that node. This procedure is repeated until the vehicle pair is returned to the depot. The drones are launched and retrieved as prescribed by their flights. We calculate the total cost for all vehicle pairs in each replication. In total, 500 replications are conducted for each instance, and the average total costs are reported.

6.3.1. Benefits of the stochastic modelling approach

In this section, we evaluate the benefits of the stochastic modelling approach for the problem concerned. To this end, we compare the total costs produced by HMSA-S against those of its deterministic counterpart HMSA which ignores truck travel time uncertainties. Moreover, we include two benchmark heuristics from the literature: the large neighbourhood search (LNS) from Kitjacharoenchai et al. (2020) and the adaptive multi-start simulated annealing algorithm (AMSA) from Masmoudi et al. (2022). Both heuristics are designed for a deterministic VRP-D. The average total costs for each of the three deterministic heuristics

are calculated via the same simulation framework as mentioned above, except that the waiting times are obtained via the second-stage model using the mean truck travel speeds; thus, the waiting time decisions remain the same across the 500 replications. In addition, three levels of truck speed variations are considered in these experiments: Low, Moderate, and High, equal to 0.1, 0.2, and 0.3, respectively. Moreover, considering that truck speeds on some arcs may experience higher uncertainties than the others in practice, a mixed scenario where CVs are evenly set to be 0.1/0.2/0.3 among all arcs is included.

The detailed results for all instances in Scenario A are presented in Table 6, which lists the total cost of the best solution among five runs for HMSA-S. It also reports the extra percentage cost owing to each deterministic heuristic compared to HMSA-S. It is shown that HMSA-S leads to considerable cost reductions compared with deterministic heuristics in all considered scenarios. Specifically, the extra costs due to LNS and AMSA are greater than 14%, and the gap increases with the problem size and degree of uncertainty. The performance of HMSA is slightly better, but still significantly worse than that of its stochastic counterpart. Therefore, stochastic modelling is essential for reducing the overall cost of such problems.

Table 6: Total costs of the stochastic and deterministic approaches.

Instance	CV	Total cost	Extra % cost over HMSA-S			Instance	CV	Total cost	Extra % cost over HMSA-S		
		HMSA-S	HMSA	LNS	AMSA			HMSA-S	HMSA	LNS	AMSA
A.1.30	L	**113.92**	7.79	6.37	10.52	A.1.70	L	**194.36**	14.09	9.87	13.09
	M	**113.92**	8.99	7.01	11.73		M	**194.57**	16.19	10.91	15.54
	H	**113.92**	10.45	7.30	12.43		H	**195.58**	15.96	10.79	15.68
	X	**113.92**	10.45	8.74	14.22		X	**195.57**	16.62	12.73	18.77
A.2.30	L	**114.30**	5.51	5.60	6.20	A.2.70	L	**200.14**	14.10	25.71	37.32
	M	**114.31**	6.63	6.79	6.91		M	**200.47**	17.41	28.32	42.17
	H	**114.33**	6.80	7.18	7.23		H	**201.46**	17.39	27.81	42.35
	X	**115.30**	7.45	8.39	7.69		X	**203.47**	17.16	28.46	44.47
A.3.30	L	**114.41**	12.10	9.29	9.29	A.3.70	L	**201.65**	13.40	23.37	7.58
	M	**114.41**	15.13	10.93	11.57		M	**202.01**	15.36	24.04	8.76
	H	**114.41**	16.77	12.16	12.16		H	**202.47**	15.36	24.31	7.62
	X	**114.41**	19.73	11.89	12.41		X	**204.90**	14.66	23.78	9.13
A.1.50	L	**189.01**	9.64	11.28	21.13	A.1.100	L	**351.44**	20.01	12.89	17.35
	M	**189.11**	11.23	13.56	24.97		M	**352.41**	23.42	15.63	19.60
	H	**192.99**	10.38	11.93	26.94		H	**357.79**	20.04	16.18	18.31
	X	**194.11**	9.75	12.59	34.11		X	**357.79**	23.11	16.95	19.63
A.2.50	L	**172.37**	18.52	19.62	17.29	A.2.100	L	**342.10**	17.44	23.69	19.33
	M	**174.93**	18.70	24.34	18.12		M	**343.52**	20.09	22.34	20.53
	H	**176.06**	18.28	24.56	18.29		H	**347.34**	19.18	26.93	19.10
	X	**176.06**	18.28	23.93	22.02		X	**349.52**	21.61	26.09	22.53
A.3.50	L	**185.29**	8.55	13.13	7.99	A.3.100	L	**441.65**	6.79	8.83	6.49
	M	**185.81**	10.65	15.34	11.19		M	**446.89**	7.50	10.28	7.52
	H	**186.97**	10.88	15.04	10.32		H	**447.75**	6.79	11.62	6.87
	X	**189.03**	9.68	15.06	10.88		X	**446.89**	8.59	11.90	9.22
Avg.	L		**12.33**	**14.14**	**14.47**		H		**14.02**	**16.32**	**16.44**
	M		**14.28**	**15.79**	**16.55**		X		**14.76**	**16.71**	**18.76**

L: low. M: moderate. H: high. X: mixed.

6.3.2. Effectiveness of the rolling-horizon waiting time optimisation approach

To evaluate the effectiveness of the proposed rolling-horizon waiting time optimisation approach (RH), we compare its performance against two other approaches to the waiting time: 1) no waiting at all (NW), and 2) ignoring the truck travel time uncertainties and solving the second-stage model with the mean truck travel speeds (MW). The total costs are calculated using the same simulation framework mentioned above. Specifically, routing decisions are first obtained via HSMA-S. The simulation then proceeds as described, with the waiting times obtained using each of these alternative approaches. The same samples are used for all alternatives. At the end of each replication, one would obtain a realisation of the travel speeds for every truck route. We then use this realisation in the second-stage model and obtain the optimal waiting times as if the truck speeds were known before departure. The resulting total cost serves as a lower bound (LB) for the problem. The results for all instances in Scenario B with the three CV values are presented in Table 7.

When no waiting is allowed, the total cost is significantly higher than that of the other two alternative approaches; the cost could be almost tripled compared with LB. The mean-speed waiting time approach clearly improves the performance, reducing the total costs, but is still 20% over the LB on average. In sharp contrast, the proposed rolling-horizon approach is very close to the LB, with an overall gap of less than 3%. Another observation is that although the total costs increase with the CV values, the performance of RH is always robust compared to the other two approaches.

To further investigate the benefits of waiting time optimisation, we compare the cost breakdown under NW and LB for Scenarios A and B with CV=0.1, as shown in Figure 12. It is shown that the total cost of LB compared with that of NW decreases more significantly in Scenario B than in Scenario A because the former has a wider range of window-opening times than the latter, allowing more opportunities to reduce the penalties by waiting. Moreover, in the absence of waiting time, the penalty cost fluctuates significantly and

Table 7: Total costs of alternative approaches to the waiting time.

Instance	CV	Total cost	Extra % cost over LB			Instance	CV	Total cost	Extra % cost over LB		
		LB	NW	MW	RH			LB	NW	MW	RH
B.1.30	L	**92.40**	153.24	28.64	2.11	B.1.70	L	**203.04**	210.13	14.18	0.89
	M	**94.06**	148.88	32.05	2.20		M	**203.04**	216.02	16.11	1.32
	H	**94.85**	148.86	34.12	2.33		H	**203.12**	265.88	16.23	0.91
B.2.30	L	**113.44**	232.55	28.15	5.53	B.2.70	L	**208.17**	205.19	15.25	3.01
	M	**113.51**	237.01	28.27	8.46		M	**208.92**	208.19	17.96	2.89
	H	**113.79**	242.12	30.42	8.17		H	**210.08**	211.49	17.80	2.45
B.3.30	L	**115.85**	136.62	11.17	0.00	B.3.70	L	**214.00**	171.28	18.87	3.43
	M	**116.87**	141.03	9.87	1.15		M	**215.79**	182.43	21.47	3.79
	H	**116.89**	146.22	11.07	1.14		H	**215.97**	184.96	22.83	4.40
B.1.50	L	**148.51**	164.03	13.96	2.30	B.1.100	L	**327.35**	186.24	19.87	1.96
	M	**151.12**	163.19	19.74	2.85		M	**328.15**	196.14	22.02	2.29
	H	**151.96**	164.25	24.26	3.78		H	**331.72**	199.35	23.63	2.82
B.2.50	L	**173.00**	256.90	18.33	3.23	B.2.100	L	**333.98**	157.21	17.60	1.00
	M	**173.05**	259.19	18.39	3.61		M	**333.95**	165.99	18.43	1.10
	H	**173.34**	269.10	22.49	4.15		H	**333.99**	179.61	18.87	1.29
B.3.50	L	**150.34**	218.91	19.85	4.86	B.3.100	L	**307.46**	170.24	20.17	0.95
	M	**150.64**	226.20	20.00	4.83		M	**307.59**	191.21	24.59	0.97
	H	**154.77**	224.01	28.82	7.42		H	**307.75**	207.02	26.54	1.27
						Avg.			**195.58**	**20.89**	**2.91**

increases with instance size, whereas in the case of waiting time optimisation, the penalty cost is relatively small and robust.

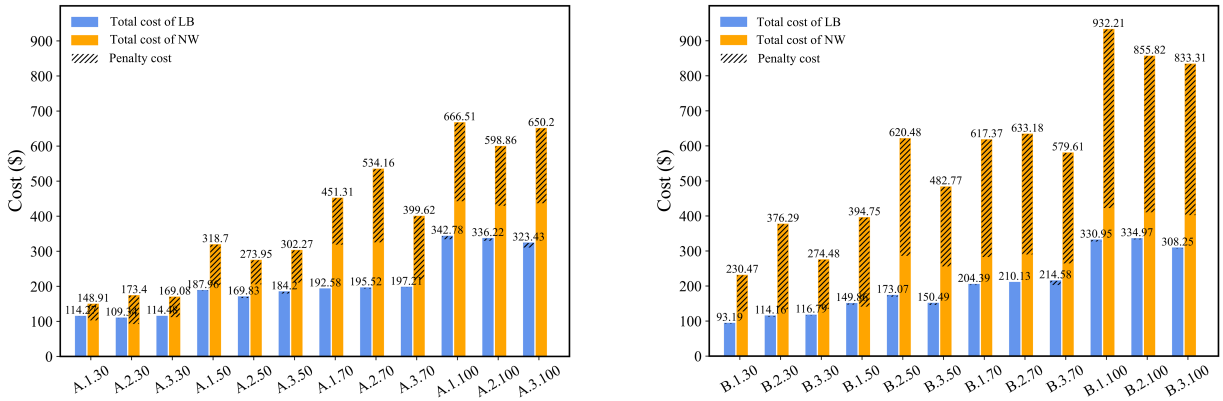

Figure 12: Cost breakdown comparison of NW and LB for Scenarios A and B.

6.4. Sensitivity analysis

In this section we conduct sensitivity analyses to identify the impact of the penalty cost rate, variation in truck travel speeds, and drone unit cost on the solutions of mDRP-TS via the simulation. Unless otherwise specified, a CV of 0.1 is used.

6.4.1. The impact of the penalty cost rate

Recall that in the experiments conducted thus far, we have $(c^e, c^l, c^k) = (0.5, 1.5, 3.0)$. We now multiply all cost rates by ρ to study the impacts of different penalty cost rates on the solutions to the problem. Figure 13 shows the cost breakdown for the six instances of B/C.#.50 for different ρ values. Overall, both the total and fixed costs increase with the penalty cost rate, whereas the operational cost is relatively stable. Interestingly, the total penalty cost decreases with ρ. This is because higher penalty cost rates force more vehicle pairs to be deployed. Thus, more customers can be served within their time windows, leading to fewer violations and lower total penalty costs. Moreover, the instances of B.#.50 are more sensitive to the penalty cost rate than those of C.#.50 because the latter have wider time windows that always contribute to fewer violations.

6.4.2. The impact of the Coefficient of Variation of truck speeds

This section investigates the impact of the variation in truck speeds on the cost breakdown in all instances in Scenario B. Table 8 displays the comparison results of the fixed, operational, penalty, and total costs in different CV scenarios. Overall, the fixed cost is the same for different CV values in each instance, implying that the number of vehicles deployed remains the same. The total cost always increases with the CV values, with the highest value observed in the mixed scenarios. Interestingly, the operational cost or

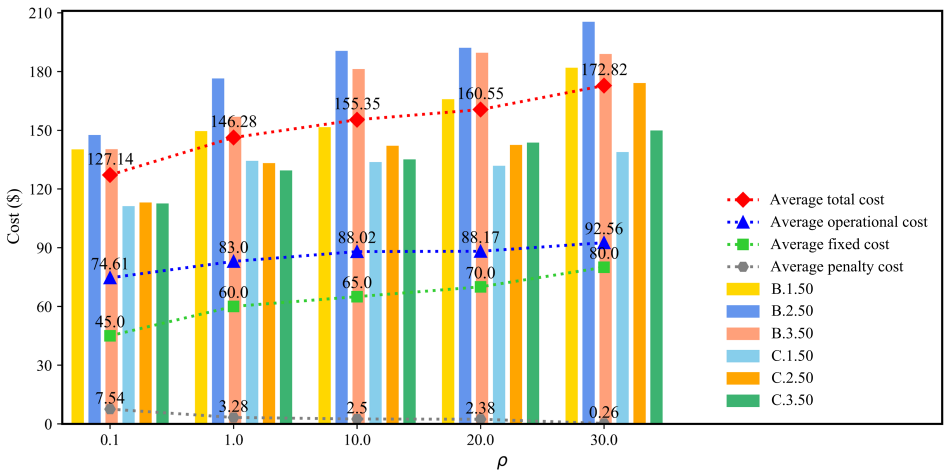

Figure 13: The impact of the penalty cost rate.

penalty cost does not always increase with the CV values. In these cases, the vehicle routes/flights are changed slightly to reduce the penalty cost, at the expense of higher operational costs.

Table 8: Cost breakdown with different CV values.

Instance	C_f	C_v				C_p				TC			
		L	M	H	X	L	M	H	X	L	M	H	X
B.1.30	30.00	62.06	62.94	62.94	**64.20**	0.24	1.09	**2.96**	2.60	92.30	94.03	95.90	**96.80**
B.2.30	30.00	82.53	82.91	82.91	**83.02**	1.04	0.70	2.81	**3.94**	113.57	113.61	115.72	**116.96**
B.3.30	60.00	55.85	56.83	56.83	**56.94**	0.00	0.12	2.15	**2.38**	115.85	116.95	118.98	**119.32**
B.1.50	60.00	87.00	97.71	97.71	**101.35**	1.56	0.42	0.82	**2.59**	148.56	158.13	158.53	**163.94**
B.2.50	60.00	109.39	109.83	115.66	**121.46**	**3.40**	3.03	0.39	2.36	172.79	172.86	176.05	**183.82**
B.3.50	60.00	86.09	**91.01**	86.71	90.17	4.41	4.33	8.63	**8.87**	150.50	155.34	155.34	**159.04**
B.1.70	90.00	112.83	127.23	127.76	**135.86**	0.35	0.28	**0.75**	0.57	203.18	217.51	218.51	**226.43**
B.2.70	90.00	120.08	117.45	115.67	**122.53**	0.00	0.72	3.25	**5.73**	210.08	208.17	208.92	**218.26**
B.3.70	90.00	112.83	123.07	124.31	**131.06**	**11.65**	6.41	5.33	6.86	214.48	219.48	219.64	**227.92**
B.1.100	120.00	206.82	208.56	203.95	**211.08**	1.35	1.83	**7.81**	5.55	328.17	330.39	331.76	**336.63**
B.2.100	150.00	**182.96**	182.86	182.87	181.53	1.09	1.25	1.28	**4.34**	334.05	334.11	334.15	**335.87**
B.3.100	120.00	187.50	215.14	216.24	**221.93**	0.01	2.39	2.46	**5.84**	307.51	337.53	338.70	**347.77**
Avg.	**80.00**	**117.16**	**122.96**	**122.80**	**126.76**	**2.09**	**1.88**	**3.22**	**4.30**	**199.25**	**204.84**	**206.02**	**211.06**

6.4.3. The impact of the truck-to-drone unit cost ratio

Figure 14 shows the impact of the truck-to-drone unit cost ratio (ε) on the total cost, number of customers served by drones, and truck travel distance in instances B/C.#.50. We notice that as ε increases; that is, when the drone unit cost drops (truck unit cost is fairly fixed), more customers are served by drones, and thus the distance travelled by trucks is reduced. The total cost also decreases, and the reduction is most obvious as ε increases from 1 to 5, after which the reduction rate decreases. These results indicate that the benefits of the truck-drone combined mode are more pronounced if the drone unit cost is kept low. Figure 14 also indicates that for the same ε the total cost and truck travel distance are higher in Scenario B.#.50 than in Scenario C.#.50; however, there are no clear differences in the number of customers served by the drones between these two scenarios.

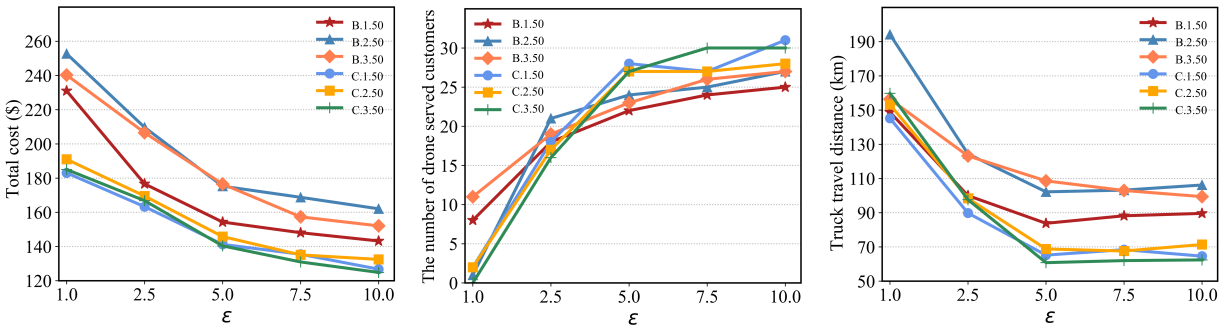

Figure 14: The impact of the truck-to-drone unit cost ratio.

7. Conclusion

In this study, we examined a new variant of the truck-drone combined routing problem in last-mile logistics; that is, the multi-visit drone-assisted routing problem with soft time windows and stochastic truck travel times. In this problem, a fleet of capacitated truck-drone pairs is deployed to provide delivery services to customers collaboratively. Each drone can be launched from and retrieved to its truck multiple times. During each flight, the drone can serve multiple customers. Each customer is associated with a soft time window, and starting a service outside this time window incurs penalty costs. Trucks are allowed to wait at the depot and each customer node before departing. Our objective is to minimise the total cost, including time window violation penalties. We formulated the problem as a two-stage stochastic model, which was then solved using a customised HMSA that integrates the SAA framework. Moreover, we proposed a rolling-horizon approach to dynamically optimise waiting time during the execution phase. This approach is efficient and scalable, and only a modest linear programming model must be solved each time. Extensive numerical experiments were conducted to assess the performance of the proposed metaheuristic approach and benefits of the stochastic model. The effectiveness of the rolling-horizon approach was thoroughly evaluated. Finally, a sensitivity analysis was performed to quantitatively assess the impacts of the key parameters on the problem results with valuable managerial insights.

Despite the potential benefits in last-mile logistics, combined truck-drone deliveries are not yet widely adopted in practice. One hurdle is the infrastructure such as self-service express lockers and control centres, which are still immature for drone operations. In addition, owing to the limitations of technological development, drone costs are usually variable and uncertain, and the cost advantages of combined delivery services may not always be obvious, especially in the early stages. Moreover, scheduling and operational challenges in synchronising trucks and drones for seamless services could be another hurdle. Finally, the relevant regulations for the commercial application of drones are still evolving and are very different across regions or countries. Nevertheless, it is worth mentioning that with the continuous development of technology, infrastructure and regulation, the adoption of the combined truck-drone deliveries would become more widely applicable.

Future studies should investigate more practical application scenarios where each truck is equipped with multiple drones where each drone can conduct multiple fights at each node. Moreover, the consideration of stochastic and time-dependent truck travel times could be an interesting extension. Another promising research direction is to consider stochastic travel times for drones that are sensitive to weather.

Acknowledgments

This work is supported by the China Scholarship Council [grant no. 202207000043], the National Nature Science Foundation of China [grant no. 72371206].

References

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4), 965–981.

Chen, L., Hà, M. H., Langevin, A., & Gendreau, M. (2014). Optimizing road network daily maintenance operations with stochastic service and travel times. *Transportation Research Part E: Logistics and Transportation Review*, 64, 88–102.

Coindreau, M.-A., Gallay, O., & Zufferey, N. (2021). Parcel delivery cost minimization with time window constraints using trucks and drones. *Networks*, 78(4), 400–420.

Crumley, B. (2023). Zipline to muscle up nascent UK medical drone delivery network. Accessed January 16, 2024, URL `https://dronedj.com/2023/12/17/zipline-to-muscle-up-nascent-uk-medical-drone-delivery-network/`.

Dellaert, N., Dashty Saridarq, F., Van Woensel, T., & Crainic, T. G. (2019). Branch-and-price–based algorithms for the two-echelon vehicle routing problem with time windows. *Transportation Science*, 53(2), 463–479.

Di Puglia Pugliese, L. & Guerriero, F. (2017). Last-mile deliveries by using drones and classical vehicles,

in *Optimization and Decision Science: Methodologies and Applications: ODS, Sorrento, Italy, September 4-7, 2017 47*, (pp. 557–565), Springer.

Edwards, D. (2018). Aha partners with Flytrex expands on-demand drone delivery service in Iceland. Accessed November 10, 2023, URL https://roboticsandautomationnews.com/2018/08/07/aha-partners-with-flytrex-expands-on-demand-drone-delivery-service-in-iceland/18612/.

FAA. (2023). FAA authorizes Zipline international, inc. to deliver commercial packages using drones that fly beyond operator's line of sight. Accessed January 22, 2024, URL https://www.faa.gov/newsroom/faa-authorizes-zipline-deliver-commercial-packages-beyond-line-sight.

Figliozzi, M. A. (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transportation Research Part C: Emerging Technologies*, 18(5), 668–679.

Gendreau, M., Laporte, G., & Séguin, R. (1996). Stochastic vehicle routing. *European journal of operational research*, 88(1), 3–12.

Hu, X. (2020). What is the development of urban air distribution in Hangzhou after the "license"? (in Chinese). Accessed November 10, 2023, URL http://www.caacnews.com.cn/1/10/202001/t20200107_1289608.html.

Jones, C. (2023). Amazon unveils plan to deliver packages by drone in UK and Italy. Accessed January 16, 2024, URL https://www.theguardian.com/technology/2023/oct/18/amazon-drone-delivery-uk-italy.

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.

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M., & Brunese, P. A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering*, 129, 14–30.

Kloster, K., Moeini, M., Vigo, D., & Wendt, O. (2023). The multiple traveling salesman problem in presence of drone-and robot-supported packet stations. *European Journal of Operational Research*, 305(2), 630–643.

Kuo, R., Lu, S.-H., Lai, P.-Y., & Mara, S. T. W. (2022). Vehicle routing problem with drones considering time windows. *Expert Systems with Applications*, 191, 116264.

Kyriakakis, N. A., Stamadianos, T., Marinaki, M., & Marinakis, Y. (2022). The electric vehicle routing problem with drones: An energy minimization approach for aerial deliveries. *Cleaner Logistics and Supply Chain*, 4, 100041.

Li, H., Wang, H., Chen, J., & Bai, M. (2020). Two-echelon vehicle routing problem with time windows and mobile satellites. *Transportation Research Part B: Methodological*, 138, 179–201.

Liu, Y., Liu, Z., Shi, J., Wu, G., & Pedrycz, W. (2020). Two-echelon routing problem for parcel delivery by cooperated truck and drone. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(12), 7450–7465.

Liu, Z., Li, X., & Khojandi, A. (2022). The flying sidekick traveling salesman problem with stochastic travel time: A reinforcement learning approach. *Transportation Research Part E: Logistics and Transportation Review*, 164, 102816.

Luo, Z., Gu, R., Poon, M., Liu, Z., & Lim, A. (2022). A last-mile drone-assisted one-to-one pickup and delivery problem with multi-visit drone trips. *Computers & Operations Research*, 148, 106015.

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.

Meng, S., Chen, Y., & Li, D. (2024). The multi-visit drone-assisted pickup and delivery problem with time windows. *European Journal of Operational Research*, 314(2), 685–702.

Meng, S., Guo, X., Li, D., & Liu, G. (2023). The multi-visit drone routing problem for pickup and delivery services. *Transportation Research Part E: Logistics and Transportation Review*, 169, 102990.

Muller, J. (2024). 2024 will be a breakout year for delivery drones. Accessed January 16, 2024, URL https://www.axios.com/2024/01/02/delivery-drones-2024-amazon-zipline-wing.

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.

Nguyen, M. A., Dang, G. T.-H., Hà, M. H., & Pham, M.-T. (2022). The min-cost parallel drone scheduling vehicle routing problem. *European Journal of Operational Research*, 299(3), 910–930.

Rave, A., Fontaine, P., & Kuhn, H. (2023). Drone location and vehicle fleet planning with trucks and aerial

drones. *European Journal of Operational Research*, 308(1), 113–130.
Sacramento, D., Pisinger, D., & Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, 102, 289–315.
Salama, M. & Srinivas, S. (2020). Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C: Emerging Technologies*, 114, 620–642.
Schermer, D., Moeini, M., & Wendt, O. (2019). A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers & Operations Research*, 109, 134–158.
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.
Taş, D., Dellaert, N., Van Woensel, T., & De Kok, T. (2013). Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research*, 40(1), 214–224.
Wang, X., Poikonen, S., & Golden, B. (2017). The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, 11(4), 679–697.
Wang, Y., Wang, Z., Hu, X., Xue, G., & Guan, X. (2022). Truck-drone hybrid routing problem with time-dependent road travel time. *Transportation Research Part C: Emerging Technologies*, 144, 103901.
Wang, Z. & Sheu, J.-B. (2019). Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, 122, 350–364.
Yang, Y., Yan, C., Cao, Y., & Roberti, R. (2023). Planning robust drone-truck delivery routes under road traffic uncertainty. *European Journal of Operational Research*, 309(3), 1145–1160.
Yin, Y., Li, D., Wang, D., Ignatius, J., Cheng, T., & Wang, S. (2023). A branch-and-price-and-cut algorithm for the truck-based drone delivery routing problem with time windows. *European Journal of Operational Research*, 309(3), 1125–1144.
Zhen, L., Gao, J., Tan, Z., Wang, S., & Baldacci, R. (2023). Branch-price-and-cut for trucks and drones cooperative delivery. *IISE Transactions*, 55(3), 271–287.
Zhou, H., Qin, H., Cheng, C., & Rousseau, L.-M. (2023). An exact algorithm for the two-echelon vehicle routing problem with drones. *Transportation Research Part B: Methodological*, 168, 124–150.