European Journal of Operational Research 318 (2024) 457–468

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Production, Manufacturing, Transportation and Logistics

Drone resupply with multiple trucks and drones for on-time delivery along given truck routes

Wenqian Liu [a], Lindong Liu [b], Xiangtong Qi [a,*]

[a] *Department of Industrial Engineering and Decision Analytics, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong*
[b] *International Institute of Finance, School of Management, University of Science and Technology of China, 230026, P.R. China*

ARTICLE INFO

Keywords:
Logistics
Drone resupply
On-time delivery
Time-expanded flow network

ABSTRACT

Drones have been increasingly used for deliveries; however, delivering packages directly to customers is still challenging, particularly in densely populated urban areas. To overcome this, drone resupply has been proposed, where drones carry packages to trucks en route, and then trucks make the final deliveries. Existing studies have only considered the case of one truck and one drone. In this study, we investigate drone resupply with multiple trucks and drones for the on-time delivery of late-available packages. Considering practical operations in urban areas, we focus on a situation where trucks deliver packages along given routes. This raises the drone resupply scheduling problem, which aims to maximise the total value of the late-available packages that can be delivered on time. We formally define the problem and analyse its optimality properties. Then, we develop an exact solution approach based on a time-expanded network flow model and design a greedy heuristic algorithm. We validate the efficiency and effectiveness of the proposed solutions through computational studies. Operational insights and guidance are derived for the application of drones in on-time delivery.

1. Introduction

In the logistics industry, drones play an increasingly important role, particularly in fast and flexible last-mile delivery services. Over the past decade, prominent logistics and e-commerce companies have invested in drone delivery systems. Experimental projects, such as those conducted by Amazon (Hern, 2016), FedEx (Norman, 2019), and Google X (Kruk, 2020), have demonstrated the feasibility of drone deliveries.

Several drone delivery systems have been proposed, which either rely solely on drones or use drones and trucks in a coordinated fashion. Most models implicitly assume that drones can directly deliver packages to customers, such as by dropping them off in the backyard. However, this approach may not be feasible for those without an appropriate drop-off site. For example, in densely populated urban areas, many people live and work in high-rise buildings, making it difficult to implement these delivery models. Therefore, all packages should still be handed over to customers by the delivery staff to ensure safe and reliable deliveries.

Such a situation leads to the development of the drone resupply model, where drones are used to carry packages from depots to trucks en route, and then, trucks make the final deliveries to customers. Several studies, including Dayarian et al. (2020), Pina-Pardo et al. (2021), and Dienstknecht et al. (2022), have explored this concept in various applications. As outlined in Table 1, these studies have primarily focused on problems with one truck and one drone. In this paper, we introduce a new drone resupply model that involves multiple trucks and multiple drones, a significant departure from existing research. It is important to note that our problem is not a direct extension of any single-truck-drone problem studied in the literature. Our model is motivated by its own unique objectives, decision-making processes, and constraints.

Our study is motivated by two critical scenarios of on-time delivery, namely, inter-city overnight shipping and guaranteed time-definite delivery for online shopping. Overnight shipping involves transporting packages from other cities to a depot during nighttime. This long-distance transportation may be delayed by adverse weather conditions, resulting in packages arriving at the depot after their intended dispatch time. In online shopping, a seller may guarantee order delivery times. An example is the ‘Free 2-Hour Grocery Delivery’ provided by Amazon. Owing to the short lead time, delivery trucks may need to be dispatched immediately after, or even before, the order cutoff time, but some packages may not have been packed yet.

In urban areas, trucks deliver packages at delivery sites, which may be residential complexes or office buildings with multiple individual deliveries. Consequently, in overnight shipping and online shopping, a

* Corresponding author.
E-mail addresses: wliubi@connect.ust.hk (W. Liu), ldliu@ustc.edu.cn (L. Liu), ieemqi@ust.hk (X. Qi).

https://doi.org/10.1016/j.ejor.2024.05.025
Received 30 January 2023; Accepted 12 May 2024
Available online 15 May 2024

delivery site may have packages that are ready for dispatch before truck departure, as well as those that become available after truck departure. To achieve the on-time delivery of these late-available packages, drones can be scheduled to transport them to trucks en route, and then trucks make the final deliveries.

During this particular delivery process, from the perspective of drone resupply scheduling, trucks adhere to given routes for package delivery. Specifically, in overnight shipping, a delivery site typically has multiple individual deliveries; thus, it is likely to be visited daily. This means that the delivery company faces the same vehicle routing problem with respect to delivery sites. Delivery along routine routes is efficient as delivery staff are familiar with the routes and can optimise the delivery process. In online shopping, truck routing is essential to ensure on-time delivery of packages that are available before truck departure. Given the short lead time of on-time delivery, truck routes are always planned before loading packages onto trucks at the depot, which is done before making drone resupply schedules. In this case, trucks follow predetermined routes to deliver packages. Therefore, from the perspective of drone resupply scheduling, trucks make deliveries along given routes, regardless of whether trucks follow routine or predetermined routes.

In light of these considerations, our study investigates drone resupply for on-time package delivery along given truck routes. Trucks following given routes only visit delivery sites in a given sequence; however, they can wait at certain locations along the route to receive packages resupplied by drones, subject to delivery deadlines. Thus, our model incorporates the impact of drone resupply decisions on truck routing by fine-tuning the truck departure times.

The major contributions of this study are as follows:

(1) We propose a drone resupply model comprising multiple trucks and multiple drones for on-time package delivery. Considering practical operations in urban areas, we investigate the situation where trucks deliver packages to delivery sites along given routes with adjustable departure times from the depot and delivery sites.

(2) We formally define the problem, investigate certain optimality conditions, and develop two solution approaches. One is the formulation of a time-expanded network flow model with side constraints, which solves practical instances to optimality. The other is a greedy algorithm that can quickly generate effective solutions.

(3) We conduct extensive computational studies to evaluate the efficiency and effectiveness of the proposed model and solution approaches. Results show that the solution approaches can serve as effective decision-making tools for generating drone resupply schedules. Additionally, we derive managerial insights to assist logistics companies in making informed decisions, such as the fleet team configuration.

The remainder of this paper is organised as follows: In Section 2, we review the relevant literature. In Section 3, we formally define the problem and present certain optimality conditions. In Section 4, we develop solution approaches. In Section 5, we report on our computational studies. Finally, Section 6 concludes this work.

2. Literature review

We discuss studies closely related to using trucks and drones collaboratively in making last-mile deliveries. For a comprehensive review of hybrid truck-drone delivery systems, we refer to Madani and Ndiaye (2022).

2.1. Truck and drone coordinated systems

Various operational models have been proposed for truck and drone coordination. Such models can be classified into parallel and tandem systems (Murray & Chu, 2015). In a parallel system, deliveries are made by either drones or trucks. In a tandem system, trucks carry drones to different locations, and drones then make deliveries to the customers around each location.

For the truck-drone parallel system, Murray and Chu (2015) introduce a parallel drone scheduling Travelling Salesman Problem (TSP). In this problem, a truck runs a travelling salesman tour, and a drone performs one direct depot-to-customer operation on each dispatched trip, owing to its limited flight duration and payload capacity. This problem is then further extended by considering multiple trucks and drones (Nguyen et al., 2022; Saleu et al., 2022). Ham (2018) follows the same operational model, in which each drone only delivers one package on each trip, but it may pick up another package on the backhaul. Nguyen and Hà (2023) investigate a new variant, where multiple drones are combined to form a collective drone for transporting heavier items.

In the basic truck-drone tandem system, trucks serve as mobile depots and drones make deliveries. A drone is dispatched from the carrying truck to deliver packages to a customer, and then returns, either to the same dispatch location (Dukkanci et al., 2023; Erdoğan & Yıldırım, 2021), or to a different location on the truck route (Karak & Abdelghany, 2019; Poikonen & Golden, 2020). In more general models, both trucks and drones can serve customers. The flying sidekick travelling salesman problem proposed in Murray and Chu (2015), considers a single truck equipped with one drone, with the drone serving only one customer on each dispatched trip. Since then, many studies have been focused on this problem and its extensions. Some of the latest studies are Tamke and Buscher (2021), Tiniç et al. (2023), Zhou et al. (2023), and Kloster et al. (2023).

All the aforementioned studies involve drones making direct deliveries to customers. However, implementing these delivery models can be challenging owing to difficulties in identifying appropriate drop-off locations, particularly in densely populated areas.

2.2. Drone resupply

Recently, a new operational model, drone resupply, has been proposed for last-mile delivery. The novelty of drone resupply is that drones transport packages from a depot to delivery trucks, and then truck drivers make final deliveries to customers to ensure reliable delivery. Drone resupply has only been explored in a limited number of contexts.

The primary distinction between our study and other studies lies in the involved vehicles. Existing studies focus on addressing the drone resupply problem with a single truck and a single drone. Our study is the first to tackle the problem considering multiple trucks and multiple drones. Additionally, our study differs from the existing studies in other aspects, such as the motivations for adopting drone resupply, modelling methods, characteristics of truck routing, and algorithms for drone resupply, as outlined in Table 1.

Dayarian et al. (2020) address a drone resupply problem for dynamic same-day delivery, where packages dynamically arrive at the depot throughout the day. In their problem, a drone is used to repeatedly resupply packages from the depot to the delivery truck, and the truck dynamically re-optimises its route upon receiving each resupply, with the main objective of maximising the number of orders delivered within the delivery deadline. In contrast, our study focuses on an offline drone resupply scheduling problem, where all relevant information is revealed before making drone resupply decisions. Truck routes are given, and are not adjusted according to resupplied packages.

Pina-Pardo et al. (2021) use drone resupply to deliver packages with different release times. The drone is used to resupply newly available packages to the truck en route, to minimise the total delivery time. They decompose the problem into two subproblems: truck routing and drone resupply, and then make decisions successively. In the first stage, they decide on the TSP routing, in which the drone schedule is considered by using time windows. The start of the time window of each customer is the order release time plus the drone flying time from the depot. In the second stage, they use MILP formulation to decide on the drone

Table 1
Summary of drone resupply literature.

	Dayarian et al. (2020)	Pina-Pardo et al. (2021)	Dienstknecht et al. (2022)	This paper
Number of vehicles	single truck single drone	single truck single drone	single truck single drone	multiple trucks multiple drones
Truck routing characteristics	prize-collecting TSP with due dates	TSP with time windows	TSP considering drone flight cost	given routes with flexible departure times
Objectives	maximise the number of deliveries	minimise the delivery time	minimise the delivery cost	maximise the delivery profit
Key constraints	promised delivery time	different order release times	limited truck capacity	order ready times and delivery deadlines
Modelling methods	dynamic optimisation	MILP	MIP	network flow with side constraints
Algorithms for drone resupply	heuristic	solving MILP	dynamic programming	solving ILP & greedy heuristic

schedule, where the truck routing can be modified by adjusting the truck departure times.

Dienstknecht et al. (2022) employ drone resupply to handle packages left at the depot owing to the limited truck capacity. Not all packages can be loaded when the truck departs from the depot. Drones are then employed to resupply the remaining packages to the truck, with the goal of minimising the overall delivery cost, which consists of the truck driving cost, tour cost, and drone flight cost. They first use different algorithms, including genetic algorithms and construction heuristics, to address the TSP that considers the flight cost associated with drone resupply scheduling. Then, they solve the drone resupply scheduling problem, where dynamic programming is developed to find the optimal drone schedule for a given truck tour.

Our drone resupply motivation differs from those of Pina-Pardo et al. (2021) and Dienstknecht et al. (2022), although we all consider offline drone resupply problems. In their problems, all packages are required to be delivered. Conversely, given the short lead time for on-time delivery, our objective is to maximise the total delivery value of late-available packages that can be delivered within the promised deadline, without guaranteeing the successful delivery of all late-available packages. Consequently, these differences in the nature of the problem raise unique truck routing models and tailored algorithms to solve the drone resupply scheduling problem.

3. Problem description

In our on-time Delivery Assisted by Drone Resupply model (DADR), we consider a depot that uses a fleet of trucks, assisted by a group of drones, to deliver packages to its customers within a promised delivery deadline of time T. Most of the packages are ready for dispatch before the planned truck departure, with just a few Unavailable For Delivery (UFD) packages.

During the delivery process, trucks deliver packages at delivery sites along given routes. The route of truck k is denoted by $\mathcal{N}_k = \{(k,0),(k,1),\ldots,(k,i),\ldots,(k,n_k)\}$, where $(k,0)$ denotes the depot, and (k,i) is the ith delivery site visited by truck k. $b_{(k,i)}$ is the planned departure time from delivery site (k,i). Despite trucks visiting delivery sites in a given sequence, they may wait for a short time at the depot to load certain UFD packages, or at some delivery sites to get resupplied by drones. This means that truck departure times from the depot and delivery sites will be optimised together with the drone resupply schedule.

UFD packages become available for resupply when trucks are en route for delivering the onboard packages. Let $\mathcal{M} = \{1,2,\ldots,m\}$ denote the set of UFD packages. Each UFD package p is characterised by $((K_p,I_p),\tau_p,w_p)$, where (K_p,I_p) represents the delivery destination of UFD package p, namely, the I_p-th delivery site visited by truck K_p, τ_p is the time when the package is available for resupply, and w_p is the delivery value, indicating the profit of on-time delivery.

To deliver UFD packages on time, the depot arranges a fleet of drones, denoted by $\mathcal{H} = \{1,2,\ldots,H\}$, to resupply them to the respective trucks en route. Considering the limited payload capacity of drones, we assume that, on each trip, a drone can resupply a maximum of θ UFD packages to a truck at one of the delivery sites that are accessible by drones. Since most commercial delivery drones have adequate endurance to cover the round-trip flight distance, as elaborated in the computational study, we can address the concern of energy consumption by replacing the batteries of drones with fully charged ones each time we load packages onto drones at the depot. Let ϵ denote the time for swapping the battery and loading packages onto a drone, μ represent the time for transferring the carried packages from a drone to a truck, and $\alpha_{(k,i)}$ be the one-way time for a drone to travel between the depot and a delivery site (k,i).

We now use an example to illustrate the problem described above.

Example 1. Consider the problem with a single truck and a single drone. The truck is dispatched from the depot $(1,0)$ at 8:00 am and visits delivery sites $\{(1,1),(1,2),(1,3),(1,4),(1,5)\}$ sequentially, with the planned departure times being {8:30 am, 9:00 am, 9:40 am, 10:10 am, 10:40 am}, respectively. There are two UFD packages: $((1,3)$, 9:30 am, 0.5) and $((1,5)$, 10:00 am, 0.8), to be delivered by 11:00 am.

Fig. 1 shows a feasible drone resupply schedule in which both UFD packages are delivered on time, but in two different ways. Case 1: The UFD package for delivery site $(1,3)$ is directly resupplied to its delivery destination. Upon completing the delivery, both the truck and drone depart from delivery site $(1,3)$ simultaneously. Case 2: The UFD package for delivery site $(1,5)$ is resupplied to delivery site $(1,4)$, an earlier delivery site on the truck route, and then the truck heads to delivery site $(1,5)$ for the final delivery. Finally, the revised truck departure times are $\{$8:00 am, 8:30 am, 9:00 am, 9:50 am, 10:22 am, 10:52 am$\}$, respectively.

In the above schedules, the sequence for the truck visiting delivery sites remains the same, but the planned departure times are delayed due to truck waiting and package transferring. We define the truck waiting and delay times at delivery site (k,i) as $d_{(k,i)}$ and $t_{(k,i)}$, respectively. The delay time at delivery site (k,i) can be calculated by

$$t_{(k,i)} = \begin{cases} t_{(k,i-1)} + d_{(k,i)} + \mu & \text{if the truck is resupplied at } (k,i) \\ t_{(k,i-1)} & \text{otherwise} \end{cases},$$

where $t_{(k,0)} = d_{(k,0)}$, because UFD packages can be directly loaded onto the truck at the depot. Consequently, the revised departure time of truck k from delivery site (k,i) is $l_{(k,i)} = b_{(k,i)} + t_{(k,i)}$.

Next, we discuss the conditions for truck and drone rendezvous. Suppose that drone h meets truck k_1 at delivery site (k_1,i_1). By departing from (k_1,i_1) at the same time as truck k_1 at time $l_{(k_1,i_1)}$, and then flying back to the depot to resupply other packages, drone h is able to meet truck k_2 at delivery site (k_2,i_2) to resupply UFD package $p:((K_p,I_p),\tau_p,w_p)$, where $k_2 = K_p,\ i_2 \le I_p$, if and only if

$$\max\{l_{(k_1,i_1)}+\alpha_{(k_1,i_1)},\tau_p\}+\epsilon+\alpha_{(k_2,i_2)} \le b_{(k_2,i_2)}+t_{(k_2,i_2)}-\mu,\ t_{(k_2,i_2)} \le T-b_{(k_2,n_{k_2})}. \quad (1)$$

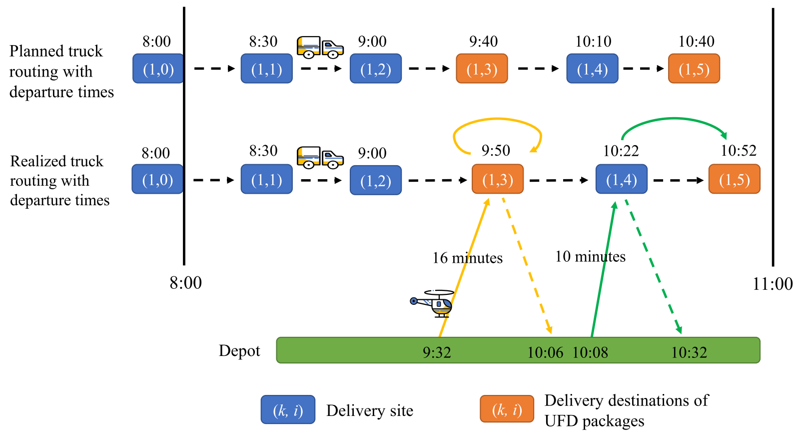


Fig. 1. Illustrative example of DADR.

For the special case of the first trip of drone h, the drone can meet truck k at delivery site (k,i) to resupply UFD package p, where $k = K_p$, $i \le I_p$, if and only if $\tau_p + \epsilon + \alpha_{(k,i)} \le b_{(k,i)} + t_{(k,i)} - \mu$, $t_{(k,i)} \le T - b_{(k,n_k)}$, where $T - b_{(k,n_k)}$ is the allowable maximum delay time of truck k.

Now we can formally state the problem. DADR: Given planned truck routes and all relevant information regarding UFD packages, we decide on the resupply schedule for each drone and revised truck departure times that maximise the resupply payoff, the total delivery value of the UFD packages that can be delivered on time.

We analyse the computational complexity of DADR and investigate one optimality property about the synchronisation of truck and drone rendezvous. All proofs are provided in the online appendix.

Theorem 1 (*Complexity*). *DADR is an $\mathcal{NP}$-hard problem.*

Theorem 2 (*Idleness Mutual Exclusion*). *Considering two consecutive trips of drone h to delivery sites (k_1, i_1) and (k_2, i_2), when drone h meets truck k_2 at (k_2, i_2) to resupply UFD package p, at least one of the following cases occurs:*

- *Case 1: $d_{(k_2,i_2)} = 0$;*
- *Case 2: $\max\{l_{(k_1,i_1)} + \alpha_{(k_1,i_1)}, \tau_p\} + \epsilon + \alpha_{(k_2,i_2)} > b_{(k_2,i_2)} + t_{(k_2,i_2-1)}$, where $t_{(k_2,i_2-1)}$ represents the delay time at the delivery site preceding (k_2, i_2).*

Theorem 2 shows that we can exclude the drone resupply trip in which the truck and the drone have some waiting/idle time. Specifically, Case 1 means that the waiting time of truck k_2 at the rendezvous location is zero. Case 2 means that drone h is dispatched once the UFD package to be resupplied on the next trip is ready. With this property and (1), the condition for truck and drone rendezvous, we can calculate the truck delay time for resupplying UFD package p at site (k_2, i_2) by $t_{(k_2,i_2)} = \max\{\max\{l_{(k_1,i_1)} + \alpha_{(k_1,i_1)}, \tau_p\} + \epsilon + \alpha_{(k_2,i_2)} - b_{(k_2,i_2)}, t_{(k_2,i_2-1)}\} + \mu$. If $t_{(k_2,i_2)} \le T - b_{(k_2,n_{k_2})}$, we say that (k_2, i_2) is a feasible rendezvous location for resupplying UFD package p on the next trip of drone h. Otherwise, drone h cannot catch up with truck k_2 at delivery site (k_2, i_2) to resupply UFD package p. Thus, given one UFD package, we can know the truck delay time at the rendezvous location directly, rather than having to try all possible values in the range. This finding can be applied to improving the efficiency of the heuristic algorithm.

4. Solution approaches

In this section, we present our solution approaches to the proposed problem. We develop a time-expanded network flow model with side constraints to solve the problem to optimality. We also design a greedy algorithm to quickly solve the problem.

4.1. Time-expanded network flow formulation

The DADR can be represented by a time-expanded network flow model with side constraints, which leads to an Integer Linear Programming (ILP) formulation.

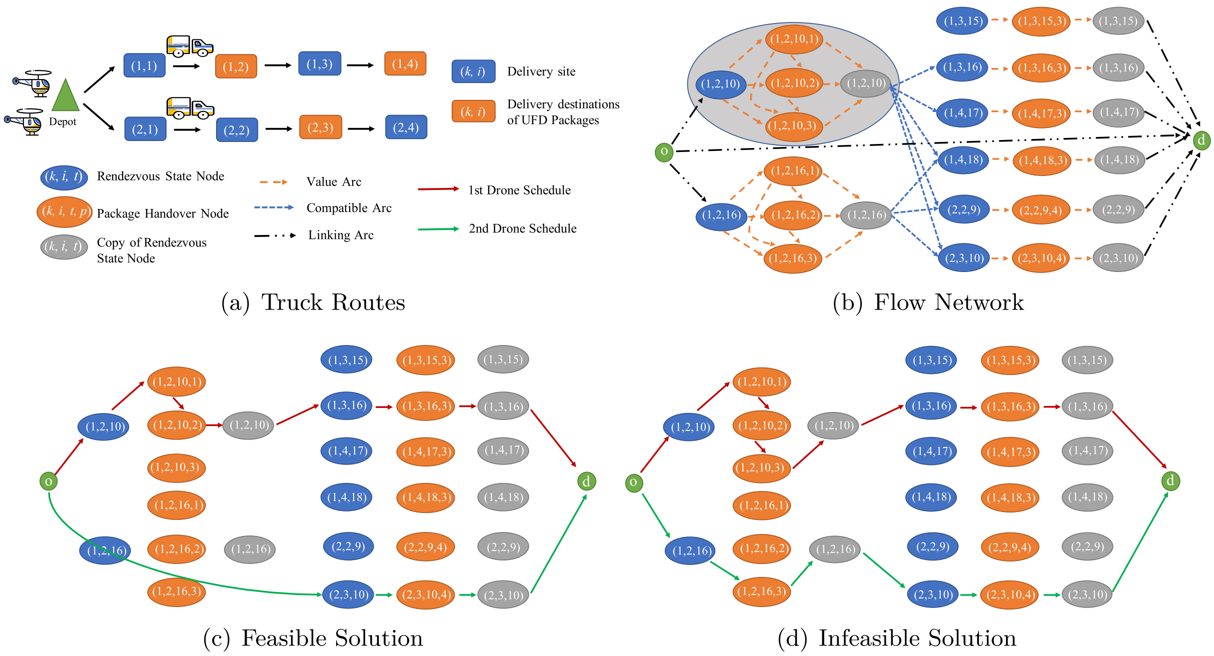
4.1.1. Time-expanded network flow model

We first identify the feasible rendezvous locations for resupplying each UFD package. For one UFD package $p : ((K_p, I_p), \tau_p, w_p)$, we can load it directly onto truck k at the depot, if and only if
(1) $k = K_p$, namely, the delivery destination (K_p, I_p) is visited by truck k, and
(2) there exists a $t_{(k,0)}$ such that $\tau_p \le b_{(k,0)} + t_{(k,0)}$, $t_{(k,0)} \le T - b_{(k,n_k)}$.
The above condition (2) ensures that the truck completes the delivery within the deadline despite being delayed by time $t_{(k,0)}$ at the depot. Alternatively, when we resupply UFD package p to the respective truck en route, a delivery site (k,i) can serve as the rendezvous location if and only if
(1) delivery site (k,i) is accessible for truck and drone rendezvous,
(2) $k = K_p$, $i \le I_p$, namely, (k,i) is prior to (K_p, I_p) on truck route $\mathcal{N}_k$, and
(3) there exists a $t_{(k,i)}$ such that $\tau_p + \epsilon + \alpha_{(k,i)} \le b_{(k,i)} + t_{(k,i)} - \mu$, $t_{(k,i)} \le T - b_{(k,n_k)}$.
All such delivery sites (k,i) and the feasible depot $(k,0)$ finally form a set of rendezvous locations for resupplying UFD package p, denoted as $\bar{\mathcal{F}}_p$. Let $\mathcal{F}$ be the set of rendezvous locations for resupplying any UFD package, that is, $\mathcal{F} = \bigcup_{p \in \mathcal{M}} \bar{\mathcal{F}}_p$.

In the network flow model, we use nodes to denote decisions concerning the resupply of UFD packages. We use Example 2, as shown in Fig. 2(a), to illustrate the design of the flow network.

Example 2. Consider a DADR with two trucks running along given routes $\mathcal{N}_1 = \{(1,0),(1,1),(1,2),(1,3),(1,4)\}$ and $\mathcal{N}_2 = \{(2,0),(2,1),(2,2),(2,3),(2,4)\}$, respectively, and two drones with a loading capacity of two packages. The maximum delay times allowed for trucks 1 and 2 are 20 and 10 minutes, respectively. The feasible rendezvous locations for resupplying UFD packages are $\bar{\mathcal{F}}_1 = \{(1,2)\}$, $\bar{\mathcal{F}}_2 = \{(1,2)\}$, $\bar{\mathcal{F}}_3 = \{(1,2),(1,3),(1,4)\}$, and $\bar{\mathcal{F}}_4 = \{(2,2),(2,3)\}$.

Now we show how to define the network. Consider one feasible rendezvous location $(k,i) \in \mathcal{F}$ at which truck k may be delayed by some time t, where $t \in \{\mu, \mu+1, \dots, T - b_{(k,n_k)}\}$, based on a 1-minute incremental discretisation. We say that (k,i,t) is a rendezvous state if there exists one UFD package p such that $(k,i) \in \bar{\mathcal{F}}_p$ and $\tau_p + \epsilon + \alpha_{(k,i)} \le b_{(k,i)} + t - \mu$. In particular, when $(k,i) = (k,0)$, $(k,0,t)$ is a rendezvous state if there exists one UFD package p such that $(k,0) \in \bar{\mathcal{F}}_p$ and $\tau_p \le b_{(k,0)} + t$.



(a) Truck Routes (b) Flow Network

(c) Feasible Solution (d) Infeasible Solution

Fig. 2. Illustrative example of time-expanded flow network.

For each rendezvous state (k, i, t), we define the following nodes in the time-expanded flow network:

• (k, i, t): Rendezvous state node, indicating that truck k is delayed by time t at delivery site (k, i) to get resupplied. Let $\tilde{\mathcal{R}}_{(k,i)}$ be the set of rendezvous state nodes for delivery site (k, i) with a different delay time t, and $\mathcal{R}$ be the set of all rendezvous state nodes in the flow network. For example, in Fig. 2(b), the blue node $(1, 2, 10)$ indicates that Truck 1 is delayed by 10 minutes at delivery site $(1, 2)$ to get resupplied.

• (k, i, t, p): Package handover node, representing that UFD package p is passed onto truck k at rendezvous state (k, i, t). For each rendezvous state (k, i, t), if delivery site $(k, i) \in \bar{\mathcal{F}}_p$ and $\tau_p + \epsilon + \alpha_{(k,i)} \leq b_{(k,i)} + t - \mu$, we define a package handover node (k, i, t, p). For the rendezvous state node $(k, 0, t)$, if $(k, 0) \in \bar{\mathcal{F}}_p$ and $\tau_p \leq b_{(k,0)} + t$, we define a package handover node $(k, 0, t, p)$. Let $\bar{\mathcal{S}}_p$ be the set of package handover nodes for UFD package p with different i and t, $\tilde{\mathcal{S}}_{(k,i,t)}$ denote the set of package handover nodes for the rendezvous state (k, i, t) with a different p, and $\mathcal{S}$ be the set of all package handover nodes in the flow network. For example, in Fig. 2(b), nodes $(1, 2, 10, 1)$, $(1, 2, 10, 2)$, and $(1, 2, 10, 3)$, represent that at rendezvous state $(1, 2, 10)$, UFD packages $\{1, 2, 3\}$ can be resupplied to Truck 1.

• $(k, i, t)'$: Copy of rendezvous state node (k, i, t) is introduced to reduce the number of arcs for modelling compatible drone trips in the flow network. To model two compatible drone trips represented by rendezvous state nodes (k_1, i_1, t_1) and (k_2, i_2, t_2), with node $(k_1, i_1, t_1)'$, we construct arcs from the nodes in $\tilde{\mathcal{S}}_{(k_1,i_1,t_1)}$ to this node and from this node to (k_2, i_2, t_2). Without node $(k_1, i_1, t_1)'$, arcs should be constructed from all nodes in $\tilde{\mathcal{S}}_{(k_1,i_1,t_1)}$ to (k_2, i_2, t_2). As shown in Fig. 2(b), to model the compatible trips associated with rendezvous state $(1, 2, 10)$, we construct three arcs leading to and five arcs departing from node $(1, 2, 10)'$. However, without $(1, 2, 10)'$, we need to construct arcs from nodes $\{(1, 2, 10, 1), (1, 2, 10, 2), (1, 2, 10, 3)\}$ to nodes $\{(1, 3, 16), (1, 4, 17), (1, 4, 18), (2, 2, 9), (2, 3, 10)\}$. This would result in a total of 15 arcs, instead of the eight arcs that are currently shown. The set of these nodes is denoted by $\mathcal{R}'$.

In addition, we add a source node o with H units of supply and a sink node d with H units of demand. Next, we introduce arcs:

• Value Arcs: used to represent the delivery values of UFD packages. First, for a package handover node $v : (k, i, t, p) \in \mathcal{S}$, we build an arc from the rendezvous state node $u : (k, i, t)$ to node v. Then, for any two package handover nodes $u : (k, i, t, p_1)$ and $v : (k, i, t, p_2)$ in $\tilde{\mathcal{S}}_{(k,i,t)}$, if $p_1 < p_2$, we introduce an arc from node u to node v. Finally, we develop an arc from node $v : (k, i, t, p)$ to the copy of rendezvous state node $(k, i, t)'$. In Fig. 2(b), the arcs from node $(1, 2, 10)$ to node $(1, 2, 10, 2)$ and from $(1, 2, 10, 1)$ to $(1, 2, 10, 2)$ are constructed to denote the delivery value of UFD Package 2.

• Compatible Arcs: defining feasible consecutive drone resupply trips. For any two rendezvous state nodes $u : (k_1, i_1, t_1)$ and $v : (k_2, i_2, t_2)$, we define an indicator parameter $f(u, v)$ to represent a relationship between nodes u and v. Specifically,

$$f(u, v) = \begin{cases} 1 & \text{if } k_1 = k_2,\ i_1 < i_2, \text{ and } t_1 + \mu > t_2 \\ 0 & \text{otherwise} \end{cases},$$

where $f(u, v) = 1$ means that the delay time at a rendezvous location that is visited earlier is longer than that of a later one on the same truck route, and thus nodes u and v cannot appear in a feasible solution simultaneously. If $f(u, v) = 0$ and $b_{(k_1,i_1)} + t_1 + \alpha_{(k_1,i_1)} + \epsilon + \alpha_{(k_2,i_2)} \leq b_{(k_2,i_2)} + t_2 - \mu$, the two resupply trips denoted by nodes u and v can be implemented in sequence by a drone, and we say nodes u and v are compatible. Under the condition of $(k_1, i_1, t_1) = (k_1, 0, t_1)$, nodes u and v are compatible if $f(u, v) = 0$. For each pair of compatible nodes u and v, we construct an arc from node u' to node v. In Fig. 2(b), the arc from node $(1, 2, 10)'$ to $(1, 3, 16)$ indicates that with the truck being delayed by 10 and 16 minutes at delivery sites $(1, 2)$ and $(1, 3)$, respectively, a drone can visit these two delivery sites consecutively.

• Linking Arcs: used to guarantee the flow from source node o to sink node d. For any rendezvous state node $v \in \mathcal{R}$, we introduce an arc (o, v). For each node $v' \in \mathcal{R}'$, we define an arc (v', d). Additionally, we create an arc (o, d) to represent the resupply schedule of drones that remains idle at the depot.

Up to this point, we have established a flow network $G = (\mathcal{V}, \mathcal{A})$, which is an acyclic-directed network with nodes $\mathcal{V} = \{\mathcal{R} \cup \mathcal{S} \cup \mathcal{R}' \cup \{o, d\}\}$ and arcs $\mathcal{A}$. Let c_{uv} denote the unit flow cost of arc $(u, v) \in \mathcal{A}$. For an arc $\{(u, v) | u : (k, i, t),\ v : (k, i, t, p)\}$, the unit flow cost $c_{uv} = -w_p$. Also, for an arc $\{(u, v) | u : (k, i, t, p_1),\ v : (k, i, t, p_2)\}$, the unit flow cost $c_{uv} = -w_{p_2}$. For the remaining arcs, the unit flow cost is zero. In this way, we can denote the delivery value of UFD packages by the negative value of the unit flow cost on certain arcs.

In the flow network, drone resupply schedules are represented by arc-disjoint flow paths from source node o to sink node d. For example, in Fig. 2(c), Drone 1 resupplies Packages $\{1, 2\}$ to delivery site $(1, 2)$ and Package 3 to delivery site $(1, 3)$, with Truck 1 being delayed by 10 and 16 minutes, respectively, and Drone 2 resupplies Package 4 to delivery site $(2, 3)$ with Truck 2 being delayed by 10 minutes. However, certain solutions that satisfy the flow conservation constraints are infeasible drone resupply schedules for various reasons, as shown in Fig. 2(d).

- *Conflict Departure Times*: Rendezvous state nodes $(1, 2, 10)$ and $(1, 2, 16)$ are visited. This means that there are two different departure times at delivery site $(1, 2)$, which conflicts with the definition of truck routing.
- *Conflict Delay Times*: Rendezvous state nodes $(1, 2, 16)$ and $(1, 3, 16)$ are visited, where the delay time of Truck 1 at rendezvous location $(1, 3)$ equals the delay time at $(1, 2)$, which is contradictory to the definition of delay time.
- *Repetitive Resupply*: Package handover nodes $(1, 2, 10, 3)$, $(1, 2, 16, 3)$, and $(1, 3, 16, 3)$ are visited, which indicates that Package 3 is resupplied twice at delivery site $(1, 2)$ and once at $(1, 3)$. This is impossible in the DADR.
- *Drone Capacity*: Package handover nodes $(1, 2, 10, 1)$, $(1, 2, 10, 2)$, and $(1, 2, 10, 3)$ are visited, which means that Drone 1 resupplies Packages $\{1, 2, 3\}$ at delivery site $(1, 2)$. This is impossible when the drone capacity is two packages.

To deal with the above issues, we introduce side constraints in the formulation of the time-expanded network flow model, which will be discussed next.

4.1.2. ILP formulation

Given the time-expanded flow network, where drone resupply schedules are represented by arc-disjoint flow paths from the source node to the sink node, we have the following decision variables:

- x_{uv}: A non-negative integer variable for arc $(u, v) \in \mathcal{A}$, indicating the flow on this arc.
- y_v: A binary variable for rendezvous state node $v : (k, i, t) \in \mathcal{R}$. If truck k is delayed by time t at delivery site (k, i) to get resupplied, $y_v = 1$; otherwise, $y_v = 0$.
- z_v: A binary variable for package handover node $v : (k, i, t, p) \in \mathcal{S}$. Specifically, if UFD package p is resupplied at rendezvous state (k, i, t), $z_v = 1$; otherwise, $z_v = 0$.

With the above notations, the DADR can be regarded as a minimum cost network flow problem with side constraints, which has the following ILP formulation.

$$\min \sum_{(u,v)\in\mathcal{A}} c_{uv} x_{uv} \tag{2}$$

The objective function (2) is to minimise the total flow cost in the flow network. Recall that in the flow network, the delivery value of UFD packages is represented by the negative value of the unit flow cost on certain arcs. Thus, the objective is equivalent to maximising the resupply payoff for the DADR, the total delivery value of the UFD packages that can be delivered on time.

Next, we introduce the involved constraints in the ILP formulation.

Constraints (3)–(5) are developed to maintain the connectivity of the drone resupply schedules. Specifically, Constraints (3) and (4) ensure that all drones begin and end their resupply schedules at the depot, respectively. Constraints (5) are the flow conservation conditions for the nodes in the flow network, excluding the source and sink nodes.

$$\sum_{v:(o,v)\in\mathcal{A}} x_{ov} = H, \tag{3}$$

$$\sum_{v:(v,d)\in\mathcal{A}} x_{vd} = H, \tag{4}$$

$$\sum_{u:(u,v)\in\mathcal{A}} x_{uv} = \sum_{u:(v,u)\in\mathcal{A}} x_{vu} \qquad \forall v \in \{\mathcal{V} \setminus \{o, d\}\}. \tag{5}$$

Constraints (6) are used to prevent conflicting departure times. Specifically, for each delivery site $(k, i) \in \mathcal{F}$, at most one of the rendezvous state nodes $v : (k, i, t) \in \tilde{\mathcal{R}}_{(k,i)}$ is visited.

$$\sum_{v\in\tilde{\mathcal{R}}_{(k,i)}} y_v \le 1 \qquad \forall (k, i) \in \mathcal{F}. \tag{6}$$

Constraints (7) deal with conflicting delay times. In a feasible solution, the delay time at a later rendezvous location should be at least time μ longer than that at an earlier rendezvous location visited by the same truck. This means that nodes $u : (k_1, i_1, t_1)$ and $v : (k_2, i_2, t_2)$ cannot be visited simultaneously if $k_1 = k_2,\ i_1 < i_2,$ and $t_1 + \mu > t_2$.

$$y_u + y_v \le 1 \qquad \forall u, v \in \mathcal{R}\ \&\ f(u, v) = 1. \tag{7}$$

Constraints (8) address the issue of repetitive resupply. For each UFD package p, at most one of the nodes $v : (k, i, t, p) \in \bar{\mathcal{S}}_p$ is visited.

$$\sum_{v\in\bar{\mathcal{S}}_p} z_v \le 1 \qquad \forall p \in \mathcal{M}. \tag{8}$$

Constraints (9) specify the conditions on drone capacity. For each rendezvous state node $v : (k, i, t)$ where $i \neq 0$, at most θ of the package handover nodes $u : (k, i, t, p) \in \tilde{\mathcal{S}}_v$ are visited.

$$\sum_{u\in\tilde{\mathcal{S}}_v} z_u \le \theta \qquad \forall v : (k, i, t) \in \mathcal{R}\ \&\ i \neq 0. \tag{9}$$

Constraints (10) define the relationships between the x-variables and the corresponding y-variables, and between the x-variables and the associated z-variables. These constraints highlight the connection between node selection and network flow.

$$\sum_{u:(u,v)\in\mathcal{A}} x_{uv} = y_v \quad \forall v \in \mathcal{R}, \qquad \sum_{u:(u,v)\in\mathcal{A}} x_{uv} = z_v \quad \forall v \in \mathcal{S}. \tag{10}$$

Constraints (11) give the domains of the decision variables.

$$x_{uv} \in \mathbb{N} \ \ \forall (u, v) \in \mathcal{A}, \quad y_v \in \{0, 1\} \ \ \forall v \in \mathcal{R}, \quad z_v \in \{0, 1\} \ \ \forall v \in \mathcal{S}. \tag{11}$$

4.2. Heuristic algorithm

We also design a greedy algorithm to quickly solve the problem. This algorithm mainly involves three types of decisions: sorting UFD packages to decide their resupply sequence, assigning UFD packages to the currently available drone for resupply, and deleting certain UFD packages to save some time for the current resupply.

Sorting UFD packages to decide the resupply sequence. For two delivery sites having UFD packages served by the same truck, the UFD package of the prior delivery site on the truck route should be resupplied earlier. Thus, we define the delivery urgency index of UFD package $p : ((K_p, I_p), \tau_p, w_p)$ as $\lambda_p = b_{(K_p,I_p)} + T - b_{(K_p,n_{K_p})}$, which is the latest time for delivering the package. A smaller λ_p implies an earlier time for resupplying package p, and only the UFD packages that are ready before the drone departs can be loaded for resupply. Considering these aspects, we first sort the UFD packages in a non-decreasing order based on their delivery urgency indices, and then we further sort them in a non-decreasing order based on their ready times. Let Q denote the list of sorted UFD packages.

Assigning UFD packages to the currently available drone. In each resupply trip, a drone rendezvous with a truck at one delivery site, so only UFD packages for delivery sites that are visited by the same truck can be assigned to the drone. Such a set of UFD packages can be regarded as a bundle or a complex UFD package. For a bundle $B = \{q_1, q_2, \dots, q_i\}$, where $i \le \theta$ and $q_1, q_2, \dots, q_i$ are sequentially selected from Q, we can denote this bundle as $B : ((K_B, I_B), \tau_B, w_B)$, where

$$K_B = K_{q_1} = \cdots = K_{q_i},\ \ I_B = I_{q_1},\ \ \tau_B = \max_{q_j\in B}\{\tau_{q_j}\},\ \ w_B = \sum_{q_j\in B} w_{q_j}, \tag{12}$$

and the delivery urgency index of bundle B is

$$\lambda_B = b_{(K_B,I_B)} + T - b_{(K_B,n_{K_B})}. \tag{13}$$

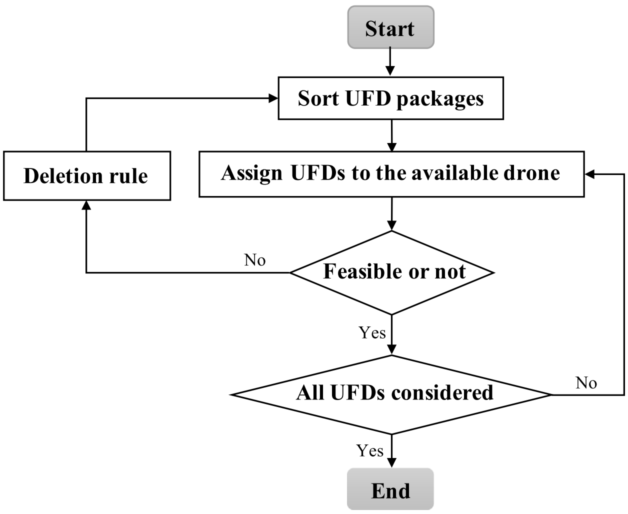


Fig. 3. The flowchart of the heuristic algorithm.

When drone h is available at time r_h, following the sequence specified by Q, we check the unassigned UFD packages that are ready before $r_h + \Delta - \epsilon$, where Δ is the allowable maximum waiting time of drones at the depot, and combine these orders into different bundles. The bundle B with the smallest λ_B should be assigned to the drone. This bundle is expected to be resupplied to truck $k = K_B$ at the delivery site (k, e), which allows the drone to return to the depot as soon as possible. That is, for the respective truck k, $(k, e) = \arg\min_{(k,j), j \le I_B} \{ b_{(k,j)} + t_{(k,j)} + \alpha_{(k,j)} | \max\{r_h, \tau_B\} + \epsilon + \alpha_{(k,j)} \le b_{(k,j)} + t_{(k,j)} - \mu,\ D_k + \mu \le t_{(k,j)} \le T - b_{(k,n_k)} \}$, where the first condition guarantees that the drone can catch up with truck k and the second one ensures the feasibility of the delay time at this rendezvous location when the truck has been delayed by D_k minutes. By Theorem 2, the condition can be simplified to

$$(k, e) = \arg\min_{(k,j), j \le I_B} \Big\{ b_{(k,j)} + t_{(k,j)} + \alpha_{(k,j)} | \; t_{(k,j)} = \max\{\max\{r_h, \tau_B\} + \epsilon + \alpha_{(k,j)} - b_{(k,j)},\ D_k\} + \mu,\ t_{(k,j)} \le T - b_{(k,n_k)} \Big\}. \tag{14}$$

If delivery site (k, e) exists, the current resupply is feasible. We update the delay time of truck k, the time when drone h returns to the depot, and the set of successfully resupplied UFD packages, which is denoted as Q', in the following way:

$$D_k = t_{(k,e)}, \quad r_h = b_{(k,e)} + t_{(k,e)} + \alpha_{(k,e)}, \quad Q' = Q' \cup B. \tag{15}$$

If delivery site (k, e) does not exist, we remove the most recently available package from bundle B, and then assign the remaining packages in B to drone h until they can be resupplied, or until no UFD packages can be resupplied successfully.

Deleting certain UFD packages to save time for the current resupply. If all packages in bundle B cannot be resupplied successfully by the available drone, the infeasibility of the current resupply is attributed to the fact that the most urgent package q_1 in bundle B cannot be resupplied in a timely manner. Therefore, to save some time for resupplying q_1, we need to delete a less valuable package that has been resupplied successfully at an earlier time. Specifically, we try to find one package $q_{j^*} \in Q'$ satisfying the following.
(1) $w_{q_{j^*}} < w_{q_1}$;
(2) q_1 can be resupplied successfully if q_{j^*} is deleted from Q.
If q_{j^*} exists, we delete it from Q; otherwise, we delete q_1. After deleting the specified package, we re-solve the drone resupply scheduling problem for the updated Q.

To conclude this part, we present a flowchart in Fig. 3 to show the main operations of the heuristic algorithm. We sort the UFD packages based on their delivery urgency indices and ready times, to determine their resupply sequence. When a drone is available at the depot, we assign packages to it and check whether the resupply of these packages is feasible or not. If feasible, we continue to assign packages to the next available drone for resupply until all packages are considered. If the current resupply is infeasible, we delete a specific package and re-solve the drone resupply scheduling problem. After obtaining the drone resupply schedules under scenarios with a different Δ, we select the one that achieves the largest resupply payoff. The pseudo-code and time complexity of the greedy algorithm can be referred to in the online appendix.

5. Computational study

We conduct a computational study to validate our model and algorithms. We first evaluate the efficiency and effectiveness of the proposed solution approaches. Thereafter, we measure the performance of DADR in scenarios with different numbers of drones and different drone capacities.

5.1. Design of computational study

We assess the scalability of our model and solution approaches using three delivery networks of different sizes: *E-n101-k8*, *M-n151-k12*, and *M-n200-k16* (provided in Christofides and Eilon (1969), Christofides et al. (1979)). In these delivery networks, there are around 100, 150, and 200 delivery sites, respectively. To align with the practical delivery range of a logistics company, we assume that the distance unit of the coordinate system is one-third of a kilometre. Based on this unit, we measure the travelling time of trucks using Manhattan distance and the flying time of drones using Euclidean distance. We separately employ 10, 15, and 20 trucks in these delivery networks to ensure that packages ready before the planned truck departure are delivered within the promised time. To label the number of trucks used in each delivery network, we rename them as *E-n101-k10*, *M-n151-k15*, and *M-n200-k20*, respectively.

The deadline for on-time delivery is $T = 120$ minutes. The service times at delivery sites are randomly generated within $[3, 7]$ minutes to capture different volumes of delivery demands at delivery sites. The travelling speed of trucks is 35 kilometres per hour. Based on these parameters, we generate the planned truck routes and departure times at delivery sites using the VRP Spreadsheet Solver developed by Erdoğan (2017) and accessible at https://people.bath.ac.uk/ge277/vrp-spreadsheet-solver/.

UFD packages become randomly available at the depot during the interval $[30, 120]$ minutes. The delivery destinations of these packages are randomly generated from the set of delivery sites in the delivery network. The delivery value of each UFD package is randomly selected from $\{0.1, 0.2, \ldots, 0.9\}$. We evaluate our model and solution approaches considering different numbers of UFD packages in various delivery networks. Specifically, in delivery networks of small, medium, and large sizes, we consider $m = \{10, 15, 20, 25, 30\}$, $m = \{10, 20, 30, 40\}$, and $m = \{10, 20, 30, 40, 50\}$, respectively. The number of UFD packages is typically not high in practice. Thus, considering these ranges of UFD package numbers would be sufficient for the evaluation of our model and solution approaches.

Next, we introduce the parameters related to the drones. We set the speed of drones at 45 kilometres per hour, which is also used in Pina-Pardo et al. (2021). By referring to the heavy lift drones made by an American manufacturing company, we note that the payload of delivery drones can range from 1.5 to 11 kilograms and the endurance of the equipped battery ranges from 40 to 70 minutes. To characterise the range of payload, we consider different drone capacities, that is, $\theta = \{1, 2, 3, 4\}$. The batteries enable the drones to cover the round-trip

Table 2
The Scalability of the time-expanded network flow model.

Delivery Network	m	Constructing the flow network						Solving the ILP formulation		
		$Node_{num}$	Arc_{num}	$\mathcal{R}^1_{num}$	$\mathcal{R}^2_{num}$	$\mathcal{R}^3_{num}$	CPU (s)	Var_{num}	$Constr_{num}$	CPU (s)
E-n101-k10	10	1360	34,232	246	48	6	0.23	35,093	31,572	2.08
	15	1836	57,345	315	81	18	0.39	58,535	42,152	4.43
	20	2227	77,219	348	126	31	0.58	78,698	50,105	7.21
	25	2539	92,679	374	156	48	0.74	94,400	55,840	12.09
	30	2807	105,631	377	184	67	0.88	107,567	60,605	18.52
M-n151-k15	10	1828	76,851	341	37	2	0.40	77,975	59,842	6.19
	20	3012	190,670	528	114	14	1.03	192,579	95,775	13.27
	30	3914	290,571	643	199	40	1.77	293,127	122,228	26.81
	40	4590	352,151	682	279	80	2.45	355,244	138,141	62.07
M-n200-k20	10	1842	87,804	337	37	1	0.41	88,932	47,477	2.92
	20	3189	241,963	572	107	11	1.18	243,968	79,448	8.78
	30	4237	393,201	718	193	31	2.18	395,927	104,502	18.33
	40	5082	515,573	796	286	68	3.14	518,924	122,624	39.49
	50	5794	612,084	853	358	114	3.98	615,993	136,516	122.62

distance. When a drone returns to the depot, its battery is replaced with a fully charged one and loaded with packages, which takes $\epsilon = 2$ minutes. Transferring the resupplied UFD packages takes $\mu = 2$ minutes.

For each group of computational tests, we consider 100 instances, each involving different UFD packages. The computational experiments are run on a computer with a 3.20 GHz Intel-Core i7–8700 processor and 32 GB of RAM. The ILP formulation is solved by Gurobi Optimiser 9.1.1, and all algorithms are implemented in Python Release 3.7.4.

5.2. Efficiency and effectiveness of solution approaches

In this part, we assess the efficiency of the time-expanded network flow model with side constraints and check the effectiveness of the greedy algorithm across different delivery networks.

5.2.1. Computation time of the time-expanded network flow formulation

The computation time is the total time for constructing the time-expanded flow network and solving the ILP formulation. To characterise the scalability of the time-expanded network flow model, we measure the numbers of nodes and arcs in the flow network, denoted as $Node_{num}$ and Arc_{num}, as well as the numbers of decision variables and constraints involved in the ILP formulation, which is denoted as Var_{num} and $Constr_{num}$.

Table 2 presents the scalability of the time-expanded network flow model across different delivery networks with varying numbers of UFD packages. In the Table, $\mathcal{R}^1_{num}$, $\mathcal{R}^2_{num}$, and $\mathcal{R}^3_{num}$ denote the numbers of rendezvous state nodes having only one, two, and three package handover nodes connected, respectively. CPU represents the average computation time for either constructing the flow network or solving the ILP formulation.

We can observe from Table 2 that the computation time in a specific delivery network increases as the number of UFD packages increases. In delivery network *M-n200-k20*, considering 10 UFD packages results in $1,842$ nodes and $87,804$ arcs in the time-expanded flow network. When 50 UFD packages are involved, the number of nodes and arcs increases to $5,794$ and $612,084$, respectively. Consequently, substantial decision variables and constraints are introduced in the ILP formulation, making the resolution of the ILP formulation the major factor that contributes to the computation time required to solve the instances with a large number of UFD packages.

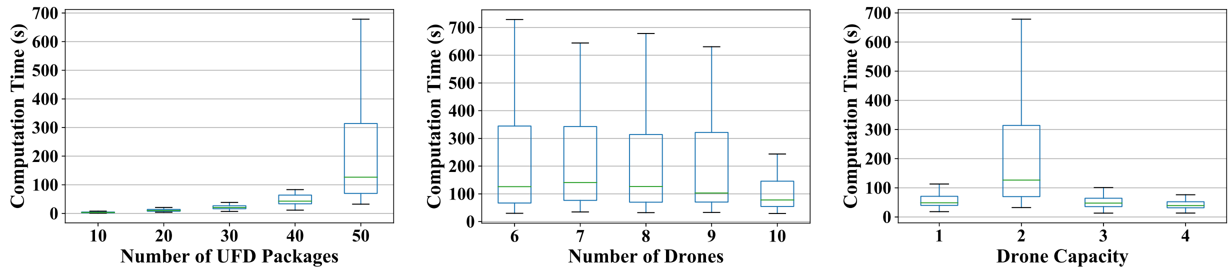
Additionally, we can see that the instances in these delivery networks can be solved to optimality with an average computation time of around 20 seconds, 1 minute, and 2 minutes, respectively. Also, with the same number of UFD packages, the average computation time does not increase significantly with the size of the delivery network. For example, in the case of 30 UFD packages, the flow network for *M-n200-k20* has more nodes and arcs than *M-n151-k15*, but the ILP formulation for *M-n200-k20* has fewer constraints than *M-n151-k15*. As a result, the computation time for solving the ILP formulation for a larger delivery network may be shorter. This is due to differences in the location of packages' destinations and truck routing across delivery networks.

Fig. 4 shows the distribution of computation times for various instances in delivery network *M-n200-k20*. The results in delivery networks *E-n101-k10* and *M-n151-k15* are depicted in Figures A.1 and A.2. The instances in small, medium, and large delivery networks can be solved to optimality within 2, 6, and 12 minutes, respectively. We also observe that the changes in the computation time remain consistent across different delivery networks, when varying numbers of UFD packages, drones, and drone capacities are considered. In the following, we focus on analysing the results in delivery network *M-n200-k20*.

Fig. 4(a) shows the distribution of the computation time for instances with different numbers of UFD packages. We find that the computation time increases as the number of UFD packages increases. For the instances with 50 UFD packages, the computation time increases more significantly, and some instances cannot be solved to optimality within 5 minutes. However, even in this scenario, the average computation time is approximately 2 minutes. This result justifies the efficiency of the time-expanded network flow formulation.

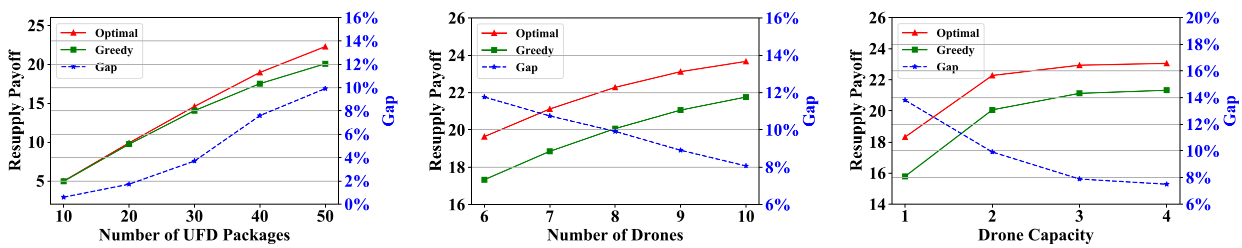
Fig. 4(b) presents the distribution of the computation time when different numbers of drones are employed to handle a certain number of UFD packages, namely, $m = 50$. As more drones are employed, the average computation time first increases slightly and then decreases. Some possible reasons for this could be as follows. The number of drones is the upper bound of feasible flow, which determines the number of flow paths from the source node to the sink node in the optimal solution. Thus, with more drones employed, for example, 7 drones used in *M-n200-k20*, more computation time is needed to find more flow paths. However, when we employ a larger fleet of drones, for example, $\{9, 10\}$, almost all UFD packages can be resupplied successfully, meaning that the optimal drone schedule may not be unique, and thus it tends to be easier to find the optimal solution.

Fig. 4(c) describes the distribution of the computation times when considering different drone capacities. We note that the computation time initially increases and then decreases, as the drone capacity is expanded. Additionally, it is observed that the computation time for instances with $\theta = 2$ is much more significant. The possible reason for this can be explained as follows. With a larger drone capacity, the number of feasible flow paths representing drone resupply schedules increases, taking more time to solve the ILP formulation. As the drone capacity is expanded, more of the constraints (9) in the ILP formulation become non-binding. From Table 2, we can see that for instances involving 50 UFD packages in delivery network *M-n200-k20*, $\mathcal{R}^1_{num} = 853$, $\mathcal{R}^2_{num} = 358$, and $\mathcal{R}^3_{num} = 114$. When $\theta = 2$, only 853 out of constraints (9) are non-binding. However, when $\theta = 3$, the constraints (9) for the rendezvous state nodes having one, two, and three package handover nodes are automatically satisfied, where $1,211$ out of constraints (9)



(a) $H = 8$ and $\theta = 2$ (b) $m = 50$ and $\theta = 2$ (c) $m = 50$ and $H = 8$

Fig. 4. Computation time of the time-expanded network flow formulation in *M-n200-k20*.



(a) $H = 8$ and $\theta = 2$ (b) $m = 50$ and $\theta = 2$ (c) $m = 50$ and $H = 8$

Fig. 5. Optimality gap of the greedy algorithm in delivery network *M-n200-k20*.

become non-binding. Therefore, compared with $\theta = 2$, it becomes much easier to find a feasible solution to the ILP formulation when $\theta = 3$.

5.2.2. Optimality gap of the greedy algorithm

After noting that each instance could be solved within one second via the greedy algorithm, we shift our focus to checking the computational effectiveness of the greedy algorithm by measuring the obtained resupply payoff. Further, to compare the effectiveness of the greedy algorithm and the time-expanded network flow formulation, we define the optimality gap as

$$gap = \frac{(optimal\ payoff - greedy\ payoff)}{optimal\ payoff} \times 100\%,$$

where the optimal payoff is obtained by solving the ILP formulation. Fig. 5 shows the computational performance of the greedy algorithm for various instances in delivery network *M-n200-k20*. We can see that the results in *E-n101-k10* and *M-n151-k15*, presented in Figures A.3 and A.4, are consistent with those in *M-n200-k20*. This indicates that the greedy algorithm performs equally well across different delivery networks.

Fig. 5(a) shows the changes in the resupply payoff and the optimality gap when different numbers of UFD packages are considered. We note that, as the number of UFD packages increases, the optimality gap rises slightly and reaches the maximum value of 10% with 50 UFD packages. When eight drones with a capacity of two packages are employed to resupply a small number of UFD packages, for example, $m = \{10, 20\}$, all the UFD packages can be resupplied under the optimal scheme. This means that it would be easy for the greedy algorithm to find an effective resupply schedule, thus resulting in a small optimality gap of around 2%. As the number of UFD packages increases, more effective resupply schedules are needed, leading to an increasing optimality gap. Despite a relatively large gap for some instances, we need to point out that, the number of UFD packages is typically small, for example, $m = \{10, 20, 30\}$, rather than reaching as many as 50. The resulting optimality gap of less than 4% confirms the practical usage of the greedy algorithm.

Fig. 5(b) presents the resupply payoffs and the optimality gap when different numbers of drones are employed. In general, the optimality gap decreases as more drones are employed. This is because, with a large fleet of drones, for example, $H = \{9, 10\}$, only a few UFD packages that are less valuable would be left at the depot. This somehow coincides with the deletion rule in the greedy algorithm, namely, that when the current resupply is infeasible, the UFD package having a smaller delivery value tends to be deleted.

Fig. 5(c) shows that even when a substantial number of UFD packages are considered, an optimality gap of 14% is almost the worst case. We also note that when drones with a larger capacity are employed, the optimality gap decreases. The reason is that with a large drone capacity, it is easy for the greedy algorithm to generate a resupply schedule such that most of the UFD packages can be resupplied successfully. For example, eight drones with a relatively large loading capacity, such as $\theta = \{3, 4\}$, are enough to resupply most of the UFD packages under the greedy scheme, leading to a small optimality gap of 8%.

5.3. Evaluating the performance of DADR

We evaluate the performance of DADR in terms of the resupply payoff, the level of occupancy of the employed drones, namely, the drone utilisation rate, and the average truck delay time. In particular, the drone utilisation rate is defined as follows:

$$utilisation\ rate = \frac{average\ flying\ time\ per\ drone}{service\ period} \times 100\%,$$

where the service period is 90 minutes according to the parameter setting.

5.3.1. Number of drones

First, we discuss the impact of the number of drones on the performance of DADR. Table 3 shows the achieved resupply payoff, drone utilisation rate, and average truck delay time when different numbers of drones are employed.

As expected, employing an additional drone leads to a higher resupply payoff, with the average marginal benefit decreasing. Besides, with more drones employed, the drone utilisation rate decreases. This is because the same workload of resupplying UFD packages is shared by more drones, indicating that the average number of resupply trips

Table 3
Performance of DADR using different numbers of drones with $\theta = 2$.

H	Resupply payoff			Utilisation rate (%)			Delay time (min)		
	$m = 10$	$m = 30$	$m = 50$	$m = 10$	$m = 30$	$m = 50$	$m = 10$	$m = 30$	$m = 50$
6	4.98	13.89	19.64	26.83	66.01	69.35	7.06	13.44	13.77
7	4.98	14.38	21.12	21.17	62.68	68.98	7.10	14.26	14.95
8	4.98	14.57	22.28	17.83	57.93	67.85	7.17	14.59	16.12
9	4.98	14.67	23.12	15.01	53.38	66.62	7.25	14.75	16.87
10	4.98	14.69	23.67	12.89	48.19	64.70	7.29	14.71	17.40

Table 4
Performance of DADR using eight drones with different capacities.

θ	Resupply payoff			Utilisation rate (%)			Delay time (min)		
	$m = 10$	$m = 30$	$m = 50$	$m = 10$	$m = 30$	$m = 50$	$m = 10$	$m = 30$	$m = 50$
1	4.96	13.83	18.32	20.10	67.00	71.23	7.18	14.43	14.92
2	4.98	14.58	22.28	17.83	57.93	67.85	7.17	14.59	16.12
3	4.98	14.62	22.94	17.90	55.65	66.11	7.17	14.57	16.47
4	4.98	14.63	23.06	17.90	55.53	65.29	7.17	14.51	16.56

implemented by each drone is reduced, leading to a lower drone utilisation rate. In addition, trucks will experience more serious delays when more drones are involved. Since using more drones to resupply UFD packages would incur more truck-drone rendezvous, more delay time is required to transfer the resupplied UFD packages.

Increasing the number of drones from 6 to 10 to resupply 50 UFD packages can improve the resupply payoff by approximately 20%. However, this improvement comes at the cost of a 5% decrease in drone utilisation rate and each truck experiencing an additional 4-minute delay. This result indicates a trade-off between resupply efficiency and vehicle utilisation. It is essential to weigh the resupply payoff against the potential drawbacks of lower drone utilisation rates and additional truck delays. Ultimately, the optimal number of drones to be employed may depend on the priority targets of the logistics company.

From Table 3, we also note that the extent of the reduction in drone utilisation rate differs across scenarios with varying numbers of UFD packages. This phenomenon can be easily explained as follows. When a few UFD packages, such as $m = 10$, are considered, six drones are enough to have almost all UFD packages resupplied on time. Therefore, when an additional drone is used, the fleet of drones would share the same amount of workload, thus leading to a significantly lower utilisation rate. However, in scenarios with a large number of UFD packages, such as 50, an additional drone will resupply extra UFD packages. This means that a larger amount of workload is shared among the drones, thus the utilisation rate remains relatively stable. To get resupplied by these busy drones, trucks must wait longer at the rendezvous locations, indicating more serious delays.

5.3.2. *Drone capacity*

Next, we examine the impact of drone capacity on the performance of DADR. Table 4 illustrates the resupply payoff, drone utilisation rate, and the average truck delay time when considering different drone loading capacities in scenarios with different numbers of UFD packages.

Intuitively, with a larger drone capacity, a greater resupply payoff is achieved, but the marginal benefit decreases significantly. For example, in the scenario with 50 UFD packages, the payoff achieved by using drones with a capacity of two packages is 22.28, which is much higher than 18.32, the payoff when using drones with a capacity of one package. However, when we use drones that carry three packages, the resupply payoff is 22.94, which is only slightly higher than that of using drones with a capacity of two packages. This is because even with a large capacity, such as $\theta = \{3, 4\}$, drones are likely to resupply two UFD packages on each trip. The above results indicate that using drones with a large capacity is not really that necessary for on-time DADR with multiple trucks and drones.

From Table 4, we also find that with a larger capacity, the drone utilisation rate tends to be lower. To travel with a full load, a drone with a larger capacity will wait a long time at the depot for more UFD packages to become available. At the same time, when more UFD packages are resupplied on a single trip, fewer resupply trips may be needed. Therefore, with a larger capacity, the total flying time of drones decreases, which leads to a lower drone utilisation rate.

Table 5
Resupply payoffs using different numbers of drones and drone capacities.

H	Resupply payoff							
	$m = 30$				$m = 50$			
	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$
5	11.30	13.02	13.25	13.28	13.75	17.81	18.86	19.11
6	12.39	13.44	14.03	14.05	15.48	19.64	20.62	20.82
7	13.23	14.38	14.45	14.46	17.00	21.12	21.95	22.11
8	13.83	14.58	14.62	14.63	18.32	22.28	22.94	23.06
9	14.17	14.67	14.69	14.70	19.46	23.12	23.60	23.67
10	14.32	14.69	14.71	14.71	20.44	23.67	23.96	24.01
11	14.36	14.69	14.71	14.72	21.25	23.94	24.13	24.17

Additionally, we note that the truck delay time remains relatively stable as the drone capacity is expanded. When drones have a larger capacity, the number of resupply trips will be reduced, maybe reducing the total time for transferring packages from drones to trucks. On the other hand, drones may need to wait longer at the depot to have more UFD packages resupplied on a single trip. Although such a consideration may result in a higher resupply payoff, it can also cause trucks to wait for a longer time at rendezvous locations. As a result, influenced by these two aspects, truck delay time remains relatively stable when a larger drone capacity is considered.

5.3.3. *Number of drones and drone capacity*

So far, we have discussed the impact of the number of drones and drone capacity separately. The results reveal that the marginal benefit of employing more drones or expanding the drone capacity in improving the resupply payoff decreases. Here, we further compare the effectiveness of employing more drones and expanding the drone capacity. Table 5 shows the resulting resupply payoffs when using different numbers of drones and various drone capacities.

From Table 5, we find that to improve the resupply payoff in scenarios with a few drones, employing additional drones is more effective than expanding the drone capacity. In contrast, when a large number of drones have already been employed, expanding the drone capacity tends to be more effective. For example, in the scenario involving five drones with a capacity of one package to resupply 30 UFD packages, by expanding the drone capacity to four packages, the payoff reaches 13.28. The improvement is close to that when employing two more drones. In the scenario involving seven drones with a capacity of one package to resupply 30 UFD packages, by expanding the drone capacity to four packages, the payoff reaches 14.46, which cannot be achieved by employing four additional drones. This phenomenon is primarily caused by the decreasing marginal benefit of using additional drones, where the advantage of adding an extra drone to an already large fleet becomes insignificant.

Furthermore, we note that solely applying one operational measure to improve the resupply payoff, by either employing extra drones or expanding the drone capacity, is sub-effective. For example, in the scenario where six drones with a capacity of one package are employed to resupply 50 UFD packages, by expanding the drone capacity to four packages and employing as many drones as 11, the payoff can only be improved to 20.82 and 21.25, respectively. Nevertheless, when we simply employ eight drones and expand the capacity to three packages, we can achieve a much higher resupply payoff of 22.94. Alternatively, by employing nine drones with a capacity of two packages, the resupply payoff reaches 23.12. This suggests that the most effective approach to improving the resupply payoff is a combination of employing more drones and expanding the drone capacity.

6. Conclusions

We study a drone resupply model with multiple trucks and drones for on-time package delivery in urban areas. In this model, trucks deliver packages at delivery sites, each of which may have multiple individual deliveries for customers in one residential complex or office building. Drones are employed to resupply late-available packages from the depot to trucks en route, and then trucks make the final deliveries to customers. During this delivery process, trucks adhere to given routes with flexible departure times from the depot and delivery sites, subject to delivery deadlines. Such a drone resupply model is applicable to two scenarios of on-time delivery, namely, inter-city overnight shipping and guaranteed time-definite delivery for online shopping.

We develop a time-expanded network flow model with side constraints, which can generate the optimal resupply schedule within a reasonable computation time, and design a greedy algorithm to quickly obtain effective solutions. The computational results demonstrate that these solution approaches can serve as effective decision-making tools to generate drone resupply schedules.

Additionally, managerial insights derived from the computational study can assist logistics companies in making informed decisions.

- At an operational level, the efficiency of drone resupply is significantly affected by the allowed delay time in the route of each truck. When truck routing is designed to have a longer maximum allowed delay time, the drone resupply is more efficient. This not only provides guidance for designing truck routing, but also justifies the practical applicability of our proposed model. It is possible for drone resupply to achieve high efficiency by respecting the current truck-based delivery model, thus leading to easy adoption of drone resupply.
- At a strategic level, our work can help the logistics company determine the fleet team configuration. When the volume of UFD packages is high, the delivery company can increase both the number of drones and drone capacity. However, both strategies have decreasing marginal efficiency. This finding is aligned with the motivation of our drone resupply model, that is, suitable for scenarios with a small volume of UFD packages. For high UFD volumes, alternative operational models are necessary.

This study conducts a preliminary exploration into on-time delivery assisted by drone resupply in urban areas, which may inspire some interesting future research. For example, the proposed model can be integrated with online shopping platforms, allowing sellers to provide a real-time response regarding whether a dynamically arriving order can be delivered on time or not. Furthermore, the proposed drone resupply model could be adopted into meal delivery, where orders arrive dynamically throughout the day and the delivery deadline is tighter. Drones could be used to transport meals from restaurants to a group of riders to reduce the number of delayed deliveries of meals.

Acknowledgments

The authors thank the associate editor and the three anonymous referees for their valuable comments. This work is partially supported by the National Natural Science Foundation of China[Grants 72022018 and 72091210], and Hong Kong RGC[GRF 16210020].

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ejor.2024.05.025.

References

Christofides, N., & Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society*, *20*, 309–318.

Christofides, N., Mingozzi, A., & Toth, P. (1979). The vehicle routing problem. *Combinatorial Optimization*, 315–338.

Dayarian, I., Savelsbergh, M., & Clarke, J.-P. (2020). Same-day delivery with drone resupply. *Transportation Science*, *54*(1), 229–249.

Dienstknecht, M., Boysen, N., & Briskorn, D. (2022). The traveling salesman problem with drone resupply. *OR Spectrum*, *44*(4), 1045–1086.

Dukkanci, O., Koberstein, A., & Kara, B. Y. (2023). Drones for relief logistics under uncertainty after an earthquake. *European Journal of Operational Research*, *310*(1), 117–132.

Erdoğan, G. (2017). An open source spreadsheet solver for vehicle routing problems. *Computers & Operations Research*, *84*, 62–72.

Erdoğan, G., & Yıldırım, E. A. (2021). Exact and heuristic algorithms for the carrier-vehicle traveling salesman problem. *Transportation Science*, *55*(1), 101–121.

Ham, A. M. (2018). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C (Emerging Technologies)*, *91*, 1–14.

Hern, A. (2016). Amazon claims first successful Prime Air drone delivery. https://www.theguardian.com/technology/2016/dec/14/amazon-claims-first-successful-prime-air-drone-delivery.

Karak, A., & Abdelghany, K. (2019). The hybrid vehicle-drone routing problem for pick-up and delivery services. *Transportation Research Part C (Emerging Technologies)*, *102*, 427–449.

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.

Kruk, B. (2020). During COVID-19, drone delivery is really taking off. https://www.walgreensbootsalliance.com/news-media/our-stories/during-covid-19-drone-delivery-is-really-taking-off.

Madani, B., & Ndiaye, M. (2022). Hybrid truck-drone delivery systems: A systematic literature review. *IEEE Access*, *10*, 92854–92878.

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.

Nguyen, M. A., & Hà, M. H. (2023). The parallel drone scheduling traveling salesman problem with collective drones. *Transportation Science*, *57*(4), 866–888.

Norman, H. (2019). Drone delivery launched by FedEx and Wing in first-of-its-kind trial in US. https://www.parcelandpostaltechnologyinternational.com/news/automation/drone-delivery-launched-by-fedex-and-wing-in-first-of-its-kind-trial-in-us.html.

Pina-Pardo, J. C., Silva, D. F., & Smith, A. E. (2021). The traveling salesman problem with release dates and drone resupply. *Computers & Operations Research*, *129*, Article 105170.

Poikonen, S., & Golden, B. (2020). The mothership and drone routing problem. *INFORMS Journal on Computing*, *32*(2), 249–262.

Saleu, R. G. M., Deroussi, L., Feillet, D., Grangeon, N., & Quilliot, A. (2022). The parallel drone scheduling problem with multiple drones and vehicles. *European Journal of Operational Research*, *300*(2), 571–589.

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.

Tiniç, G. O., Karasan, O. E., Kara, B. Y., Campbell, J. F., & Ozel, A. (2023). Exact solution approaches for the minimum total cost traveling salesman problem with multiple drones. *Transportation Research, Part B (Methodological)*, *168*, 81–123.

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.