Computers & Operations Research 185 (2026) 107295

Contents lists available at ScienceDirect

Computers and Operations Research

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

On-demand meal delivery: Drone scheduling with battery replacement optimization

Wenqian Liu [a], Yandong He [b,*], Ginger Y. Ke [c,*], Lianmin Zhang [d]

[a] School of Economics and Management, Southwest Jiaotong University, Chengdu, China
[b] School of Management, Shenzhen Polytechnic University, Shenzhen, China
[c] Faculty of Business Administration, Memorial University of Newfoundland, St. John's, Newfoundland and Labrador, A1B 3X5, Canada
[d] Shenzhen International Center for Industrial and Applied Mathematics, Shenzhen Research Institute of Big Data, Shenzhen, China

ARTICLE INFO

Keywords:
Online meal delivery
Drone delivery
Battery replacement
Time-expanded network flow

ABSTRACT

Drones have become a promising solution for on-demand delivery thanks to their ability to travel fast and navigate without road restrictions. In the context of direct meal delivery using drones, it is a common practice to replace the drone's battery after each round trip between the central launch site and customer locations to prevent power interruptions. This practice can lead to frequent battery replacements, resulting in increased downtime and decreased drone utilization. To enhance the efficiency of drone delivery, we take into account load-dependent energy consumption for the drones and optimize battery replacement along with drone scheduling. A mixed integer programming formulation is constructed to mathematically capture the problem. Additionally, we develop a time-expanded network flow method and a tailored hybrid variable neighborhood search algorithm to solve the problem exactly and heuristically. Computational studies validate the effectiveness and efficiency of the proposed operational model and solution approaches. The results indicate that optimizing battery replacement can induce an increase in on-time deliveries by up to 7.69% compared to replacing batteries after each return, and by 14.29% compared to only replacing them when energy levels are low. Such benefits are particularly significant in scenarios with tighter delivery deadlines and longer battery replacement times.

1. Introduction

With advances in technology and the proliferation of online meal ordering platforms, such as Grubhub, Uber Eats, and Meituan, online meal ordering services are more accessible and affordable than ever. This service allows customers to order food from restaurants with just a few clicks and have it delivered to their doorstep; such convenience has led to a significant increase in demand for online meal ordering, with the market size exceeding 100 billion dollars (Mao et al., 2022).

Customers expect prompt deliveries, making delivery speed a crucial factor in retaining their loyalty. Therefore, platforms strive to deliver meals as quickly as possible. For example, the average delivery times for Meituan and Ele.me, two popular meal ordering platforms in China, are 38 and 41 min, respectively (Mao et al., 2022). To achieve high service-quality targets, platforms often rely on a large fleet of couriers to fulfill orders. However, due to rising labor costs and traffic congestion caused by couriers using motorbikes, platforms are actively searching for alternative solutions to ensure timely meal delivery.

Drones have emerged as a compelling option thanks to their ability to travel fast and navigate freely without road restrictions. Drone delivery is already operational in real-world applications. For instance, Wing has launched a drone service in Australia for delivering food, coffee, and medicines (Brown, 2019). Foodpanda has initiated trials on drone-based food delivery in Singapore (Foodpanda.com, 2020). Zipline continues to expand drone food delivery in U.S. Heier (2023). Meituan has even built drone airports on the rooftops of certain shopping malls in Shenzhen to facilitate the direct/point-to-point delivery of meals to vending-machine-like kiosks, from which customers can conveniently retrieve their food. In this setup, drones run back-and-forth trips between the airport and customer locations to fulfill the delivery of meal orders. This innovative system proves to be remarkably faster than traditional methods, with an average delivery time of approximately 12 min (Xu, 2023). These pioneer projects have paved the way for adopting drones in on-time delivery services.

However, concerns have been raised regarding the potential risks associated with power disruptions in drones. This underscores the importance of considering energy consumption to ensure a reliable and safe drone-based meal delivery system. During the early stage of implementing drone delivery, energy consumption is typically addressed

* Corresponding authors.
E-mail addresses: wenqian_liu@swjtu.edu.cn (W. Liu), heyandong@szpu.edu.cn (Y. He), gke@mun.ca (G.Y. Ke), lmzhang@sribd.cn (L. Zhang).

https://doi.org/10.1016/j.cor.2025.107295
Received 6 September 2024; Received in revised form 28 August 2025; Accepted 30 September 2025
Available online 4 October 2025

by imposing a predetermined limit on the drone's flight endurance or range. In recent years, an increasing number of studies (e.g., Cheng et al., 2020; Ghelichi et al., 2021; Liu, 2023) examined the impact of drone design and delivery operations on energy consumption, such as drone weight, battery weight, and payload weight. Those works, despite the efforts, entirely overlooked the impact of battery replacement on the efficiency of the delivery system.

There are two common practices for battery replacement during drone delivery operations: one involves replacing the drone's battery upon its return to the depot, as currently implemented by Meituan and other companies, while the other replaces the battery when the energy level is low and insufficient for the next delivery trip. These two approaches may remain consistently efficient when drones manage multiple orders within a single trip. In such cases, the drone's battery energy is often significantly depleted during the long-distance journey, thereby necessitating a replacement upon its return. However, these strategies may not be as effective for direct/point-to-point delivery. While the time required for battery replacement is generally minimal compared to the overall service period, frequent replacements can increase downtime and, as a result, reduce the number of orders fulfilled within a given timeframe. Hence, it is imperative to explore innovative solutions for battery replacement aimed at optimizing the performance of delivery drones, specifically in the context of direct on-time meal delivery.

Motivated by the aforementioned challenges, we study a drone scheduling problem with battery replacement optimization. Our research specifically focuses on addressing the practical application of drones in ensuring the timely delivery of meals, where drones implement direct round trips between a central launch site and customer locations. Given the time-sensitive nature of meal orders, the goal is to maximize the number of meal orders successfully delivered by the promised delivery deadlines. We apply a load-dependent energy consumption model to determine whether the drone's battery should be replaced upon its return to the launch site. To the best of our knowledge, this research represents the first effort to integrate battery replacement optimization into the framework of drone delivery scheduling.

In summary, the contributions of this paper are as follows.

- We study a drone scheduling problem in the context of direct timely meal delivery. To promote the energy efficiency of the drone-based delivery system, a load-dependent energy consumption model is developed to factor in the drone's weight, battery, and payload. Furthermore, we incorporate decisions related to drone energy refueling into the drone scheduling problem, determining the optimal timing for battery replacement to improve the on-time delivery performance.
- We formally define the problem and formulate it as a mixed integer programming model that mathematically captures the objective and key constraints. Two solution methods are presented to solve the problem exactly and heuristically, respectively. We remodel the problem using a time-expanded flow network formulation with additional side constraints. We also design a hybrid variable neighborhood search algorithm, where operators and several rules are proposed by capitalizing on problem-specific characteristics, to quickly attain high-quality solutions.
- We conduct a series of computational studies to validate our model and solution approaches. From an algorithmic perspective, our solution approaches are effective and efficient. Notably, the time-expanded network flow method demonstrates high efficiency in solving practical instances. From a managerial perspective, optimizing battery replacement is advantageous over other practical alternatives, especially in situations where the promised delivery window is tight or when battery replacement takes a long time. Delivery companies using batteries with large capacities can implement a strategy of replacing the battery when it reaches a low energy level. While this operation may slightly reduce the efficiency of drone delivery, it simplifies decision-making in practice.

The remainder of this study is as follows. Section 2 gives a formal description of the problem, followed by a comprehensive review of the related literature in Section 3. Section 4 presents the solution methods, including a mixed integer programming formulation. In Section 5, we analyze the results of an extensive computational study. Finally, Section 6 concludes this paper with remarks.

2. Problem description

We consider a Drone-based On-demand Meal Delivery Problem (DOMDP) for a logistics company that employs drones to fulfill timely delivery of meals for customers that are covered within the designated service range of a specified launch site.

Meal orders are prepared at restaurants and carried to the launch site by the restaurant's staff. The set of meal orders available at the launch site is denoted by $\mathcal{M} = \{1, 2, \dots, M\}$. Each meal order p is characterized by (w_p, d_p, e_p, l_p), where w_p represents the weight of the meal, d_p indicates the delivery destination, e_p denotes the time when the meal is available for load at the launch site, and l_p is the promised delivery deadline of order p.

A fleet of drones, denoted as $\mathcal{K} = \{1, 2, \dots, K\}$, is deployed to transport orders from a central launch site to customer locations. Each drone can perform multiple round trips, carrying a single meal for a customer on each trip. This limitation is due to current drone technology and regulations, which make it easier for drones to navigate in urban areas. The availability status of drone k is represented by (θ_k, δ_k), indicating the time and battery energy when the drone is ready at the launch site. The time taken for a drone to travel one way between the launch site and the delivery destination of meal order p is denoted as t_p. The time required for loading a meal onto the drone at the launch site or unloading the meal at the delivery destination is denoted by s.

Considering the limited flight endurance of drones, we introduce Q to represent the energy level of fully charged batteries equipped on a drone. Whenever drone k returns to the launch site, we check the remaining battery energy and decide whether to replace its batteries with fully charged ones. If its battery is replaced, the duration required for battery replacement is τ, and the battery energy is updated to Q. According to Dorling et al. (2016), the power in Watts consumed by a n-rotor drone is

$$(W+m)^{\frac{3}{2}}\sqrt{\frac{g^3}{2\rho\zeta n}},$$

where W is the weight of the drone frame, m is the total weight of the equipped battery and payload, g is the gravity, ρ is the fluid density of air, and ζ is the area of the spinning blade disc. Based on this power consumption model, we can derive the energy consumed in the round-way trip for delivering meal order p, denoted by E_p. Specifically,

$$E_p = (W+m+w_p)^{\frac{3}{2}}\sqrt{\frac{g^3}{2\rho\zeta n}}t_p + (W+m)^{\frac{3}{2}}\sqrt{\frac{g^3}{2\rho\zeta n}}t_p, \tag{1}$$

where the weights of the battery and payload (the meal carried) are indicated separately by m and w_p to calculate the energy consumption during outbound and inbound legs. In Eq. (1), the first term represents the energy consumed during the outbound leg for delivering meal order p, and the second gives the energy consumption on the return leg.

To maximize the number of orders delivered within the promised delivery deadlines, we optimize the drone scheduling with battery replacement, which encompasses the following key decisions:

- Whether meal order p is accepted to be fulfilled within the promised delivery deadline;
- If order p is promised to be delivered on time, it is assigned to which drone for delivery;
- The sequence in which the assigned meal orders are delivered;
- Whether to replace the drone batteries with fully charged ones when a drone returns.

Ultimately, we can characterize the delivery schedule for drone k as

$$\phi_k = \{(y_1, p_1), (y_2, p_2), \ldots, (y_r, p_r), \ldots\},$$

where (y_r, p_r) denotes the decision related to the rth delivery trip. y_r indicates whether the drone battery is replaced ($y_r = 1$) or not ($y_r = 0$) before starting the rth delivery trip, and p_r represents the order delivered on the trip.

3. Literature review

This section provides a comprehensive review of the literature closely related to our study. We begin by examining the research on meal delivery routing problems, and then delve into the studies that investigate drone scheduling problems.

3.1. Vehicle routing for meal delivery

Recently, online meal delivery problems have attracted significant attention and have emerged as a growing research topic. Some studies developed effective order assignment schemes without explicitly considering rider routing. Notable examples include Liu et al. (2021), Ulmer et al. (2021), Jahanshahi et al. (2022), and Agnetis et al. (2023). Others focused on the vehicle routing problems, such as Wang (2018), Reyes et al. (2018), Yildiz and Savelsbergh (2019), Steever et al. (2019), Tu et al. (2019), Behrendt et al. (2023), and Simoni and Winkenbach (2023).

Specifically, Wang (2018) considered three logistics services for meal delivery, namely, exclusive, sharing, and sharing+, where riders are shared across multiple restaurants in different ways. Reyes et al. (2018) aimed to optimize various performance metrics, including the total number of orders delivered and the average click-to-door time, by studying the courier assignment and shift scheduling problem. Yildiz and Savelsbergh (2019) proposed a simultaneous column and row generation method to solve a similar problem that was examined by Reyes et al. (2018). Steever et al. (2019) examined dynamic courier routing that allows customers to order food from multiple restaurants, with the goal of maximizing the total earliness of all deliveries to customers. Simoni and Winkenbach (2023) and Behrendt et al. (2023) delved into the realm of crowd-sourced on-demand meal delivery services, each exploring distinct features within this domain. Simoni and Winkenbach (2023) enabled riders to gather orders from various restaurants and deliver them individually. On the other hand, Behrendt et al. (2023) looked into the potential benefits of utilizing scheduled and crowd-sourced riders concurrently to manage fluctuating demands efficiently, seeking to reduce the overall expenses associated with courier payments and expired orders.

All the research mentioned above studied the delivery routing for riders. As far as we know, only two studies investigated the scheduling of drones. Liu (2019) examined the online fleet dispatch operations for on-demand meal delivery using drones, where the energy consumption depends on the size of the payload, namely, the number of orders carried, and drones are refueled at charging depots when the battery charge drops below a certain level. Huang et al. (2021) considered a dynamic task scheduling problem in drone-based meal delivery to minimize total tardiness. They assumed that energy consumption varies linearly with the payload weight and that drones with low energy levels are recharged at the dispatch center.

3.2. Drone scheduling problems

In our study, drones are scheduled to fulfill on-time deliveries without the involvement of trucks. Therefore, we herein focus on the studies that examine drone scheduling problems, where a fleet of drones is dispatched to implement a series of trips. For a literature review on the applications of drones across various fields, such as infrastructure, agriculture, surveillance, and transportation/logistics, the readers are referred to Chung et al. (2020), or to Macrina et al. (2020) and Pasha et al. (2022) for a comprehensive survey of drone routing and scheduling problems in last-mile delivery. For studies focusing on various truck-and-drone collaborative delivery systems, the readers are referred to Murray and Chu (2015), Dayarian et al. (2020), Tamke and Buscher (2021), Saleu et al. (2022), Dukkanci et al. (2023), and Mahmoudinazlou and Kwon (2024), among others.

Drone scheduling has been widely investigated in the last-mile package delivery service, where drones are employed to deliver packages to customers within specified time windows or delivery deadlines. Dorling et al. (2016) studied two drone routing problems to minimize the total cost within a delivery time limit and the delivery span subject to budget constraints, respectively. To minimize the overall delivery cost, Cheng et al. (2020) worked on a drone routing problem, where batteries are swapped when a new delivery trip starts. Dukkanci et al. (2021) examined a range-limited drone delivery problem with speed optimization to minimize the overall operational cost and energy consumption of drones. Wen and Wu (2022) worked on a delivery problem using heterogeneous drones, in which a large drone carries and launches small drones to deliver parcels. Liu (2023) proposed a drone routing model for on-demand package delivery to maximize the delivery profit. Li et al. (2024) inspected a drone routing problem with swarm synchronization for delivering heavy items. Cheng et al. (2024) explored the impact of uncertain wind conditions on the delivery times of drones. Zhu et al. (2024) considered a locker-based drone delivery problem, where drones collect and deliver goods at the lockers, and the lockers serve as parking platforms for battery replacement. Cicek et al. (2024) specially addressed the influence of communication constraints on the performance of a drone-based package delivery system.

Additionally, drones can be employed to implement monitoring tasks and deliver healthcare items. Xia et al. (2019) analyzed a drone scheduling problem for monitoring the emissions of vessels, where the sequence and times for inspecting vessels are determined to maximize the total weighted number of inspected vessels. Ghelichi et al. (2021) considered a drone-based medicine delivery problem, where charging stations are determined along with the drone scheduling. During a trip, a drone can be recharged at one or more charging stations.

3.3. Research gaps

First, in the realm of meal delivery, our research stands as one of the pioneering studies exploring the routing and scheduling of drones. Existing literature primarily focuses on studying the routing problem for riders, and the derived solution techniques are no longer applicable when the delivery is conducted by drones. In comparison to conventional delivery vehicles powered by gasoline or electricity, energy disruptions in drones can potentially cause severe damage to the public, especially in heavily populated urban areas. This emphasizes the importance of battery management for a reliable and safe drone delivery system. Therefore, additional realistic factors, such as energy consumption and recharging schemes, should be incorporated into the scheduling of drones, making the scheduling problem even more computationally challenging.

Second, in the literature regarding drone scheduling problems, our study is the first to incorporate battery replacement decisions. Instead of replacing the battery when it is in low energy or whenever the drone returns, our study optimizes the battery replacement decisions along with drone scheduling so as to fulfill as many orders within the promised delivery deadlines as possible. In addition to battery replacement decisions, our study differs from the existing drone scheduling problems in other aspects, such as the objectives for adopting drones and the features of drone operations, as outlined in Table 1. These differences and the unique decisions on battery replacement raise a new drone scheduling problem, which has not been explored in the existing literature. We develop solution techniques to address this problem.

Table 1
Summary of the Literature on Drone Scheduling Problems.

References	Objectives	Tasks	Drone operations			Algorithms
			Energy consumption	Battery replacement	Capacity	
Dorling et al. (2016)	Minimize cost or delivery time	Package delivery	Payload weight	Every return	Multiple	Simulated annealing
Liu (2019)	Minimize total delivery lateness	Meal delivery	Payload size	Low energy	Multiple	MIP formulation & rolling-horizon
Xia et al. (2019)	Maximize weights of inspected vessels	Monitoring	Fixed endurance	Low energy	Multiple	Network flow & Lagrangian relaxation
Cheng et al. (2020)	Minimize delivery cost	Package delivery	Payload weight	Every return	Multiple	Branch-and-cut
Huang et al. (2021)	Minimize total delivery tardiness	Meal delivery	Payload weight	Low energy	Multiple	K-Means++ & simulated annealing
Dukkanci et al. (2021)	Minimize cost and energy	Package delivery	Payload weight, and drone speed	Every return	Single	Second order core programming
Ghelichi et al. (2021)	Minimize the delivery time	Medicine delivery	Payload weight	Charging stations	Single	Mathematical formulation & reduction method
Wen and Wu (2022)	Minimize delivery time and cost	Package delivery	Payload weight	Every return	Multiple	Three-stage method & iterative optimization
Liu (2023)	Maximize profit	Package delivery	Payload weight & wind conditions	Every return	Multiple	MIP formulation & optimization-simulation
Li et al. (2024)	Minimize fixed and variable costs	Package delivery	Fixed endurance	Every return	Multiple	Adaptive large neighborhood search
Cheng et al. (2024)	Minimize the delivery delays	Package delivery	Fixed range	Every return	Single	Robust optimization & branch-and-cut
Zhu et al. (2024)	Minimize the travel distance	Package delivery	Sufficiently large endurance	Every landing	Single	Two-stage method & branch-and-cut
Cicek et al. (2024)	Minimize the flight distance	Package delivery	Fixed endurance	Charging stations	Single	MIP formulation & genetic algorithm
This paper	Maximize on-time deliveries	Meal delivery	Payload weight	Optimized with drone scheduling	Single	Network flow & neighborhood search

Finally, while there are similarities between electric vehicle (EV) scheduling (e.g., Zang et al., 2022; Klein and Schiffer, 2023; Lera-Romero et al., 2024; Xiao et al., 2024; Jeong et al., 2024) and drone-based meal delivery scheduling, the unique objective and constraints of drone delivery system necessitate specialized research. EVs can operate for hours or days on a single charge and often rely on offline scheduling (e.g., overnight charging) to minimize total operational costs. In contrast, meal delivery operates within strict time windows, where the goal of recharging scheduling is to maximize the number of on-time deliveries. Furthermore, energy consumption for EVs is typically assumed to be proportional to traveling distance, while drone delivery systems employ a load-dependent energy consumption model, introducing complex constraints. These distinctions create a unique problem, where existing EV-focused methods are not applicable.

4. Solution approaches

In this section, we first address the proposed drone scheduling problem using the mathematical formulation method (Section 4.1). Then, we tailor two additional solution approaches to solve the problem exactly and heuristically, respectively. In particular, we develop an exact algorithm (Section 4.2) which involves modeling the problem as a time-expanded flow network and then reformulating it as a minimum-cost network flow model with side constraints. The optimal solutions can be obtained by solving this formulation. Next, we design a heuristic algorithm based on the neighborhood search technique (Section 4.3), aiming to quickly find high-quality solutions.

4.1. Mixed integer programming formulation

To formulate DOMDP, we first introduce $\mathcal{R} = \{1, 2, \ldots, R\}$ to denote the possible trips that a drone can undertake. Here, R represents the maximum number of delivery trips that a drone can implement within the given service period. To estimate R, we assume that all meals are delivered to the closest customer location and have the latest delivery deadline. Thus, R can be determined as the maximum number of round trips that a drone can complete from the launch site to the nearest customer location within the latest delivery deadline, which is accordingly denoted by

$$R = \left\lceil \frac{\max_{p\in\mathcal{M}}\{l_p\}}{2(\min_{p\in\mathcal{M}}\{t_p\} + s)} \right\rceil.$$

Note that R is an unattainable upper bound of the number of trips a drone can implement because customer locations and delivery deadlines are not likely to be the same in practical instances.

Additionally, we define the following decision variables.

- x^r_{pk}: A binary variable, indicating whether order p is delivered by drone k on the rth trip or not. If $x^r_{pk} = 1$, order p is delivered by drone k on the rth trip.
- y^r_k: A binary variable for the battery replacement decision. $y^r_k = 1$ means that the battery of drone k is replaced before starting the rth trip.
- z^r_k: A non-negative variable, denoting the remaining energy level of drone k before starting the rth delivery trip.
- a^r_k: A non-negative variable, denoting the starting time of the rth trip of drone k.

With the above notation, we establish a mixed integer programming formulation to describe the drone scheduling problem mathematically, i.e., DOMDP.

$$\max \sum_{p\in\mathcal{M}}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}} x^r_{pk} \tag{2}$$

subject to:

$$\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}} x^r_{pk} \le 1 \qquad \forall p \in \mathcal{M}; \tag{3}$$

$$\sum_{p\in\mathcal{M}} x^r_{pk} \le 1 \qquad \forall k \in \mathcal{K},\ r \in \mathcal{R}; \tag{4}$$

$$a^1_k \ge \theta_k + \tau y^1_k + s \qquad \forall k \in \mathcal{K}; \tag{5}$$

$$a^r_k \ge \sum_{p\in\mathcal{M}} e_p x^r_{pk} + s \qquad \forall k \in \mathcal{K},\ r \in \mathcal{R}; \tag{6}$$

$$a_k^{r+1} \geq a_k^r + 2\sum_{p\in\mathcal{M}} t_p x_{pk}^r + s + \tau y_k^{r+1} \quad \forall k \in \mathcal{K},\ r \in \mathcal{R}; \tag{7}$$

$$a_k^r + t_p + (x_{pk}^r - 1)\mathrm{INF} \leq l_p \quad \forall p \in \mathcal{M},\ k \in \mathcal{K},\ r \in \mathcal{R}; \tag{8}$$

$$\sum_{p\in\mathcal{M}} x_{pk}^{r+1} \leq \sum_{p\in\mathcal{M}} x_{pk}^r \quad \forall k \in \mathcal{K},\ r \in \mathcal{R}; \tag{9}$$

$$\sum_{p\in\mathcal{M}} E_p x_{pk}^r \leq z_k^r \quad \forall k \in \mathcal{K},\ r \in \mathcal{R}; \tag{10}$$

$$z_k^1 \leq \delta_k + Q y_k^1 \quad \forall k \in \mathcal{K}; \tag{11}$$

$$z_k^{r+1} \leq z_k^r - \sum_{p\in\mathcal{M}} E_p x_{pk}^r + Q y_k^{r+1} \quad \forall k \in \mathcal{K},\ r \in \mathcal{R}; \tag{12}$$

$$x_{pk}^r \in \{0,1\} \quad \forall p \in \mathcal{M},\ k \in \mathcal{K},\ r \in \mathcal{R}; \tag{13}$$

$$y_k^r \in \{0,1\} \quad \forall k \in \mathcal{K},\ r \in \mathcal{R}; \tag{14}$$

$$0 \leq z_k^r \leq Q \quad \forall k \in \mathcal{K},\ r \in \mathcal{R}; \tag{15}$$

$$a_k^r \geq 0 \quad \forall k \in \mathcal{K},\ r \in \mathcal{R}. \tag{16}$$

Objective function (2) maximizes the number of orders that can be fulfilled within the delivery deadlines. Constraint set (3) ensures that each meal order is delivered at most once. Constraint set (4) is the condition for drone capacity, which specifies that a drone makes a single delivery per trip. Constraints (5)–(7) calculate the starting times of delivery trips. In particular, drone k can start a delivery trip only after it has returned to the launch site following the completion of the previous delivery trip, successfully loaded the assigned meal order, and undergone the necessary battery replacement if required. Constraint set (8) specifies the condition for on-time delivery, where INF denotes a very large number. Specifically, if order p is promised to be fulfilled on time, it has to be delivered before the delivery deadline; otherwise, this order would not be assigned to any drones. Constraint set (9) secures the connectivity of drone delivery trips. Constraint set (10) gives the conditions on energy consumption, ensuring that each delivery trip is conducted without energy disruption. Constraints (11) and (12) update the drone's energy level before starting the next delivery trip. Specifically, if the battery is replaced with a fully charged one, the energy level for the next trip is Q; otherwise, it is the remaining energy after running the previous delivery trip. Constraints (13)–(16) indicate domains of the decision variables.

4.2. Time-expanded network flow method

In the Time-Expanded Network Flow (TENF) method, we first model the proposed problem using a time-expanded flow network, where feasible delivery trips of drones are denoted by nodes, and the scheduling of drones is represented by arc-disjoint flow paths. Subsequently, we establish a formulation based on the classical minimum-cost network flow model with additional constraints that capture the specific challenges present in our scenario. Solving this formulation allows us to obtain the optimal drone schedules.

The TENF approach provides a natural and rigorous framework for modeling the temporal dependencies between drone dispatch trips and the constraints on energy consumption involved in the drone-based meal delivery problem. Specifically, a time-expanded flow network can explicitly capture the evolution of system states over time, enabling precise scheduling of drone operations while accounting for energy levels and battery replacement requirements. This method is well-suited for drone-based meal delivery, where strict time windows (e.g., meal freshness guarantees) and energy limitations (e.g., finite battery capacity) introduce complex inter-dependencies between scheduling and recharging decisions.

Such solution approaches have been successfully applied in fields like transportation (Zheng and Chiu, 2011; Boland et al., 2017; Ocampo-Giraldo et al., 2024) and scheduling (Lim et al., 2012; Xia et al., 2019). By leveraging the method's strengths in temporal modeling, we extend its applicability to a novel domain with unique drone-specific operational constraints.

4.2.1. Time-expanded flow network

For each drone dispatch trip, we first determine the order to be assigned to the drone and decide whether to replace the drone's battery, and then check the feasibility conditions related to delivery deadlines and energy consumption. If the drone can arrive at the delivery destination before the specified deadline and the battery energy is sufficient to cover the round-way trip, then the trip is considered feasible.

Specifically, the delivery schedule is constrained by feasibility requirements, including the promised delivery time for meal orders and the load-dependent energy consumption of drones. Suppose that drone k delivers order p_1 on the current trip, during which drone k arrives at the delivery destination at time $\bar{a}$ and returns to the launch site with remaining battery energy $\bar{z}$. Drone k can deliver order p_2 timely on the next trip, if either of the following conditions holds.

- $\max\{\bar{a} + s + t_{p_1}, e_{p_2}\} + s + t_{p_2} \leq l_{p_2}$ and $\bar{z} \geq E_{p_2}$. When the drone's battery is not replaced, its remaining battery energy must be sufficient to cover the trip and the drone must arrive at the delivery destination before the specified delivery deadline.
- $\max\{\bar{a} + s + t_{p_1} + \tau, e_{p_2}\} + s + t_{p_2} \leq l_{p_2}$ and $Q \geq E_{p_2}$, namely, the delivery deadline remains achievable even after accounting for the time required for battery replacement.

In the case of the first delivery trip of drone k, meal order p can be delivered on time if at least one of the following conditions is met:

- $\max\{\theta_k, e_p\} + s + t_p \leq l_p$ and $\delta_k \geq E_p$;
- $\max\{\theta_k + \tau, e_p\} + s + t_p \leq l_p$ and $Q \geq E_p$.

These conditions ensure that the delivery deadlines for the orders are feasible whether the battery is replaced or not.

To systematically represent the feasible delivery trips for each drone, we use a search tree. The root node of the tree denotes that drone k starts the delivery service from the launch site. Each node π of the tree represents a feasible delivery trip, which is characterized by $(k, y, p, \bar{a}, \bar{z})$, where k is the drone that implements this delivery trip, y indicates whether the drone's battery is replaced or not before starting this trip, p is the order delivered on this trip, $\bar{a}$ is the time when the drone arrives at the delivery destination of order p, and $\bar{z}$ is the energy level upon the drone's return to the launch site after the preceding trip. Every node π may have a set of child nodes π', representing consecutive trips following the delivery trip denoted by π.

To find all feasible delivery trips, we adopt a depth-first search algorithm to traverse the entire search tree. Specifically, for a given trip denoted by node π, we prioritize the search for consecutive delivery trips that do not require battery replacement and evaluate the feasibility of the trip. If the trip represented by node π' is feasible, the subsequent delivery trips following trip π' are explored. This process continues recursively, examining each subsequent node and assessing the corresponding feasibility. If the trip represented by node π' is infeasible, indicating that the drone's battery energy is insufficient, we backtrack to the parent node π to identify the trips following trip π when the drone's battery is replaced at the launch site. In this way, all feasible delivery trips can be identified with battery replacement decisions.

Example 1 (*Feasible Trips Denoted by Nodes*)**.** As shown in Fig. 1(a), node $(1, 0, 2, 16, 0.08)$ denotes a feasible delivery trip implemented by Drone 1, where the drone arrives at the delivery destination of Order 2 at 16 minutes, and the remaining energy level when it returns to the launch site is 0.08. After battery replacement at the launch site, the drone can deliver Order 3 on the next trip, as indicated by the child node $(1, 1, 3, 35, 0.15)$ of node $(1, 0, 2, 16, 0.08)$.

Based on the search trees representing feasible delivery trips of drones, we construct a time-expanded flow network. We first introduce a source node o with K units of supply and a sink node d with K units of

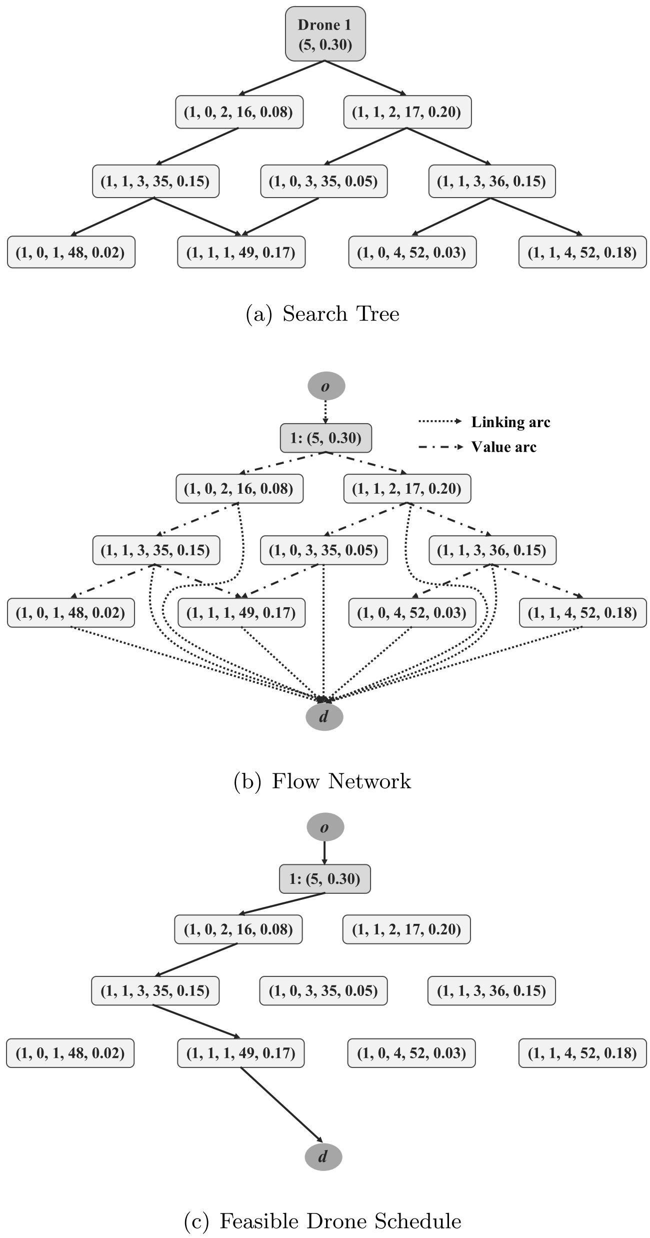

Fig. 1. An illustrative example of the time-expanded flow network.

demand in the flow network. Then, for the root node of the search tree of drone k, we define a node $u : (\theta_k, \delta_k)$ to represent the initial status of drone k at the launch site. Additionally, for each node $\pi : (k, y, p, \bar{a}, \bar{z})$ in the search tree, we define a corresponding node $u : (k, y, p, \bar{a}, \bar{z})$ in the flow network, and the set of nodes in the flow network involving the same order p is denoted by $\bar{\mathcal{V}}_p$.

Further, we introduce two types of arcs in the flow network. *Value Arcs* are used to track and count the number of orders delivered on time. We construct a directed arc from node $u : (\theta_k, \delta_k)$ to the child nodes of the root node in the search tree of drone k. For two nodes $u : (k, y_1, p_1, \bar{a}_1, \bar{z}_1)$ and $v : (k, y_2, p_2, \bar{a}_2, \bar{z}_2)$, if u is the parent node of v in the search tree of drone k, we establish a directed arc (u, v) from node u to node v. *Linking Arcs* are introduced to guarantee the flow connectivity from the source node to the sink node. For each node $u : (\theta_k, \delta_k)$ in the flow network, we construct a linking arc (o, u) from the source node o to node u. For any node u in the flow network, we introduce a linking arc from node u to the sink node d.

Example 2 (*Value and Linking Arcs*)**.** As shown in Fig. 1(b), the arcs from node $(1, 1, 2, 17, 0.20)$ to nodes $(1, 0, 3, 35, 0.05)$ and $(1, 1, 3, 36, 0.15)$ are value arcs constructed to represent that Order 3 can be delivered after Order 2, regardless of whether the drone's battery is replaced or not. The arc from node $(1, 1, 2, 17, 0.20)$ to the sink node d is a linking arc, indicating that the drone remains idle at the launch site after delivering Order 2.

So far, we have constructed a flow network $G = (\mathcal{V}, \mathcal{A})$, which is an acyclic-directed network with nodes $\mathcal{V}$ and arcs $\mathcal{A}$. Let λ_{uv} denote the unit flow cost of arc $(u, v) \in \mathcal{A}$. For an arc $\{(u, v) | v : (k, y, p, \bar{a}, \bar{z})\}$, the unit flow cost is $\lambda_{uv} = -1$. When flows from the source node to the sink node pass through arc (u, v), a cost of "-1" is added to the flow cost. This signifies that the relevant order p is successfully fulfilled within the specified delivery deadline. Since linking arcs are built to ensure the flow connectivity in the flow network, the unit flow cost for all linking arcs is zero. Thus, drone schedules are represented by the arc-disjoint flow paths from source node o to sink node d.

Example 3 (*Drone Schedules Denoted by Flow Paths*)**.** As shown in Fig. 1(c), the flow path from the source node to the sink node indicates that Drone 1 delivers Orders $\{2, 3, 1\}$ sequentially with battery replacements before starting the trips for delivering Orders $\{3, 1\}$.

Some solutions that satisfy the flow conservation constraints may still correspond to infeasible drone schedules due to the repeated delivery of certain orders. To address this issue, we introduce additional side constraints in the formulation.

4.2.2. ILP formulation

For the time-expanded flow network, we bring in a binary variable χ_{uv} to indicate whether arc (u, v) is traversed in the network flow problem. In particular, $\chi_{uv} = 1$ means that arc (u, v) is visited in the network flow solution, which indicates that the order associated with node v is fulfilled on time. Consequently, the decisions made in the network flow problem are equivalent to those in our drone scheduling problem.

The objective of our drone scheduling problem is to maximize the number of meal orders that are delivered on time. In the flow network, each on-time order fulfillment is represented by setting the unit flow cost on certain arcs to "-1". This means the resultant flow cost in the network flow problem corresponds to the negative value of the number of on-time deliveries. Therefore, maximizing on-time deliveries is equivalent to minimizing the flow cost within the network.

With the notation, the drone scheduling problem can be viewed as a minimum-cost network flow problem with side constraints, which has the following ILP formulation.

$$\min \sum_{(u,v)\in\mathcal{A}} \lambda_{uv} \chi_{uv} \tag{17}$$

subject to

$$\sum_{v:(o,v)\in\mathcal{A}} \chi_{ov} = K; \tag{18}$$

$$\sum_{v:(v,d)\in\mathcal{A}} \chi_{vd} = K; \tag{19}$$

$$\sum_{u:(u,v)\in\mathcal{A}} \chi_{uv} = \sum_{u:(v,u)\in\mathcal{A}} \chi_{vu} \qquad \forall v \in \mathcal{V} \setminus \{o, d\}; \tag{20}$$

$$\sum_{u:(u,v)|v\in\bar{\mathcal{V}}_p} \chi_{uv} \le 1 \qquad \forall p \in \mathcal{M}; \tag{21}$$

$$\chi_{uv} \in \{0, 1\} \qquad \forall (u, v) \in \mathcal{A}. \tag{22}$$

Objective (17) is to minimize the total flow cost in the flow network. Constraints (18)–(20) are developed to maintain the connectivity of the drone schedules. Specifically, Constraints (18) and (19) ensure that all drones begin and end their delivery service at the launch site, respectively. Constraint set (20) presents the flow conservation conditions for the nodes in the flow network, excluding the source and sink nodes. Constraints (21) are a set of side constraints used to address the issue of repetitive delivery of meal orders. For each order p, at most one of the nodes $v : (k, y, p, \bar{a}, \bar{z}) \in \tilde{\mathcal{V}}_p$ is visited, ensuring that each order can be fulfilled at most once. Constraint set (22) gives the domains of decision variables.

4.3. Hybrid variable neighborhood search algorithm

In this section, we design a Hybrid Variable Neighborhood Search (HVNS) algorithm to efficiently solve the problem. The algorithm is built upon the general neighborhood search framework proposed by Mladenović and Hansen (1997), which involves generating an initial feasible solution and iteratively updating it via neighborhood search. We specifically craft neighborhood search operators to address the unique characteristics and constraints of our problem. This customization ensures that the neighborhood operators are well-suited to the problem, enhancing the algorithm's ability to explore and exploit the solution space effectively.

4.3.1. Initial solution construction

Recalling that a drone schedule consists of a series of delivery trips along with the corresponding decisions on battery replacement, which is denoted as

$$\phi_k = \{(y_1, p_1), (y_2, p_2), \dots, (y_r, p_r), \dots\},$$

where y_r indicates whether the drone undergoes battery replacement before starting the rth delivery trip, and p_r represents the order delivered on the trip.

To construct an initial feasible solution, we constantly assign meal orders to drones for delivery until all orders are examined. Specifically,

1. Initialize an empty set of drone schedules: $\Phi = \{\phi_1 = \{\}, \phi_2 = \{\}, \dots, \phi_K = \{\}\}$.
2. Repeat the following steps until all orders are considered.
    - Randomly select one order from the set of unassigned orders, assign it to a drone, and identify the drone schedule with the shortest delivery completion time based on the current drone schedule.
    - Compare the completion times when the order is assigned to each drone, and select the assignment that can complete the delivery within the shortest time.
    - Update the drone schedule by assigning the order to the corresponding drone.
    - Remove the assigned order from the set of unassigned orders.
3. Once all orders are considered, the drone schedules $\Phi = (\phi_1, \phi_2, \dots, \phi_K)$ form the initial feasible solution.

By prioritizing the order assignment scheme with the shortest delivery completion time, more available delivery time can be retained for subsequent orders. This enhances the likelihood of meeting the delivery deadlines for the unassigned orders when constructing an initial solution. It also allows for greater flexibility during the process of improving the solution, making it easier to explore different assignments and schedule adjustments.

4.3.2. Solution improvement

The initial solution is iteratively refined by neighborhood search, which consists of two key operations: deletion and re-allocation. These operations are responsible for the destruction and reconstruction of solutions, respectively. While conventional neighborhood search techniques, such as the 1–1 Exchange and 1–2 Exchange, can be utilized, their effectiveness is limited in the context of our drone scheduling problem, which focuses on maximizing the number of deliveries completed before their respective deadlines. These methods relying on exchange do not increase the total number of deliveries, regardless of whether they are performed on different trips for a single drone or across multiple drones. Therefore, we tailor the deletion and re-allocation operations by selectively removing certain orders and reallocating additional ones to facilitate more on-time deliveries.

Specifically, the following five deletion operations are developed.

- *Random Order Removal*: Randomly remove β_1 orders that have been assigned to the drones for on-time delivery. Random destruction operators are common in neighborhood search algorithms, playing a crucial role in balancing exploration with exploitation by selectively perturbing the schedule, thereby circumventing local optima.
- *Energy Order Removal*: Remove β_1 orders with the highest energy consumption among the orders that have been assigned to the drones for delivery. This operator directly incorporates energy consumption as a removal criterion, specifically designed for battery-constrained drones. By targeting high-energy orders to free up battery capacity, it may facilitate more on-time deliveries within each battery replacement cycle.
- *Random Schedule Removal*: Randomly clear the delivery schedules of β_2 drones. This approach removes entire drone schedules rather than individual orders, capitalizing on the inter-dependencies of deliveries by the same drone. By eliminating all orders from specific drones, it allows for more effective scheduling of those drones. These drone-level operators, along with order-level operators, facilitate both fine-grained and broad exploration of the solution space.
- *Greedy Schedule Removal*: Clear the delivery schedules of β_2 drones that have the fewest numbers of deliveries. The rationale behind this is that drones with fewer deliveries are often underutilized. By freeing up these underutilized drones, the algorithm gains flexibility to assign larger sets of orders to them during the reallocation phase.
- *Unassigned-Order-Aware Greedy Removal*: Employ a greedy heuristic to maximize the allocation of unassigned orders to an idle drone. Specifically, the resulting number of on-time deliveries is evaluated against existing drone schedules. If the number exceeds that of an existing drone schedule, clear the corresponding delivery schedule.

The solution improvement process begins with an initial solution Φ, and then undergoes any of the above deletion operations. After a deletion operation results in a solution denoted as Φ', a reallocation operation is performed to allocate extra orders to drones. When selecting an order from the group of unassigned orders, there may be multiple allocation options or none at all. If there are multiple viable options, the one with the least additional delivery time is chosen. If no feasible allocation scheme is found, the order cannot be delivered within the promised timeframe and is therefore retained in the set of orders awaiting assignment.

Furthermore, to enhance the computational efficiency of the search algorithm, we propose the following rules to guide the implementation of the allocation operation.

Rule 1 (*Relative Delivery Sequence*)**.** *Consider two meal orders p and q, where $e_p < e_q$ and $t_p < t_q$, if they are assigned to the same drone, then order p should be delivered before order q.*

The implication of Rule 1 is that drone scheduling should prioritize the meal order with an earlier ready time and a closer destination. With this rule, we can directly identify the best delivery sequence when a new order is assigned to a drone.

Algorithm 1: The pseudo-code of HVNS

Input: the set of orders $\mathcal{M}$ and the fleet of drone $\mathcal{K}$

Output: final solution $\Phi^* = (\phi_1, \phi_2, \ldots, \phi_K)$, where $\phi_k = \{(y_1, p_1), (y_2, p_2), \ldots, (y_r, p_r), \ldots\}$

Initialize drones schedules: $\Phi = \{\phi_1 = \{\}, \phi_2 = \{\}, \ldots, \phi_K = \{\}\}$

repeat

    Randomly assign an order to a drone

    Identify the drone schedule with the shortest completion time

    Update the drone schedule and the set of unassigned orders

until all orders are considered.

Initialize the solution $\Phi^* = \Phi$

Initialize the number of iterations without improvement as $\Delta = 0$

repeat

    Select one deletion operation

    repeat

        $\Phi' \leftarrow$ conduct the selected deletion operation on Φ^*

        $\Phi'' \leftarrow$ conduct the re-allocation operation on Φ' with Rules 1 and 2

        if Φ'' allows more on-time deliveries than Φ^* **then**

            $\Phi^* \leftarrow \Phi''$ and $\Delta \leftarrow 0$

        else

            $\Delta \leftarrow \Delta + 1$

        end if

    until $\Delta \geq N_t$, reaching the maximal number of iterations without improvement

until all deletions operations are implemented

Rule 2 (*Unnecessary Battery Replacement*)**.** *Consider a drone schedule represented by $\phi_k = \{(y_1, p_1), \ldots, (y_r, p_r), \ldots, (y^*, p^*), \ldots\}$, where p^* is the newly-assigned order. For the delivery trip of order p^*, $y^* = 0$ if the drone returns to the launch site after time $e_{p^*} - \tau$ and its battery energy is sufficient for delivering order p^*.*

Rule 2 indicates that if certain conditions are satisfied, we can avoid examining the delivery trip requiring a battery replacement after the current trip, and thus eliminates redundant attempts during reallocation operation.

By consecutively implementing a deletion operation and the re-allocation operation on the current solution, a new solution Φ'' can be obtained. If the new solution has more on-time deliveries than the current one, it is accepted, and further neighborhood searches based on Φ'' are conducted. This neighborhood search continues until the maximum number of iterations without finding better solutions, denoted as N_t, is reached. After obtaining drone schedules from all neighborhood search operations, the one with the maximum number of deliveries is selected. The pseudo-code of HVNS is presented in Algorithm 1.

5. Computational experiments

This section presents a thorough computational analysis to verify the efficiency and effectiveness of our proposed model and solution methods. We first evaluate the computational performance of our solution methods across various scaled scenarios. Then, the benefits of incorporating battery replacement optimization are examined, especially under scenarios with varying promised ready-to-door times, battery replacement durations, and battery capacities.

5.1. Experiment design

To align with the practical drone-based meal delivery service offered within a maximum delivery distance of 10 km (Wang, 2023), we limit our analysis to meal delivery within the service area of a particular launch site. We adopt the locations of customers from the delivery network *X-n207-k16* provided by Uchoa et al. (2017), and set the launch site at the center of this region.

The flying speed of drones is 50 km per hour, and the flight time for drones is calculated based on the Euclidean distance. Drones are randomly available at the launch site during [0,10] minutes. The default energy capacity of fully charged batteries is $Q = 0.3$ kWh, and the default time required for battery replacement is $\tau = 1$ minute. Loading meals onto drones at the launch site or unloading the carried meal at the delivery destination takes $s = 1$ minute. By referring to Cheng et al. (2020), we set the parameters of the energy consumption model for a 6-rotor drone as $W = 2.0$ kg, $m = 1.0$ kg, $g = 9.80$ N/kg, $\rho = 1.204$ kg/m^2, and $\zeta = 0.0064$ m^2.

Meal orders are generated randomly. The delivery destinations are randomly selected from the set of customer locations in the specified network. Meal orders are promised to be delivered within a default ready-to-door time of 20 min. The delivery weight of meal orders is randomly generated within the range from 0.5 to 2.0 kg, which covers the weights of most meal orders and is also consistent with the maximum payload of small drones. We evaluate the scalability of the proposed solution approaches by considering different order volumes, specifically, $M \in \{20, 40, 60, 80, 100, 120, 140\}$. Accordingly, we consider different numbers of drones, namely, $K \in \{10, 20, 30\}$, to cope with different order volumes. In the default design, we consider the case with 20 drones employed to deliver 80 orders. For each group of tests, we measure the average result based on 10 instances, each featuring a variety of meal orders that differ in ready times, delivery destinations, and weights.

The series of tests are conducted on a computer with a 2.00 GHz Intel-Core i9-13900 processor and 64.0 GB of RAM. The ILP formulation is solved using Gurobi Optimizer 11.0.1, and all algorithms are implemented in Python Release 3.11. The parameters involved in the HVNS algorithm are tailored for different sizes of instances. For instances with up to 20 meal orders, $\beta_1 = 3$, $\beta_2 = 2$, and $N_t = 50$; while for instances with at least 30 orders, $\beta_1 = 8$, $\beta_2 = 2$, and $N_t = 200$.

Table 2
Computational efficiency of the time-expanded network flow method.

K	M	DOMDP				TENF					
		Objval avg.	VarN. avg.	ConstrN. avg.	CPU_t (s) avg.	Objval avg.	VarN. avg.	ConstrN. avg.	P1_t (s) avg.	P2_t (s) avg.	CPU_t (s) avg.
10	**20**	18.10	920	3,634	2.41	18.10	2,344	3,538	0.02	0.14	0.17
	40	23.20	1,720	12,890	12.25	23.20	7,542	11,355	0.15	0.49	0.64
	60	27.00	3,150	36,260	152.37	27.20	27,663	41,556	0.75	1.81	2.58
20	**60**	45.50	6,300	72,460	1801.20	46.30	58,739	88,171	1.60	4.02	5.74
	80	49.60	8,300	126,248	1801.87	51.00	143,434	215,233	5.14	13.88	19.55
	100	51.80	12,360	243,140	1803.43	54.40	305,065	457,699	14.51	34.70	50.34
30	**100**	67.10	18,540	364,660	1804.88	73.50	439,297	659,047	20.53	55.70	77.77
	120	67.70	29,520	737,871	1809.24	78.10	1,225,344	1,838,138	70.03	330.02	403.51
	140	72.40	34,320	996,581	1811.99	81.80	1,945,992	2,919,129	130.23	586.04	721.39

5.2. Efficiency and effectiveness of solution approaches

In this part, we evaluate the computational performance of the proposed solution approaches. The drone scheduling problem can be solved exactly using both the DOMDP model and the TENF method. Our focus is on assessing the computational efficiency of the TENF method in terms of the computation time. As the HVNS algorithm demonstrates a fast problem-solving capability (solving all instances within several seconds), we focus on examining its effectiveness, instead of the computational efficiency.

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

First, we measure the computation time of TENF, which is the overall time required for constructing the flow network and solving the ILP formulation. To compare the complexity of the DOMDP formulation and the formulation based on TENF, we present the numbers of variables and constraints, denoted as *VarN* and *ConstrN*, respectively. For the TENF method, we also measure the computation times required for constructing the flow network and solving the ILP formation, denoted by *P1_t* and *P2_t*, respectively. Table 2 showcases the computational performance of DOMDP and TENF across various scaled instances.

Referring to Table 2, we observe that the computation time in both methods is directly influenced by the number of drones and the size of the order volume. This relationship is indicated by the number of variables and constraints involved in each instance. However, when comparing the two methods, it is evident that TENF demonstrates superior computational efficiency.

In particular, for the instances involving 10 and 20 drones, the TENF method can achieve the optimal solutions within one minute. Even for the largest instances with a fleet of 30 drones, the average computation time is approximately 12 min. In contrast, the DOMDP formulation, while capable of solving small instances with a fleet of 10 drones in three minutes, experiences a significant increase in average computation time for large instances. For cases with 60 orders, as the number of drones increases from 10 to 20, solving the DOMDP formulation becomes considerably time-consuming, with several instances failing to reach optimality within the 1800-second time limit. When additional drones or larger order volumes are considered, nearly all instances cannot be solved within the specified time limit, resulting in apparently inferior drone schedules.

It is also interesting to note that the TENF formulation requires larger numbers of variables and constraints, yet it still demonstrates its ability to effectively solve the drone delivery scheduling problem. For example, with 10 drones and 60 meal orders, TENF involves approximately nine times as many variables and slightly more constraints, but its resulting computation time is only a few seconds, which is significantly shorter than that of DOMDP. As the number of drones increases to 20, the average computation time for TENF remains at just several seconds; however, the DOMDP formulation is unable to achieve optimal solutions within the given time limit of 1800 s, despite involving fewer decision variables and constraints.

5.2.2. Optimality gap of the HVNS algorithm

Next, we assess the computational effectiveness of the HVNS algorithm by comparing the achieved on-time deliveries, namely, the number of orders delivered within the promised deadlines. The effectiveness is quantified by *Optimality Gap*, which is defined as

$$Gap = \frac{\text{Optimal Objval} - \text{HVNS Objval}}{\text{Optimal Objval}} \times 100\%,$$

where the optimal objective value is obtained from the TENF method. Table 3 illustrates the optimality gap of the HVNS algorithm across various scaled instances.

Analyzing Table 3, we note that the HVNS algorithm successfully resolves all instances within one minute, while TENF generally requires more computation time, especially in medium and large instances. More specifically, from Table 2, we observe that for large instances with 30 drones serving 120 and 140 orders, only the construction of the flow network takes approximately 70 and 130 s, respectively, which already exceeds one minute. This observation underscores the value of the heuristic in scenarios where timely solutions are crucial.

The optimality gap remains below 15%, although it increases as the problem scale expands. Notably, as the ratio of the number of orders to the number of drones decreases, the optimality gap tends to fall. For example, when using a fixed fleet of drones, the optimality gap decreases as the number of orders decreases. With 10 drones employed, the average optimality gap significantly decreases from 9.90% to 3.83% as the number of orders decreases from 60 to 20, with each drone anticipated to deliver six and two orders, respectively. Similar results are evident when employing 20 and 30 drones. Also, the optimality gap marginally decreases as more drones are engaged in delivering the same quantity of orders. With 60 orders, the use of 20 drones leads to an average gap of 8.21%, which is lower than the gap observed with only 10 drones. This trend can be attributed to the fact that a lower expected delivery load per drone allows for greater flexibility in scheduling. Consequently, the HVNS algorithm can more easily devise an effective delivery plan. In contrast, when drones are required to adhere to tighter schedules for more substantial workloads, attaining precise coordination among the drone fleet becomes increasingly challenging, resulting in a larger optimality gap.

In short, the HVNS algorithm proves to be practically usable with an acceptable optimality gap of approximately 8% when enough drones are deployed to undertake a reasonable workload, i.e., two or three deliveries within a 20 min span. While there may be certain scenarios in which the algorithm exhibits larger gaps, its ability to quickly generate satisfactory results remains valuable. Therefore, the HVNS algorithm can serve as a complementary tool to the TENF method for solving large-scale instances.

5.3. Benefits of optimizing battery replacement

In this part, we analyze the advantages of incorporating decision-making on battery replacement into drone scheduling over existing practical schemes. Specifically, one scheme is to replace the drone's

Table 3
Computational effectiveness of the HVNS Algorithm.

K	M	TENF		HVNS Algorithm				
		Objval	CPU_t (s)	Objval	Gap			CPU_t (s)
		avg.	avg.	avg.	min.	avg.	max.	avg.
10	**20**	18.10	0.17	17.40	0.00%	3.83%	5.88%	0.51
	40	23.20	0.64	21.40	4.17%	7.76%	8.70%	5.13
	60	27.20	2.58	24.50	7.41%	9.90%	14.29%	10.22
20	**60**	46.30	5.74	42.50	6.52%	8.21%	8.89%	9.98
	80	51.00	19.55	46.20	7.84%	9.40%	11.54%	18.32
	100	54.40	50.34	49.00	7.41%	9.92%	12.73%	30.96
30	**100**	73.50	77.77	67.10	6.85%	8.70%	12.16%	27.30
	120	78.10	403.51	70.10	7.69%	10.24%	13.92%	46.62
	140	81.80	721.39	72.70	7.32%	11.13%	14.63%	61.93

Table 4
Performance of battery replacement schemes for different-scaled instances.

K	M	DOMDP				Every Return					Low Energy Level				
		Objval	ReN.	FinT	UtiR	RoI		ReN.	FinT	UtiR	RoI		ReN.	FinT	UtiR
		avg.	avg.	avg.	avg.	avg.	max.	avg.	avg.	avg.	avg.	max.	avg.	avg.	avg.
10	**20**	18.10	7.40	49.13	83.03%	0%	0%	8.10	49.16	82.30%	4.28%	13.33%	7.10	48.74	80.07%
	40	23.20	9.70	50.69	83.46%	2.71%	4.76%	12.60	49.87	81.71%	2.77%	14.29%	9.00	50.88	82.28%
	60	27.20	9.70	50.71	82.46%	4.62%	7.69%	16.00	49.54	79.97%	1.48%	3.70%	9.10	49.99	82.26%
20	**60**	46.30	18.60	51.72	84.57%	1.76%	2.27%	25.50	51.28	82.66%	1.33%	4.55%	18.40	51.34	83.66%
	80	51.00	19.70	51.52	83.57%	2.20%	4.00%	29.90	51.62	81.42%	1.42%	4.17%	18.60	51.59	82.43%
	100	54.40	19.30	51.35	83.13%	3.24%	4.00%	32.70	51.52	80.28%	4.79%	13.04%	18.30	51.18	79.28%
30	**100**	73.50	28.80	51.91	84.48%	2.09%	2.90%	42.00	51.77	82.10%	5.14%	12.31%	26.80	51.40	79.45%
	120	78.10	28.80	52.01	83.88%	2.50%	2.74%	46.20	51.91	81.37%	6.40%	14.08%	26.70	51.67	78.16%
	140	81.80	29.10	52.01	83.88%	3.16%	3.95%	49.30	51.93	80.49%	6.35%	12.00%	27.00	51.55	78.15%

battery every time it returns to the launch site (named as *Every Return*). Another is to conduct replacement only when the energy level becomes insufficient to support the next delivery trip (named as *Low Energy Level*). However, in our model (DOMDP), whether to replace the battery when a drone returns is optimized to maximize the number of on-time deliveries, rather than following certain prescriptive schemes. In this case, batteries may be replaced, even though the remaining energy level is sufficient for the next trip.

To quantify the benefits of our proposed model compared with the practical operations, we introduce the *Ratio of Improvement (RoI)*, namely,

$$RoI = \frac{\text{DOMDP Objval} - \text{Other Objval}}{\text{Other Objval}} \times 100\%,$$

where *DOMDP Objval* and *Other Objval* refer to the numbers of on-time deliveries achieved in our proposed model and a practical scheme, respectively. Additionally, to analyze the performance of the drone fleet, we measure the frequency of battery replacements (*ReN*), latest time when they finish the delivery schedules (*FinT*), and utilization rate of drones (*UtiR*), which is calculated as the fraction of the flight time over the service duration, from the available time of the drone to the time when it finishes the delivery task and returns to the launch site.

5.3.1. Overall performance

We first examine the benefits of our model, where battery replacement decisions are optimized concurrently with drone scheduling. Table 4 shows the performance metrics when adopting different schemes across various scaled instances.

It can be seen that DOMDP can result in more timely deliveries, even when the time needed for battery replacement is only one minute. In particular, DOMDP notably outperforms the *Every Return* and *Low Energy Level* schemes, leading to improvement ratios of up to 7.69% and 14.29% in the number of on-time deliveries, respectively. This demonstrates the advantages of optimizing battery replacement when adopting drones for on-time delivery.

The benefits of DOMDP are primarily attributed to the reduced frequency of battery replacements and flexible order assignment achieved through optimized battery replacement decisions. As indicated in Table 4, the number of battery replacements under DOMDP is substantially lower than that of the *Every Return* scheme. In the context of on-time meal delivery, where the delivery deadline is pressing, even minor time savings can help fulfill more on-time deliveries. Reducing the number of battery replacements can free up more time for drones to make additional deliveries. Meanwhile, yet involving slightly more battery replacements, the delivery schedule under DOMDP tends to be more flexible than that of the *Low Energy Level* scheme. This is due to the fact that the *Low Energy Level* scheme restricts the set of orders assigned for the next delivery based on the drone's remaining energy level, thereby diminishing the flexibility of order assignments. These findings are consistent with the observation that DOMDP achieves the highest drone utilization rate while facilitating a greater number of on-time deliveries within a comparable timeframe.

5.3.2. Impact of the ready-to-door time

Next, we explore the impact of the ready-to-door time on the benefits of optimizing battery replacement. Table 5 shows the performance of different battery replacement schemes in the scenario with varying ready-to-door (RtoD) times.

As expected, the number of on-time deliveries grows when extending the promised delivery deadline. This is because a longer promised ready-to-door time provides more available time for drones to complete additional delivery trips, thus ensuring the on-time delivery of more orders. With the fleet of drones performing more delivery trips, the frequency of battery replacement, finishing time of delivery schedules, and average drone utilization rate also increase.

The benefit of considering battery replacement decisions is more significant in scenarios with tighter promised delivery deadlines. When the promised ready-to-door time is 25 min, the average improvement ratios compared to the *Every Return* and *Low Energy Level* schemes are 2.90% and 4.30%, respectively. When the promised ready-to-door time is as narrow as 10 min, the average improvement ratios increase to 6.48% and 8.00%, respectively. Furthermore, DOMDP consistently attains the high delivery efficiency of drones, as indicated by the

Table 5
Benefits of optimizing battery replacement under different ready-to-door times.

RtoD	DOMDP				Every Return					Low Energy Level				
	Objval	ReN.	FinT	UtiR	RoI		ReN.	FinT	UtiR	RoI		ReN.	FinT	UtiR
	avg.	avg.	avg.	avg.	avg.	max.	avg.	avg.	avg.	avg.	max.	avg.	avg.	avg.
10	40.00	7.60	38.46	80.89%	6.48%	11.76%	17.60	38.52	77.34%	8.00%	30.00%	8.90	38.24	75.85%
15	47.80	14.60	45.97	82.35%	3.69%	6.38%	26.10	46.07	79.54%	6.52%	17.95%	14.30	45.32	77.95%
20	54.40	19.30	51.35	83.13%	3.24%	4.00%	32.70	51.52	80.28%	4.79%	13.04%	18.30	51.18	79.28%
25	60.50	21.00	56.88	83.99%	2.90%	3.51%	38.80	56.88	80.91%	4.30%	11.32%	20.40	56.36	79.95%

Table 6
Benefits of optimizing battery replacement under different replacement durations.

τ (min)	DOMDP				Every Return					Low Energy Level				
	Objval	ReN.	FinT	UtiR	RoI		ReN.	FinT	UtiR	RoI		ReN.	FinT	UtiR
	avg.	avg.	avg.	avg.	avg.	max.	avg.	avg.	avg.	avg.	max.	avg.	avg.	avg.
1	54.40	19.30	51.35	83.13%	3.24%	4.00%	32.70	51.52	80.28%	4.79%	13.04%	18.30	51.18	79.28%
2	53.00	18.00	51.29	80.73%	7.51%	10.42%	29.30	51.04	75.27%	5.19%	15.91%	17.00	51.35	77.06%
3	51.50	16.10	51.26	78.27%	11.70%	14.89%	26.10	51.34	71.04%	4.88%	13.95%	15.80	50.77	75.14%
4	50.30	15.40	51.24	76.24%	15.91%	20.93%	23.40	50.68	66.82%	5.24%	14.29%	15.30	50.94	73.30%

highest drone utilization rate within a similar completion time. These results indicate the importance of optimizing battery replacement when the delivery deadline is tight.

5.3.3. Impact of battery replacement duration

We also examine the impact of battery replacement duration on the benefits of incorporating battery replacement decisions. The computational results considering different battery replacement durations, i.e., $\{1,2,3,4\}$ minutes, are presented in Table 6.

As the time required for battery replacement increases, the number of on-time deliveries tends to decrease. This result aligns with the finding that the number of on-time deliveries decreases when the promised delivery time becomes tighter. With a fixed ready-to-door time, when battery replacement takes longer, there is less time available for drones to travel between the launch site and customer locations. Consequently, the delivery of certain meal orders would be inevitably delayed, leading to fewer on-time deliveries. As more time is spent on battery replacement within a similar timeframe, the proportion of flight time over the service duration, namely, the drone utilization rate, also decreases.

Additionally, when the time required for battery replacement increases, the benefits of optimizing battery replacement become more pronounced. Specifically, compared to the *Every Return* scheme, as the battery replacement duration increases from one to four minutes, the average improvement ratios rise from 3.24% to 15.91%, with certain instances showing improvement ratios as high as 20.93%. A similar trend is observed when contrasted with the *Low Energy Level* scheme, where the average improvement ratio increases slightly from 4.79% to 5.24% and the maximum ratio reaches up to 14.29%. Moreover, drone utilization remains superior under the DOMDP scheme, particularly when contrasted with the *Every Return* scheme. When battery replacement takes four minutes, the drone utilization rates are approximately 76%, 67%, and 73% under DOMDP, *Every Return*, and *Low Energy Level*, respectively. This significantly higher drone utilization accounts for the advantageous outcomes observed under the DOMDP scheme.

5.3.4. Impact of battery capacity

Lastly, the impact of battery capacity on the performance of drone-based on-time meal delivery is investigated, as well as the benefit of incorporating battery replacement decisions. Table 7 presents the results for scenarios involving various battery capacities.

When employing batteries with a larger energy capacity, DOMDP and *Low Energy Level* can lead to more on-time deliveries. The reason is that a larger battery capacity allows drones to travel longer distances without needing a battery replacement. As a result, the frequency of battery replacements diminishes significantly, with average figures declining from 28.70 and 26.40 to 3.40 and 8.10, thereby saving more time for drone travel. Nevertheless, the performance of the *Every Return* scheme remains unchanged as the battery is replaced every time the drone returns to the launch site, regardless of the remaining energy level. This means that employing a larger battery capacity would not influence the battery replacement schedule, and thus the performance of the drone delivery system under the *Every Return* scheme remains the same.

As the battery capacity expands, DOMDP still exhibits advantages, with the benefit increasing and decreasing when compared to the *Every Return* and *Low Energy Level* schemes, respectively. When the battery capacity is increased to 0.60 kWh, the average improvement ratio in comparison to *Every Return* rises to 5.51%, and DOMDP still outperforms *Low Energy Level*, achieving average and maximum improvement ratios of 2.66% and 9.80%, respectively. This finding suggests that utilizing large-capacity batteries and replacing them upon reaching a low energy level may yield a drone schedule that is comparably effective to that generated with battery replacement optimization.

6. Conclusion

This paper investigates a drone scheduling problem in the context of on-demand meal delivery, where drones are dispatched from a central launch site to deliver as many orders within the promised delivery deadlines as possible. We specifically address the drone recharging issues caused by the limited battery life of drones to improve the efficiency of drone delivery systems. Instead of following common practices for battery replacement, we optimize battery replacement in conjunction with drone scheduling. To solve the problem exactly, we develop a time-expanded network flow method, where the problem is characterized as a flow network and reformulated as a minimum-cost network flow model. We also design a hybrid variable neighborhood search algorithm with tailored search operations to quickly obtain high-quality solutions.

The computational results show that the solution approaches can solve the proposed problem effectively and efficiently. The ability of these approaches to deliver satisfactory solutions demonstrates their efficacy in addressing the challenges associated with drone scheduling for on-time delivery services.

In the following, we summarize the managerial insights derived from our numerical experiments which can be used to facilitate the decision-making process of drone-based on-time meal delivery.

- Integrating battery replacement decisions into the drone scheduling problem is both essential and beneficial. Specifically, optimizing battery replacement can result in increased on-time deliveries.

Table 7
Benefits of optimizing battery replacement under different battery capacities.

Q	DOMDP				Every Return					Low Energy Level				
	Objval	ReN.	FinT	UtiR	RoI		ReN.	FinT	UtiR	RoI		ReN.	FinT	UtiR
	avg.	avg.	avg.	avg.	avg.	max.	avg.	avg.	avg.	avg.	max.	avg.	avg.	avg.
0.20	52.40	28.70	48.43	81.22%	0.97%	2.00%	31.90	48.57	80.04%	10.37%	29.27%	26.40	48.45	74.01%
0.30	54.40	19.30	51.35	83.13%	3.24%	4.00%	32.70	51.52	80.28%	4.79%	13.04%	18.30	51.18	79.28%
0.40	55.10	9.40	51.77	84.13%	4.57%	6.00%	32.70	51.07	80.46%	5.09%	17.02%	13.30	51.43	80.63%
0.50	55.30	3.40	51.86	84.76%	4.95%	7.69%	32.70	51.07	80.46%	2.85%	16.67%	9.90	51.60	82.23%
0.60	55.60	3.40	51.91	84.97%	5.51%	7.69%	32.70	51.07	80.46%	2.66%	9.80%	8.10	51.65	82.83%

This improvement can be attributed to enhanced drone utilization, stemming from a reduced frequency of battery replacements and more adaptable order assignments.

- Furthermore, the superiority of optimizing battery replacement is particularly pronounced in scenarios characterized by tighter delivery deadlines (or ready-to-door times) and extended battery replacement durations. These findings underscore the significance of optimizing battery replacement choices for drone scheduling in both drone-only and truck-and-drone collaborative modes in on-time delivery services.
- Optimizing battery replacement continues to exhibit advantages as the battery capacity expands; however, the benefit in comparison to the scheme of replacing the battery upon reaching a low energy level becomes marginal. This indicates that delivery companies providing on-time delivery services can choose to equip drones with large-capacity batteries and replace them upon reaching a low energy level. While this operation may slightly diminish the efficiency of drone delivery, it simplifies the decision-making process, thereby facilitating practical implementation.
- The proposed solution approaches can also be employed to address the dynamics associated with the continuous and random arrival of orders. Particularly, the time-expanded network flow method, which can obtain optimal solutions within mere seconds for medium-sized instances, allows for easy parameter updates and ensures prompt solution generation. By continually updating information on order arrivals, delivery companies can segment the service period into a series of planning horizons and sequentially determine the delivery plan at the beginning of each horizon.

In our study, the main focus of online meal delivery is to ensure on-time deliveries, and thus drones should travel at their maximum speed. Therefore, our research is centered on optimizing battery replacement under a fixed drone speed. Future research could explore the combined impact of battery replacement decisions and varying drone speeds for different applications. Another potential area for future research is to develop a rolling-horizon decision framework for fulfilling dynamic and random meal orders. This framework could be used to plan drone scheduling over consecutive periods and continually adjust the plan as time progresses. Additionally, coordinating drone scheduling with the planning of couriers who pick up meals from restaurants and deliver them to the launch site could enhance the efficiency of the entire delivery process.

CRediT authorship contribution statement

Wenqian Liu: Writing – original draft, Writing – review & editing Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Yandong He:** Writing – original draft, Validation, Software, Methodology, Investigation. **Ginger Y. Ke:** Writing – review & editing, Validation, Project administration, Funding acquisition, Formal analysis. **Lianmin Zhang:** Writing – review & editing, Validation, Supervision, Project administration, Conceptualization.

Acknowledgments

The authors thank the associate editor and the three anonymous referees for their valuable comments. Yandong He is supported by the Shenzhen Polytechnic University Key University-Level Research Project in Humanities and Social Sciences (Grant No. ZX2025001001) and the Basic and Applied Basic Research Foundation of Guangdong Province (Grant No. 2614050002563). Ginger Y. Ke is supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (RGPIN-2022-03514). Lianmin Zhang is supported by the Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone Project under Grant HZQSWS-KCCYB-2024016, the 1+1+1 CUHK-CUHK(SZ)-GDSTC JOINT COLLABORATION FUND with project "Empty Carriage Dispatching in Large-Scale Railway Logistics Networks".

Data availability

Data will be made available on request.

References

Agnetis, A., Cosmi, M., Nicosia, G., Pacifici, A., 2023. Two is better than one? Order aggregation in a meal delivery scheduling problem. Comput. Ind. Eng. 183, 109514.
Behrendt, A., Savelsbergh, M., Wang, H., 2023. A prescriptive machine learning method for courier scheduling on crowdsourced delivery platforms. Transp. Sci. 57 (4), 889–907.
Boland, N., Hewitt, M., Marshall, L., Savelsbergh, M., 2017. The continuous-time service network design problem. Oper. Res. 65 (5), 1303–1321.
Brown, J., 2019. First home delivery drone service launched. Available at https://www.coverdrone.com/nz/first-home-delivery-drone-service-launched/. (Accessed on 16 April 2025).
Cheng, C., Adulyasak, Y., Rousseau, L.-M., 2020. Drone routing with energy function: Formulation and exact algorithm. Transp. Res. Part B: Methodol. 139, 364–387.
Cheng, C., Adulyasak, Y., Rousseau, L.-M., 2024. Robust drone delivery with weather information. Manuf. Serv. Oper. Manag. 26 (4), 1402–1421.
Chung, S.H., Sah, B., Lee, J., 2020. Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. Comput. Oper. Res. 123, 105004.
Cicek, C.T., Koç, Ç., Gultekin, H., Erdoğan, G., 2024. Communication-aware drone delivery problem. IEEE Trans. Intell. Transp. Syst. 25 (8), 9168–9180.
Dayarian, I., Savelsbergh, M., Clarke, J.-P., 2020. Same-day delivery with drone resupply. Transp. Sci. 54 (1), 229–249.
Dorling, K., Heinrichs, J., Messier, G.G., Magierowski, S., 2016. Vehicle routing problems for drone delivery. IEEE Trans. Syst. Man Cybern.: Syst. 47 (1), 70–85.
Dukkanci, O., Kara, B.Y., Bektaş, T., 2021. Minimizing energy and cost in range-limited drone deliveries with speed optimization. Transp. Res. Part C: Emerg. Technol. 125, 102985.
Dukkanci, O., Koberstein, A., Kara, B.Y., 2023. Drones for relief logistics under uncertainty after an earthquake. European J. Oper. Res. 310 (1), 117–132.
Foodpanda.com, 2020. Foodpanda takes to the skies with pandaFly. Available at https://www.foodpanda.com/newsroom/foodpanda-takes-to-the-skies-with-pandafly/. (Accessed on 20 March 2024).
Ghelichi, Z., Gentili, M., Mirchandani, P.B., 2021. Logistics for a fleet of drones for medical item delivery: A case study for Louisville, KY. Comput. Oper. Res. 135, 105443.
Heier, B., 2023. Zipline continues to expand drone food delivery in U.S.. Available at https://foodondemand.com/05182023/zipline-continues-to-expand-drone-food-delivery-in-u-s/. (Accessed on 16 April 2025).
Huang, H., Hu, C., Zhu, J., Wu, M., Malekian, R., 2021. Stochastic task scheduling in UAV-based intelligent on-demand meal delivery system. IEEE Trans. Intell. Transp. Syst. 23 (8), 13040–13054.

Jahanshahi, H., Bozanta, A., Cevik, M., Kavuk, E.M., Tosun, A., Sonuc, S.B., Kosucu, B., Başar, A., 2022. A deep reinforcement learning approach for the meal delivery problem. Knowl.-Based Syst. 243, 108489.

Jeong, J., Ghaddar, B., Zufferey, N., Nathwani, J., 2024. Adaptive robust electric vehicle routing under energy consumption uncertainty. Transp. Res. Part C: Emerg. Technol. 160, 104529.

Klein, P.S., Schiffer, M., 2023. Electric vehicle charge scheduling with flexible service operations. Transp. Sci. 57 (6), 1605–1626.

Lera-Romero, G., Bront, J.J.M., Soulignac, F.J., 2024. A branch-cut-and-price algorithm for the time-dependent electric vehicle routing problem with time windows. European J. Oper. Res. 312 (3), 978–995.

Li, H., Wang, F., Zhan, Z., 2024. Drone routing problem with swarm synchronization. European J. Oper. Res. 314 (2), 477–495.

Lim, G.J., Zangeneh, S., Baharnemati, M.R., Assavapokee, T., 2012. A capacitated network flow optimization approach for short notice evacuation planning. European J. Oper. Res. 223 (1), 234–245.

Liu, Y., 2019. An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. Comput. Oper. Res. 111, 1–20.

Liu, Y., 2023. Routing battery-constrained delivery drones in a depot network: A business model and its optimization–simulation assessment. Transp. Res. Part C: Emerg. Technol. 152, 104147.

Liu, S., He, L., Max Shen, Z.-J., 2021. On-time last-mile delivery: Order assignment with travel-time predictors. Manag. Sci. 67 (7), 4095–4119.

Macrina, G., Pugliese, L.D., Guerriero, F., Laporte, G., 2020. Drone-aided routing: A literature review. Transp. Res. Part C: Emerg. Technol. 120, 102762.

Mahmoudinazlou, S., Kwon, C., 2024. A hybrid genetic algorithm with type-aware chromosomes for traveling salesman problems with drone. European J. Oper. Res. 318 (3), 719–739.

Mao, W., Ming, L., Rong, Y., Tang, C.S., Zheng, H., 2022. On-demand meal delivery platforms: Operational level data and research opportunities. Manuf. Serv. Oper. Manag. 24 (5), 2535–2542.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Comput. Oper. Res. 24 (11), 1097–1100.

Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transp. Res. Part C: Emerg. Technol. 54, 86–109.

Ocampo-Giraldo, D.M., Anaya-Arenas, A.M., Contardo, C., 2024. An iterative exact algorithm over a time-expanded network for the transportation of biomedical samples. INFORMS J. Comput. null.

Pasha, J., Elmi, Z., Purkayastha, S., Fathollahi-Fard, A.M., Ge, Y.-E., Lau, Y.-Y., Dulebenets, M.A., 2022. The drone scheduling problem: A systematic state-of-the-art review. IEEE Trans. Intell. Transp. Syst. 23 (9), 14224–14247.

Reyes, D., Erera, A., Savelsbergh, M., Sahasrabudhe, S., O'Neil, R., 2018. The meal delivery routing problem. Optimization Online, 6571.

Saleu, R.G.M., Deroussi, L., Feillet, D., Grangeon, N., Quilliot, A., 2022. The parallel drone scheduling problem with multiple drones and vehicles. European J. Oper. Res. 300 (2), 571–589.

Simoni, M.D., Winkenbach, M., 2023. Crowdsourced on-demand food delivery: An order batching and assignment algorithm. Transp. Res. Part C: Emerg. Technol. 149, 104055.

Steever, Z., Karwan, M., Murray, C., 2019. Dynamic courier routing for a food delivery service. Comput. Oper. Res. 107, 173–188.

Tamke, F., Buscher, U., 2021. A branch-and-cut algorithm for the vehicle routing problem with drones. Transp. Res. Part B: Methodol. 144, 174–203.

Tu, W., Zhao, T., Zhou, B., Jiang, J., Xia, J., Li, Q., 2019. OCD: Online crowdsourced delivery for on-demand food. IEEE Internet Things J. 7 (8), 6842–6854.

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., Subramanian, A., 2017. New benchmark instances for the capacitated vehicle routing problem. European J. Oper. Res. 257 (3), 845–858.

Ulmer, M.W., Thomas, B.W., Campbell, A.M., Woyak, N., 2021. The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. Transp. Sci. 55 (1), 75–100.

Wang, Z., 2018. Delivering meals for multiple suppliers: Exclusive or sharing logistics service. Transp. Res. Part E: Logist. Transp. Rev. 118, 496–512.

Wang, M., 2023. Meituan's fourth-generation drone unveiled: maximum load capacity 2.5kg, full load 10km for takeout delivery. Available at https://www.ithome.com/0/703/868.htm. (Accessed on 20 March 2024 in Chinese).

Wen, X., Wu, G., 2022. Heterogeneous multi-drone routing problem for parcel delivery. Transp. Res. Part C: Emerg. Technol. 141, 103763.

Xia, J., Wang, K., Wang, S., 2019. Drone scheduling to monitor vessels in emission control areas. Transp. Res. Part B: Methodol. 119, 174–196.

Xiao, J., Liu, X., Liu, T., Li, N., Martinez-Sykora, A., 2024. The electric vehicle routing problem with synchronized mobile partial recharging and non-strict waiting strategy. Ann. Oper. Res. 1–45.

Xu, W., 2023. Meituan's drone food deliveries on the rise. Available at https://www.chinadaily.com.cn/a/202309/19/WS6508e4fca310d2dce4bb66a8.html. (Accessed on 20 March 2024).

Yildiz, B., Savelsbergh, M., 2019. Provably high-quality solutions for the meal delivery routing problem. Transp. Sci. 53 (5), 1372–1388.

Zang, Y., Wang, M., Qi, M., 2022. A column generation tailored to electric vehicle routing problem with nonlinear battery depreciation. Comput. Oper. Res. 137, 105527.

Zheng, H., Chiu, Y.-C., 2011. A network flow algorithm for the cell-based single-destination system optimal dynamic traffic assignment problem. Transp. Sci. 45 (1), 121–137.

Zhu, W., Hu, X., Pei, J., Pardalos, P.M., 2024. Minimizing the total travel distance for the locker-based drone delivery: A branch-and-cut-based method. Transp. Res. Part B: Methodol. 184, 102950.