A Traveling Salesman Problem with Drone Stations and Speed-Optimized Drones

Ya Qiu[a], Chun Cheng[a,*], Peisong Liu[a], Lina Yu[b]

[a]School of Economics and Management, Dalian University of Technology, Dalian, 116024, China

[b]College of Urban Transportation and Logistics, Shenzhen Technology University, Shenzhen, Guangdong 518118, China

Abstract: With e-commerce expanding rapidly, last-mile delivery challenges have been exacerbated, necessitating innovative logistics to reduce operational costs and improve delivery speed. This paper investigates a traveling salesman problem with drone stations, where a truck collaborates with multiple drones docked at candidate drone stations to serve customers. In contrast to existing studies that typically assume fixed drone speeds, this work treats drone speeds as decision variables and introduces a comprehensive energy consumption model that accounts for all phases of drone flight. The objective is to jointly optimize truck routing, station selection, drone–customer assignment, and drone speed to minimize the total delivery cost. Through a speed-discretion method, we formulate the problem as a mixed-integer linear programming model and develop a tailored adaptive large neighborhood search (ALNS) algorithm. Computational experiments indicate that for large-sized instances with 80–100 customers and 16–20 candidate stations, ALNS produces solutions within 50 seconds, with average optimality gaps below 1.8% compared to Gurobi's solutions obtained under a 5000-second time limit. The results also show that the speed optimization strategy consistently outperforms fixed-speed approaches across multiple performance metrics, including total cost, service completion time, energy consumption, and service coverage.

Key words: truck and drone delivery; drone station location; drone speed optimization; adaptive large neighborhood search

1. Introduction

Last-mile delivery constitutes a costly and complex segment of the supply chain, a challenge intensified by the rapid expansion of e-commerce (Alverhed et al. 2024). Conventional delivery vehicles encounter operational barriers in both urban and rural contexts: in dense urban areas, traffic congestion delays service, while in rural regions, the long distances to widely scattered customers drive up costs. Drones offer a potential remedy, enabling vertical takeoff and direct, point-to-point flights to circumvent ground traffic and shorten delivery times. Amazon Prime Air has demonstrated drones' commercial viability, delivering over 60,000 items with delivery times accurate to within five minutes (Amazon 2025). In Asia, SF Express operates more than 500 drone routes and has delivered 5.2 million packages (Logistics Research 2025), while Meituan has completed over 400,000 deliveries

[*] Corresponding author. Email: chun.cheng@polymtl.ca.

across 53 routes in major cities (Meituan 2024). Despite their advantages, drones face inherent limitations that hinder standalone deployment. Payload restrictions confine them to small parcels, while limited battery capacity constrains flight range and operational duration. These factors make drones unsuitable for heavy or long-distance deliveries, limiting their ability to replace conventional vehicles. As a result, delivery systems with both trucks and drones have gained attention as promising alternatives, combining the speed and flexibility of aerial delivery with the capacity and reliability of ground transportation.

Murray and Chu (2015) propose the parallel drone scheduling traveling salesman problem (PDSTSP), in which trucks and drones operate independently, and show it to be most effective when many customers are within drone range of the depot. In practice, however, customers are often located beyond this range, which limits the model's applicability. To address such limitations, Kim and Moon (2018) propose using drone stations to extend drones' operational range. Compared to traditional depots, drone stations are smaller and more cost-effective facilities that provide shelter for drones and parcels, as well as capabilities for recharging or replacing drone batteries. These stations serve as intermediate infrastructure to support longer-range and more flexible drone operations.

Several companies have already begun deploying drone stations. For instance, DHL has developed drone cabinets capable of accepting packages and delivering them to designated destinations (DHL Taiwan Press 2019). Valqari, a Chicago-based company, has introduced a drone station comprising six individual storage units designed to accommodate both drones and parcels (Urban Air Mobility 2020). Valqari's station enables fully autonomous point-to-point operations, including outbound deliveries to other stations or end customers, as well as inbound acceptance of packages from drones or customers. The configurations of these stations are shown in Figure 1. Integrating drone stations with truck delivery is expected to enhance operational flexibility while reducing service completion time and overall costs (Kim and Moon 2018, Schermer et al. 2019). In such a system, the truck not only serves customers directly but also visits selected stations to offload packages. Drones docked at these locations then deliver the packages to nearby customers.

Drone energy consumption directly affects flight range, cost, and emissions. Integrating energy models into route optimization is thus crucial for fully evaluating trade-offs across multiple performance metrics (Zhang et al. 2021). Typically, a drone flight can be divided into four phases: takeoff, level flight/cruising, hovering, and landing (Kirschstein 2020). Each operational phase is associated with distinct energy consumption patterns. The disparity in payload between outbound deliveries (laden with packages) and return flights (typically unladen) leads to asymmetric energy profiles. Furthermore, hovering—necessitated by delays at synchronization points with trucks or by customer unavailability at delivery sites—constitutes an additional source of energy expenditure. Among various influence factors, travel speed plays a significant role, which further affects travel duration. Raj

(a) DHL

(b) Valqari

Figure 1 Drone delivery stations

and Murray (2020) empirically demonstrate that optimizing drone speed can reduce both truck travel distances and drone energy consumption.

In this paper, we investigate a traveling salesman problem with drone stations (TSPDS), in which a truck can directly serve customers and also deliver packages to selected/activated drone stations. At these stations, docked drones are then deployed to serve designated customers. The problem involves simultaneously determining truck routing, station activation, drone-customer assignment, and drone speed to minimize the overall costs. The main contributions of this study are as follows:

1. We propose a novel last-mile delivery model that integrates the traditional truck with newly developed drone stations. A key feature of this model is the explicit incorporation of drone energy consumption throughout the entire flight into the decision-making process. Unlike existing studies, our approach treats drone speed during level flight—a major determinant of energy consumption—as a decision variable rather than a fixed constant.
2. To solve this NP-hard problem efficiently, we develop a tailored adaptive large neighborhood search (ALNS) algorithm, which incorporates customized destroy and repair operators. Local search strategies are further integrated to enhance solution quality and computational efficiency.
3. Computational experiments on instances derived from real-world geographical data demonstrate that the ALNS efficiently generates high-quality solutions for the TSPDS. Moreover, the speed optimization strategy consistently outperforms fixed-speed approaches across multiple performance metrics.

2. Literature review

This section reviews related work on drone energy modeling, drone routing with energy considerations, and drone facility location problems.

2.1. Drone energy model

Given the complexity of drone energy consumption, which is influenced by flight patterns, environmental conditions, and operational parameters, a variety of modeling approaches have been proposed. Zhang et al. (2021) review and evaluate various drone energy models, categorizing them into three types: integrated, component-based, and regression models. Integrated models approximate total energy consumption using estimates from level flight; component models differentiate among flight phases, each with distinct energy profiles; and regression models are calibrated from field experiments to establish empirical relationships between energy consumption and operational parameters. The authors emphasize that drone speed significantly impacts energy consumption and should be explicitly incorporated into energy modeling. Kirschstein (2020) propose a component-based energy consumption model that accounts for multiple influencing factors, including drone weight, flight speed, wind conditions (both crosswind and head/tailwind), and hovering time. Building on their work, Dodge et al. (2024) further incorporate directional wind effects to model wind blowing at angles not aligned with the flight direction.

2.2. Drone routing with energy consumption

Most studies on the drone routing problem account for payload-dependent power consumption but treat drone speed as constant. Dorling et al. (2016) address a multi-trip drone routing problem with energy constraints, assuming that energy consumption during takeoff, landing, and cruising is roughly equivalent to that during hovering. While the actual energy function is a nonlinear convex function of payload, they approximate it using a linear model. In contrast, Cheng et al. (2020) develop a branch-and-cut algorithm that handles the nonlinear function directly by incorporating two types of cuts. Meng et al. (2023) consider simultaneous pickup and delivery operations by drones within a truck-drone routing problem, where energy consumption differs between the pickup and delivery phases. Meng et al. (2024) further incorporate time window constraints, which may require drones to hover at customer locations or retrieval stations, resulting in additional energy consumption influenced by both payload and waiting time. Pei et al. (2024) address the drone delivery problem with battery swapping operations. Unlike conventional one-delivery-one-swap approaches, this work incorporates dynamic battery replacement decisions based on real-time energy consumption. Yang et al. (2025) propose a bi-objective drone scheduling model that minimizes flight time and maximizes the remaining battery capacity, modeled as a linear function of payload weight and flight time.

Recently, some studies have begun treating drone speed as a decision variable to enable more realistic modeling of energy consumption. Raj and Murray (2020) study the flying sidekick traveling salesman problem with drones, incorporating energy consumption across different flight phases. They optimize drone cruising speed while treating takeoff and landing speeds as fixed. Tamke and Buscher

(2023) incorporate discrete drone speed choices into their model and account for in-route drone charging. Meskar and Ahmadi-Javid (2025) study an integrated speed optimization and multi-visit drone routing problem, which minimizes the total energy consumption. Amorosi et al. (2025) address a multi-period truck–drone routing and scheduling problem in which trucks do not directly serve customers but instead function as mobile charging stations and transport drones closer to customer locations. Drone energy consumption is modeled using fluid dynamics, and drone speeds are treated as decision variables influencing energy usage.

2.3. Drone facility location

Based on truck-drone configurations, drone facility location problems are broadly classified into three categories: (1) *drone-only systems*, where deliveries are performed exclusively by drones; (2) *independent truck-drone systems*, where trucks and drones operate separately without direct coordination; and (3) *integrated truck-drone systems*, where drones are transported and launched from trucks during the delivery process. Dukkanci et al. (2024a) provide a comprehensive literature review of the first two types of problems. They highlight that existing studies often overlook practical aspects of drone operations in their theoretical models, particularly the need to incorporate more accurate and nonlinear energy consumption functions. Additionally, Dukkanci et al. (2024b) review facility location problems involving integrated truck-drone systems. In the following, we review related studies on the first two types, as they are closer to our work.

A review of existing studies reveals that most research focuses on maximizing service coverage through optimized station placement, typically treating drone flight range or duration as fixed parameters, rather than as variables influenced by energy consumption dynamics. Huang and Savkin (2020) optimize charging station deployment for drone delivery by identifying the minimum number of stations required to meet a specified coverage ratio. This work is later extended by Pinto and Lagorio (2022), who propose a bi-objective optimization model that simultaneously minimizes both the number of charging stations and the total flight distance. Shavarani (2019) investigate a multi-level facility location-allocation problem involving drones, where relief centers and recharging stations are established to deliver emergency supplies to affected areas. They assume demand follows a Poisson distribution and is served by the nearest facility. When a drone's flight duration is insufficient to reach a customer, it may visit one or multiple recharging stations to extend its range.

Schermer et al. (2019) investigate a traveling salesman problem (TSP) variant that incorporates drone station location decisions. In their model, drones are stationed at designated sites and are deployed to deliver packages to nearby customers after being replenished by a truck. While the truck continues its route to serve other customers, the drone completes its delivery task and returns to its base station. Building on this framework, Kloster et al. (2023) examine multiple TSPs with drone stations, where a fixed number of stations are activated and drones can perform multiple missions. It is worth noting that, although our work is closely related to these two studies, they assume a fixed drone flight duration, whereas we consider that endurance is related to payload and travel speed, modeling it as a function of energy consumption.

Recognizing that drone flight range and duration are constrained by energy consumption, several studies have incorporated energy models into facility location optimization. Chauhan et al. (2019) investigate a maximum coverage facility location problem involving drones. In their model, drone-facility assignments are first determined, followed by the allocation of customers to each drone. These drones can perform multiple one-to-one deliveries, provided their energy capacity is not exceeded. The energy consumption is modeled as a linear relationship with both flight distance and carried payload. In contrast, Bruni et al. (2023) consider a nonlinear model, where energy usage depends nonlinearly on both flight time and payload. Dukkanci et al. (2021) examine a two-echelon drone delivery system where traditional vehicles transport drones to selected launch points, from which the drones operate to serve customers. In this framework, drone cruising speed becomes a decision variable, and the authors reformulate the nonlinear optimization model as a second-order cone program, subsequently solved using commercial solvers.

In summary, substantial progress has been made in modeling drone energy consumption, and recent routing studies have increasingly incorporated speed optimization to improve energy efficiency. However, in the context of drone facility location, few studies have treated drone speed as a decision variable. To address this gap, our work integrates speed optimization into the drone facility location framework, aiming to enhance both energy efficiency and service coverage.

3. Mathematical model

This section describes the problem, introduces the drone power model, and presents the mathematical model.

3.1. Problem definition

Our problem is defined on a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of arcs. Set $\mathcal{N}$ is composed of three parts: the set of customers $\mathcal{N}_c = \{1, \cdots, n\}$, the set of drone stations $\mathcal{N}_s = \{s_1, \cdots, s_m\}$, and the set of depots $\{0, n+1\}$. The starting depot is denoted by 0, and the returning depot is $n+1$, which is a copy of depot 0. Thus, we have $\mathcal{N} = \mathcal{N}_c \cup \mathcal{N}_s \cup \{0, n+1\}$. For convenience, we introduce the notations $\mathcal{N}^+ = \mathcal{N} \setminus \{n+1\}$, $\mathcal{N}^- = \mathcal{N} \setminus \{0\}$, and $\mathcal{N}' = \mathcal{N}_c \cup \mathcal{N}_s$. Set $\mathcal{A}$ is defined as $\mathcal{A} = \{(i,j) : i \in \{0\}, j \in \mathcal{N}' \text{ and } i \in \mathcal{N}', j \in \mathcal{N}^-, i \neq j\}$.

Our delivery system comprises one truck and multiple drones, each docked at designated drone stations. Each customer has a non-negative demand d_i, $i \in \mathcal{N}_c$, which does not exceed the payload capacity of a drone. In cases where a customer's demand exceeds the capacity, multiple copies of the customer can be created, each with the same location, allowing each customer to be served by a separate drone. Following the common assumption in truck-drone literature, the truck's capacity is assumed to be unlimited. The ground and aerial travel distances between arc $(i, j) \in \mathcal{A}$ are denoted as c_{ij} and $\bar{c}_{ij}$ ($\bar{c}_{ij} \leq c_{ij}$), respectively. All drones have a uniform energy capacity Θ. Each drone station can accommodate $|\mathcal{D}|$ drones. A drone delivery is defined as a round trip from the station to a customer (forward trip) and back to the same station (backward trip). The cruise altitude of drones is assumed to be h. The parameters e_t and e_d represent the truck's operational cost per unit distance and the drone's unit energy cost, respectively. The parameter f_d denotes the fixed cost associated with each dispatch of drone $d \in \mathcal{D}$, reflecting the amortized costs of drone depreciation and the construction and maintenance of drone stations.

The problem assumptions are as follows:

- All drone stations and drones have identical operational characteristics, with each station accommodating the same number of drones.
- Service times at customer sites for both the truck and drones are negligible.
- Each drone can undertake multiple trips from its station, starting each with a fully charged battery. Each station has a sufficient supply of batteries to support all dispatches.
- Forward and backward level flight speeds are assumed to be identical, as empirical evidence from Dukkanci et al. (2021) indicates that allowing asymmetric speeds offers only marginal benefits (a 0.01% reduction in energy and cost).
- Energy consumption from hovering is disregarded because no time window constraints are imposed, meaning drones do not wait at customer sites. The energy function for hovering, defined in Equation (3), is only weight-related and can be easily incorporated if mandatory waiting requirements exist.

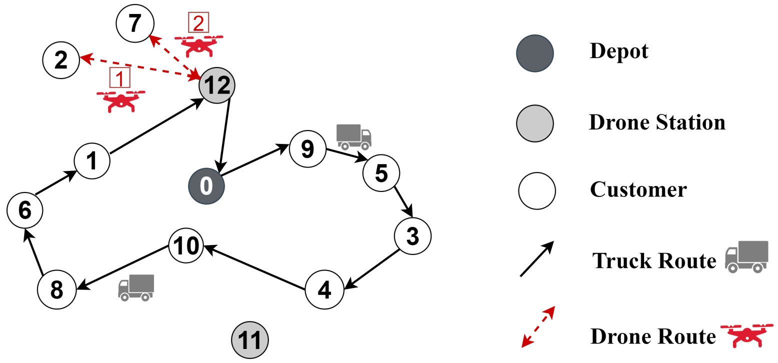


Figure 2 **Illustration of a feasible solution to the TSPDS**

The objective is to determine the optimal truck route, drone station activations, drone-to-customer assignments, and drone speeds, subject to the constraints that each customer is served exactly once and that drone energy consumption remains within the capacity limit. A feasible solution is illustrated in Figure 2, where the truck departs from the depot and sequentially visits customers $\{9,5,3,4,10,8,6,1\}$. Station 11 is not activated by the truck, and thus, no drone deliveries are initiated from it. In contrast, Station 12 is activated. Since each station accommodates $|\mathcal{D}| = 2$ identical drones, Station 12 can dispatch two drones simultaneously to serve customers 2 and 7. After dropping off the packages at Station 12, the truck returns directly to the depot.

3.2. Drone power model

We assume that multi-rotor drones are employed and adopt the power consumption models proposed by Liu et al. (2017), which is widely used in the literature (see, e.g., Raj and Murray (2020), Raj et al. (2021), Campuzano et al. (2023)). The power consumption for each flight phase is modeled as a function of payload and/or flight speed, and is expressed as follows.

- Vertical move (takeoff or landing):

$$P_v(m, v_v) = k_1(m_f + m_b + m)g\left(\frac{v_v}{2} + \sqrt{\left(\frac{v_v}{2}\right)^2 + \frac{(m_f + m_b + m)g}{k_2^2}}\right) + c_2\left((m_f + m_b + m)g\right)^{\frac{3}{2}}. \quad (1)$$

- Level flight/cruising:

$$P_l(m, v_l) = (c_1 + c_2)\left(\left((m_f + m_b + m)g - c_5(v_l \cos\alpha)^2\right)^2 + (c_4 v_l^2)^2\right)^{\frac{3}{4}} + c_4 v_l^3. \quad (2)$$

- Hovering:

$$P_h(m) = (c_1 + c_2)\left((m_f + m_b + m)g\right)^{\frac{3}{2}}. \quad (3)$$

Here, c_1, c_2, c_4, c_5, k_1, and k_2 are parameters determined by the drone's physical characteristics and operating environment. The parameter g denotes gravitational acceleration, and α is the angle of attack. The terms m_f, m_b, and m represent the drone's frame weight, battery weight, and payload weight, respectively. v_v and v_l are the speeds in the vertical and horizontal directions, respectively.

Figure 3 illustrates the power consumption characteristics during cruising under various payload–speed combinations. As expected, heavier payloads result in higher power consumption. However, the general shape of the power–speed curves remains consistent across different payload levels. Specifically, power consumption stays relatively stable at lower speeds (approximately 0–15 m/s), but rises sharply and nonlinearly at higher speeds (beyond 15 m/s). Importantly, energy consumption depends not only on power usage but also on travel time. Although higher speeds increase power consumption, they also reduce travel time, which may lead to lower overall energy use. Consequently, drone speeds should not be treated as fixed constants; instead, their optimal values vary across scenarios and should be determined endogenously within the optimization framework.

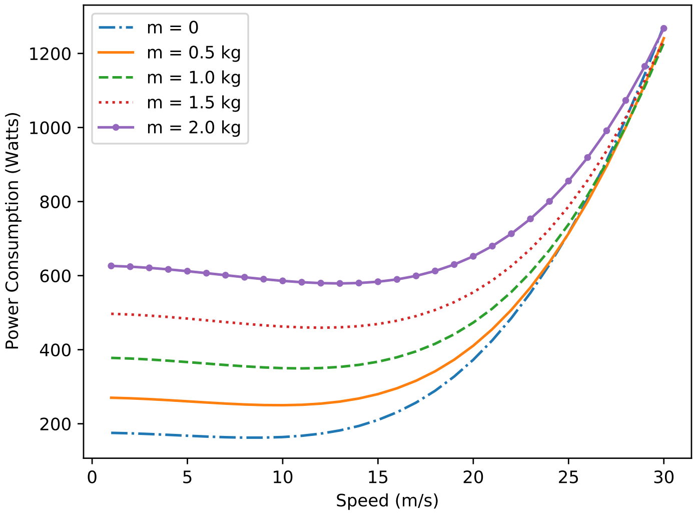


Figure 3 Power consumption during cruising under different payloads and speeds

3.3. Cost minimization model

To model the problem, we first define three sets of binary variables: $x_{ij} = 1$ if and only if arc $(i, j) \in \mathcal{A}$ is traversed by the truck, $z_s = 1$ if and only if station $s \in \mathcal{N}_s$ is activated for drone operation, and $y^s_{di} = 1$ if and only if customer $i \in \mathcal{N}_c$ is served by drone $d \in \mathcal{D}$ from station $s \in \mathcal{N}_s$. v_i is an auxiliary variable, representing the position of node $i \in \mathcal{N}^+$ in the truck's visiting sequence. We define continuous variable v^s_{di} as the speed of level flight of drone $d \in \mathcal{D}$ between station $s \in \mathcal{N}_s$ and customer $i \in \mathcal{N}_c$. Speeds during takeoff and landing are treated as fixed parameters rather than decision variables, as these phases generally consume less energy than the longer cruising segments between nodes (Raj and Murray 2020). Moreover, drone flight controllers using the MAVLink protocol do not provide options for configuring takeoff or landing speeds (MAVLink 2020).

The power function in (2) is nonlinear with respect to the speed variable v^s_{di}. To address this, we linearize the function by discretizing the speed variable, following the approach in Bektaş and Laporte (2011) and Cheng et al. (2017). We denote the lower and upper bounds of drone speed by $\underline{v}$ and $\bar{v}$, respectively. We then define a set of speed levels $\mathcal{R} = \{1, \cdots, |\mathcal{R}|\}$, where each level $r \in \mathcal{R}$ corresponds to a speed interval $[\underline{v}_r, \bar{v}_r]$ with $\underline{v}_1 = \underline{v}$ and $\bar{v}_{|\mathcal{R}|} = \bar{v}$. The speeds also satisfy $\bar{v}_r = \underline{v}_{r+1}$, $r \in \{1, \cdots, |\mathcal{R}| - 1\}$. The average speed of each level $r \in \mathcal{R}$ can be calculated as $v'_r = (\underline{v}_r + \bar{v}_r)/2$. We introduce a binary variable $w^{sr}_{di} = 1$ if and only if customer $i \in \mathcal{N}_c$ is served by drone $d \in \mathcal{D}$ from station $s \in \mathcal{N}_s$ at speed level $r \in \mathcal{R}$. For notational simplicity, we denote the power consumption between a station and customer $i \in \mathcal{N}_c$ during the level flight as $P_l(\cdot, v'_r)$ if a speed level $r \in \mathcal{R}$ is utilized. Similarly, $P_v(\cdot, v^t_v)$ and $P_v(\cdot, v^l_v)$ denote the power consumption during the takeoff and landing stages with a takeoff speed v^t_v and a landing speed v^l_v, respectively.

The mixed-integer linear programming (MILP) model for the TSPDS is constructed as follows:

$$\begin{aligned}\min\ & e_t \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + f_d \sum_{s\in\mathcal{N}_s}\sum_{d\in\mathcal{D}}\sum_{i\in\mathcal{N}_c} y^s_{di} \\ & + e_d \sum_{s\in\mathcal{N}_s}\sum_{d\in\mathcal{D}}\sum_{i\in\mathcal{N}_c} \Bigg\{ \left(P_v(d_i, v^t_v)\frac{h}{v^t_v} + P_v(d_i, v^l_v)\frac{h}{v^l_v} + P_v(0, v^t_v)\frac{h}{v^t_v} + P_v(0, v^l_v)\frac{h}{v^l_v} \right) y^s_{di} \\ & \qquad\qquad + \sum_{r\in\mathcal{R}} \left(P_l(d_i, v'_r) + P_l(0, v'_r)\right) \frac{\bar{c}_{si}}{v'_r} w^{sr}_{di} \Bigg\},\end{aligned} \tag{4}$$

$$\text{s.t.} \quad \sum_{\substack{i\in\mathcal{N}^+ \\ i\neq j}} x_{ij} + \sum_{s\in\mathcal{N}_s}\sum_{d\in\mathcal{D}} y^s_{dj} = 1 \qquad \forall j\in\mathcal{N}_c, \tag{5}$$

$$\sum_{j\in\mathcal{N}'} x_{0j} = \sum_{i\in\mathcal{N}'} x_{i,n+1} = 1, \tag{6}$$

$$\sum_{\substack{i\in\mathcal{N}^+ \\ i\neq k}} x_{ik} = \sum_{\substack{j\in\mathcal{N}^- \\ j\neq k}} x_{kj} \qquad \forall k\in\mathcal{N}', \tag{7}$$

$$u_i - u_j + 1 \leq |\mathcal{N}|(1 - x_{ij}) \quad \forall i,j\in\mathcal{N}^+, i\neq j, \tag{8}$$

$$u_0 = 1, \tag{9}$$

$$\sum_{\substack{i\in\mathcal{N}^+ \\ i\neq s}} x_{is} = z_s \qquad \forall s\in\mathcal{N}_s, \tag{10}$$

$$y^s_{di} \leq z_s \qquad \forall i\in\mathcal{N}_c, d\in\mathcal{D}, s\in\mathcal{N}_s, \tag{11}$$

$$\sum_{r\in\mathcal{R}} w^{sr}_{di} = y^s_{di} \qquad \forall i\in\mathcal{N}_c, d\in\mathcal{D}, s\in\mathcal{N}_s, \tag{12}$$

$$\begin{aligned}& \left(P_v(d_i, v^t_v)\frac{h}{v^t_v} + P_v(d_i, v^l_v)\frac{h}{v^l_v} + P_v(0, v^t_v)\frac{h}{v^t_v} + P_v(0, v^l_v)\frac{h}{v^l_v} \right) y^s_{di} \\ & + \sum_{r\in\mathcal{R}} \left(P_l(d_i, v'_r) + P_l(0, v'_r)\right) \frac{\bar{c}_{si}}{v'_r} w^{sr}_{di} \leq \Theta \qquad \forall i\in\mathcal{N}_c, d\in\mathcal{D}, s\in\mathcal{N}_s,\end{aligned} \tag{13}$$

$$x_{ij}\in\{0,1\} \qquad \forall (i,j)\in\mathcal{A}, \tag{14}$$

$$z_s, y^s_{di}, w^{sr}_{di} \in \{0,1\} \qquad \forall i\in\mathcal{N}_c, d\in\mathcal{D}, s\in\mathcal{N}_s. \tag{15}$$

The objective function (4) minimizes the total cost, comprising the truck's routing cost, the fixed cost of deploying drones, and the drones' energy consumption cost. Constraints (5) impose that each customer is visited exactly once. Constraints (6)–(9) define the truck's route. Specifically, constraints (6) ensure that the truck departs from and returns to the depot exactly once. Constraints (7) impose the flow conservation at customers and stations. Constraints (8)–(9) eliminate subtours using Miller-Tucker-Zemlin formulation. Constraints (10) indicate that we only activate the station when the truck visits it and drops packages. Constraints (11) further impose that we can only dispatch drones from an activated station. Constraints (12) link variables $\boldsymbol{w}$ and $\boldsymbol{y}$. Constraints (13) impose that the drones' energy capacity limit must be respected. Constraints (14)–(15) restrict variables to be binary.

To accelerate the resolution of the MILP model, we introduce the following valid inequalities, which eliminate symmetry issues stemming from the identical drones at a station.

$$\sum_{i \in \mathcal{N}_c} y^s_{d-1,i} \geq \sum_{i \in \mathcal{N}_c} y^s_{d,i} \qquad \forall s \in \mathcal{N}_s, d \in \{2, \ldots. |\mathcal{D}|\} \tag{16}$$

Constraints (16) enforce that drone d at station s can be used only if drone $d-1$ is also used.

4. Solution method

Since the TSPDS is NP-hard, commercial solvers such as Gurobi often fail to produce satisfying solutions for large-sized instances within a reasonable time. To achieve scalability, we develop an ALNS algorithm. The ALNS method, introduced by Ropke and Pisinger (2006), has been widely used for routing optimization problems. A comprehensive review by Voigt (2025) shows that ALNS consistently yields high-quality solutions across a variety of routing contexts, owing to its flexible operator selection mechanism and adaptive search strategy. Given that our problem involves station activation decisions beyond the scope of traditional routing, we develop a tailored ALNS algorithm with customized enhancements. Specifically, we design problem-specific operators and integrate local search strategies to further improve solution quality and computational efficiency.

The overall algorithmic framework is outlined in Algorithm 1. The process begins with the construction of an initial solution using a spatial clustering-based greedy strategy. The main optimization phase consists of iterative destroy-and-repair cycles, governed by a simulated annealing acceptance criterion. To preserve high-quality solutions, an elite solution pool is maintained, while restart mechanisms are triggered to mitigate search stagnation. Additionally, we periodically employ local search methods to enhance solution quality.

4.1. Initial solution

We employ a spatial clustering-based heuristic to construct initial solutions. Specifically, we first implement K-means clustering algorithm to group customers based on their spatial proximity. Then, the following steps are executed to generate the initial solution.

(1) **Station activation**: Since the clustering is based on customer proximity, a cluster may or may not contain candidate stations. If multiple candidates are present within a cluster, we compute the number of customers that each candidate can cover and activate the station with the highest coverage. In the event of a tie, the station with the shortest average flight distance to its reachable customers is selected.

(2) **Customer assignment**: After station activation, customer allocation proceeds in two stages. In the first stage, for each activated station, reachable customers are sorted by ascending flight distance. Customers are then sequentially assigned to drones docked at the station, beginning with the closest, until all customers within the drones' energy capacity are allocated. In the

Algorithm 1: ALNS Framework

```
Input: Problem instance, algorithm parameters
Output: Optimized solution $S^*$ and cost
Generate an initial solution;                                         // Section 4.1
Set current solution $S$, optimal solution $S^*$, temperature $T$, operator weights;
Initialize elite pool $\mathcal{E} \leftarrow \emptyset$;
for iteration ← 1 to $K_{max}$ do
    Select destroy and repair operators based on adaptive weights;
    $S' \leftarrow$ Apply destroy and repair operators on $S$;           // Sections 4.2, 4.3
    if accept($S'$, $S$, $T$) then
        $S \leftarrow S'$;
        if $f(S) < f(S^*)$ then
            $S^* \leftarrow S$;
    Update elite pool $\mathcal{E}$ with the current solution $S$;
    Update operator weights based on performance;
    if stagnation detected then
        $S \leftarrow$ Select and perturb solution from $\mathcal{E}$;
    if iteration % $K_{local}$ = 0 then
        $S'' \leftarrow$ Local Search($S$);                              // Section 4.4
        if $f(S'') < f(S^*)$ then
            $S^* \leftarrow S''$;
        Update elite pool $\mathcal{E}$ with $S''$;
    Update temperature $T$;
return $S^*$ and the correpsonding cost
```

second stage, any remaining unassigned customers—including those not covered in the first stage and those belonging to clusters without activated drone stations—are reassigned to the nearest reachable station within the drone's service range.

(3) **Truck route construction**: For customers that cannot be served by drones, a truck route is constructed. All activated drone stations are first inserted into the truck route. The remaining customers are then added using a greedy insertion heuristic, whereby each customer is placed at the position that yields the smallest increase in total detour distance. Finally, we perform 2-opt local search to further improve the route.

4.2. Destroy operators

This section describes the destroy operators used to remove customers from the current solution.

randomRemove operator. This operator randomly removes customers from the current solution. If a station no longer serves any customers after customer removal, it is marked as inactive and deleted from the truck route.

worstRemove operator. This operator evaluates the potential cost savings associated with removing each customer and prioritizes those with the highest reductions. For customer j assigned to a drone from station s, the associated cost reduction after removal is

$$C_j^{\text{drone}}(s) = C_{\text{dispatch}} + C_{\text{energy}}(j,s) + C_{\text{detour}}(s). \tag{17}$$

Here, C_{dispatch} denotes the fixed cost of dispatching a drone, $C_{\text{energy}}(j,s)$ represents the energy cost for the drone to serve customer j from station s, and $C_{\text{detour}}(s)$ accounts for the truck's detour cost to visit station s. Note that $C_{\text{detour}}(s)$ is included only when station s exclusively serves customer j; in such cases, removing the customer eliminates the need for the truck to visit the station, thus yielding additional savings. For truck-served customers, the cost reduction is calculated as

$$C_j^{truck} = e_t(d_{ij} + d_{ji'} - d_{ii'}), \tag{18}$$

where the predecessor and successor nodes of customer j in the truck route are represented by i and i'. To avoid repeatedly removing the same customers, the operator ranks all customers in descending order based on their estimated cost savings. These values are then normalized to form a probability distribution, and a roulette wheel selection mechanism is applied, giving higher removal probabilities to customers associated with greater cost reductions.

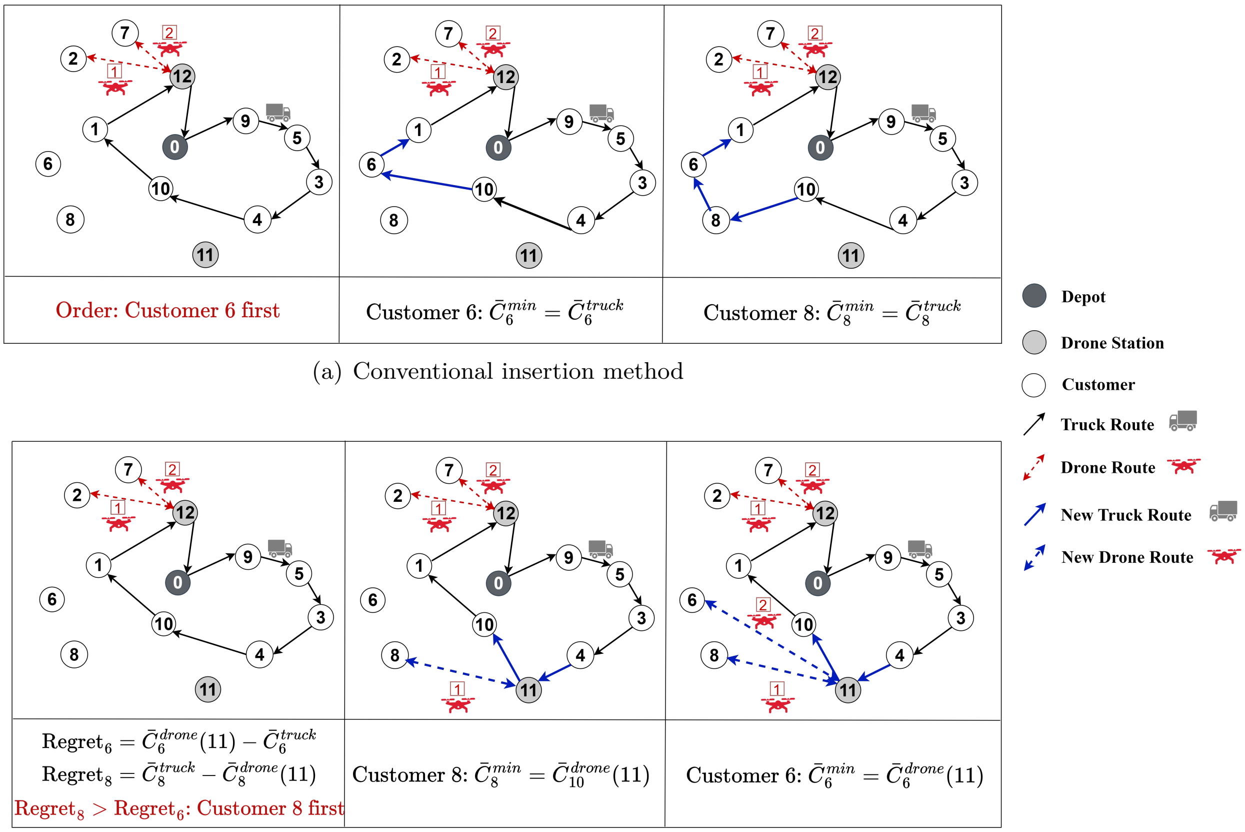
4.3. Repair operators

This section describes the repair operators used to reinsert the removed customers into the routes.

greedyInsertion operator. This operator reconstructs a feasible solution by sequentially inserting unassigned customers based on their indices. For each customer j, both truck and drone service options are evaluated. Insertion cost increments are computed according to the logic outlined in Equations (17) and (18). For truck service, the operator identifies the position that minimizes the route cost increase, denoted by $\bar{C}_j^{truck}$. For drone service, the cost increase $\bar{C}_j^{\text{drone}}(s)$ is computed for each available station, where the detour cost $\bar{C}_{\text{detour}}(s)$ is included only if activating a new station is required. The customer is then assigned to the mode yielding the lowest cost increase, i.e., the mode associated with $\min\{\bar{C}_j^{\text{truck}}, \min_{s\in\mathcal{N}_s} \bar{C}_j^{\text{drone}}(s)\}$.

regretInsertion operator. This operator determines the insertion order of unassigned customers based on the well-established k-regret insertion strategy (Sarasola and Doerner 2020). We set $k = 1$ to maintain computational efficiency while ensuring solution quality, considering only the difference in cost increase between the best and second-best insertion options. For each unassigned customer j, the regret value is computed as

$$\text{Regret}_j = \bar{C}_j^{\text{second}} - \bar{C}_j^{\text{min}},$$


Order: Customer 6 first
Customer 6: $\bar{C}_6^{min} = \bar{C}_6^{truck}$
Customer 8: $\bar{C}_8^{min} = \bar{C}_8^{truck}$
Regret$_6 = \bar{C}_6^{drone}(11) - \bar{C}_6^{truck}$
Regret$_8 = \bar{C}_8^{truck} - \bar{C}_8^{drone}(11)$
Regret$_8 >$ Regret$_6$: Customer 8 first
Customer 8: $\bar{C}_8^{min} = \bar{C}_{10}^{drone}(11)$
Customer 6: $\bar{C}_6^{min} = \bar{C}_6^{drone}(11)$
Depot
Drone Station
Customer
Truck Route
Drone Route
New Truck Route
New Drone Route

(a) Conventional insertion method

(b) Regret-based insertion method

Figure 4 **Illustration of repair operators with and without regret value consideration**

where $\bar{C}_j^{\text{min}}$ and $\bar{C}_j^{\text{second}}$ represent the lowest and second-lowest increases of insertion costs among all feasible service options, respectively. The operator processes customers in descending order of regret values, prioritizing those whose delayed insertion would incur the greatest regret.

To illustrate this operator, we consider the 10-customer instance depicted in Figure 2. Suppose that customers 6 and 8 have been removed and are now to be reinserted. Figure 4 compares the outcomes of the conventional insertion method and the regret-based prioritization insertion. As shown in Figure 4(a), the conventional insertion strategy processes customers in order of their indices—customer 6 is inserted first and assigned the lowest-cost truck service, followed by customer 8, who is also allocated to truck service based on minimal cost. In contrast, Figure 4(b) illustrates the regret-based insertion strategy, where customer 8 is prioritized due to a higher regret value. Customer 8 is assigned to station 11 via drone service, thereby activating the station. As a result, customer 6 can also be served by the newly activated station 11 at a lower cost than the truck option.

collaborativeInsertion operator. The previous two repair operators may frequently activate new stations to achieve locally optimal insertion costs, which may increase truck travel distance and lead to structurally similar solutions. To mitigate this, this operator prioritizes the use of already activated stations, aiming to avoid excessive new station deployments. Unassigned customers are first reassigned to the nearest activated station within the drone's service range; any remaining customers are then assigned to truck service.

4.4. Local search operators

The local search component is integrated within the ALNS framework and is periodically invoked after destroy-repair cycles. It explores neighborhood structures in the current solution to facilitate incremental improvements. Unlike the destroy-repair operations that utilize simulated annealing acceptance criteria, the local search component accepts only improved moves.

2-opt operator. This operator randomly selects two positions along the truck route and reverses the segment between them. It performs a predetermined number of trials, evaluating the route cost following each exchange and retaining the configuration that yields the lowest cost.

serviceModeSwitch operator. This operator evaluates potential cost savings by switching customers between truck and drone services. To enhance efficiency, it focuses on two types of candidate customers: (1) those served by drones in the current solution but by trucks in the best-known solution, and vice versa; and (2) randomly selected customers to introduce search diversification and help escape local optima. For each candidate customer, the operator evaluates the cost change associated with switching to the alternative service mode. A greedy strategy is employed: any switch that yields a cost reduction is immediately accepted.

stationSwap operator. In TSPDS, stations can serve an unlimited number of customers through multiple drone trips, which may result in uneven station utilization—some stations accumulate many customers while others remain underused. Since destroy operators only remove a fixed percentage of customers, highly loaded stations often remain activated even after partial customer removal, potentially trapping the search in local optima with suboptimal station configurations. To mitigate this, the `stationSwap` operator randomly selects an activated station for complete removal. All its assigned customers are then reassigned to other activated stations if feasible; otherwise, they are inserted into the truck route.

5. Computational experiments

In this section, we present the test instances, evaluate the ALNS algorithm, and analyze the effect of speed optimization. The ALNS algorithm was implemented in Python 3.11, and the MILP model was solved using Gurobi 11.0.1. Experiments were conducted on a laptop equipped with a 2.10 GHz Intel(R) Core(TM) i5-13420H processor and 16 GB of memory under the Windows 11 operating system. The time limit for Gurobi was set to 5000 seconds, and all computation times are reported in seconds.

5.1. Parameters and instances

The dataset employed in this study derives from real-world geographical data of the Kartal district, Istanbul, Turkey (Kilci et al. 2018). The original instance includes 45 nodes: one warehouse, three schools, four hospitals, and 37 population centers. In our setting, the warehouse serves as the distribution center, the schools and hospitals form the seven candidate drone stations, and the population centers represent the customer locations. While preserving the original network structure, we expand the regional coverage by a factor of three, scaling the number of stations to 24 and customer sites to 120 to form the comprehensive dataset for this study. Test instances are generated with varying sizes, where $(|\mathcal{N}_c|, |\mathcal{N}_s|) \in \{(10,2),(20,4),(30,6),(40,8),(60,12),(80,16),(90,18),(100,20)\}$. For each configuration, 5 instances are randomly generated.

The parameter values for the drone power models are set as those in Liu et al. (2017): $k_1 = 0.8554$ (dimensionless), $k_2 = 0.3051\sqrt{kg/m}$, $c_1 = 2.8037\sqrt{m/kg}$, $c_2 = 0.3177\sqrt{m/kg}$, $c_4 = 0.0296\ kg/m$, and $c_5 = 0.0279\ Ns/m$. Based on Raj and Murray (2020), we set the vertical takeoff speed to $v_v^t = 10\ m/s$ and the landing speed to $v_v^l = 5\ m/s$. The battery capacity of drones is $\Theta = 0.1389\ kWh$. Following Zeng et al. (2019), the horizontal flight speed range is set to $\underline{v} = 1\ m/s$ to $\bar{v} = 30\ m/s$. The speed is discretized into $|\mathcal{R}| = 20$ levels. The drone cruise altitude is fixed at $h = 100\,m$, complying with the FAA commercial small UAV altitude limit of approximately $120\ m$.

For cost parameters, the unit energy cost is set to $e_d = 0.1644\,\$/kWh$ based on current electricity rates (ChooseEnergy 2024). The truck's unit operational cost is set at $e_t = 1.28\,\$/km$ according to trucking freight rates (Dynamic Logistix 2024). The drone dispatch cost is calculated as $f_d = 2.69\,\$$, following the methodology in an industry report on logistics drones (Kaiyuan Securities 2024), considering a small-to-medium multi-rotor drone priced at \$5,400 (Aurelia Aerospace 2024) with an estimated 2,500 dispatches per drone, and a Valqari automated drone station costing \$35,000 (The Spoon 2024). Additional parameter settings include two standby drones at each station, truck-to-drone distance ratio $c_{ij} = 1.3\bar{c}_{ij}$ for all $(i,j) \in \mathcal{A}$, and truck speed $v_t = 10\ m/s$. Customer demands (in kg) are assumed to follow a uniform distribution over $[0.45, 2.0]$.

5.2. Performance of ALNS

To validate the proposed ALNS, we compare its performance against Gurobi. Given the sensitivity of ALNS performance to parameter settings, we performed preliminary experiments to optimize the parameters. Their final values are provided in Appendix A.

The comparison results are reported in Tables 1 and 2. Column *Best* reports the best solution identified by Gurobi within the time limit. Column *Gap* indicates the optimality gap between the upper and lower bounds generated by Gurobi at the time of termination, where a gap of 0 indicates that the solution has been proven optimal. Column *Time* shows the computational time consumed by Gurobi. Each instance is solved five times by ALNS to mitigate the effect of randomness.

Columns *ALNS_best* and *ALNS_avg* report the best and the average results obtained from the five runs. Correspondingly, columns *Bst_gap* and *Avg_gap* reports the percentage gaps, calculated as $Bst\_gap = (ALNS\_best - Best)/Best \times 100$ and $Avg\_gap = (ALNS\_avg - Best)/Best \times 100$, respectively. Column *Avg_time* indicates the average computational time of ALNS.

Table 1 Performance comparison between ALNS and Gurobi on small-sized instances

	Gurobi			ALNS				
Instance	Best	Gap(%)	Time(s)	ALNS_best	ALNS_avg	Bst_gap(%)	Avg_gap(%)	Avg_time(s)
10_2_1	32.85	0.00	0.03	32.85	32.85	0.00	0.00	1.46
10_2_2	35.86	0.00	0.02	35.86	35.86	0.00	0.00	0.95
10_2_3	36.63	0.00	0.19	36.63	36.63	0.00	0.00	0.83
10_2_4	30.34	0.00	0.03	30.34	30.34	0.00	0.00	1.41
10_2_5	36.47	0.00	0.04	36.47	36.47	0.00	0.00	1.33
20_4_1	51.23	0.00	0.71	51.23	51.70	0.00	0.92	3.50
20_4_2	59.00	0.00	0.61	59.00	59.00	0.00	0.00	3.77
20_4_3	57.54	0.00	0.45	57.54	57.54	0.00	0.00	4.56
20_4_4	58.86	0.00	0.45	58.86	59.39	0.00	0.91	3.91
20_4_5	60.29	0.00	1.84	60.29	60.29	0.00	0.00	3.70
30_6_1	81.61	0.00	88.33	82.00	82.92	0.48	1.60	5.31
30_6_2	79.30	0.00	32.08	79.30	79.48	0.00	0.23	6.17
30_6_3	78.31	0.00	0.97	78.43	80.22	0.15	2.44	4.41
30_6_4	78.85	0.00	1.66	78.98	79.53	0.16	0.86	4.88
30_6_5	82.79	0.00	13.54	82.79	82.79	0.00	0.00	3.92
40_8_1	97.22	0.00	259.80	97.79	98.59	0.59	1.41	6.77
40_8_2	95.80	0.00	10.59	97.17	98.32	1.43	2.63	5.80
40_8_3	99.20	0.00	364.23	99.62	99.74	0.42	0.54	5.86
40_8_4	98.33	0.00	2.69	98.64	99.75	0.32	1.44	6.39
40_8_5	95.00	0.00	215.44	95.24	96.39	0.25	1.47	7.21
60_12_1	124.46	0.00	50.02	126.17	127.29	1.37	2.27	13.21
60_12_2	131.06	0.00	200.94	132.10	132.58	0.79	1.16	15.26
60_12_3	128.60	0.00	142.06	128.60	130.17	0.00	1.22	23.38
60_12_4	126.93	0.00	199.89	127.04	129.07	0.09	1.68	12.05
60_12_5	130.55	0.00	135.98	133.11	133.63	1.96	2.36	13.12
Average	79.48	0.00	68.90	79.84	80.42	0.32	0.93	6.45

Table 1 shows that for instances with 10–20 customers, ALNS finds the optimal solutions in about 5 seconds. For instances with 30–60 customers, it produces near-optimal solutions with an average optimality gap of 0.53% compared to those obtained by Gurobi. Across all small-sized instances, the average gap remains at 0.32%. In terms of computational time, Gurobi consumes 68.90 seconds on average, while ALNS only requires 6.45 seconds. Table 2 further presents a comparative analysis using large-sized instances. The results show that ALNS can generate high-quality solutions in a fraction of Gurobi's computation time. On average, the best solutions identified by ALNS show a gap of only 1.78% relative to Gurobi's best results, with an average gap of 2.85% across all runs and instances. In terms of computational time, Gurobi takes 4128.84 seconds on average, whereas ALNS requires only 36.16 seconds. These results confirm that the proposed ALNS can efficiently produce high-quality solutions for the TSPDS.

Table 2 Performance comparison between ALNS and Gurobi on large-sized instances

	Gurobi			ALNS				
Instance	Best	Gap(%)	Time(s)	ALNS_best	ALNS_avg	Bst_gap(%)	Avg_gap(%)	Avg_time(s)
80_16_1	145.91	0.00	3886.08	151.43	152.71	3.78	4.66	30.43
80_16_2	146.33	0.00	2910.17	148.59	149.78	1.55	2.36	32.65
80_16_3	148.12	0.00	2146.95	149.04	151.27	0.62	2.13	27.64
80_16_4	144.82	0.90	5000.85	148.60	150.42	2.61	3.86	27.61
80_16_5	150.82	0.60	5003.33	152.05	154.50	0.82	2.44	40.46
90_18_1	157.07	0.00	1156.27	159.38	161.04	1.47	2.53	32.91
90_18_2	161.10	0.00	2013.65	163.81	165.20	1.68	2.54	31.25
90_18_3	164.24	1.24	5000.57	167.13	169.81	1.76	3.39	32.96
90_18_4	160.66	0.48	5000.49	162.36	163.32	1.06	1.66	33.25
90_18_5	166.67	0.00	4811.40	168.79	170.22	1.27	2.13	32.89
100_20_1	170.07	0.70	5000.50	172.33	174.16	1.33	2.40	47.56
100_20_2	173.78	0.68	5000.51	178.51	179.79	2.72	3.46	45.45
100_20_3	170.86	0.14	5000.66	175.11	176.57	2.49	3.34	42.84
100_20_4	171.23	0.78	5000.58	174.39	177.66	1.85	3.76	39.72
100_20_5	171.77	2.76	5000.59	174.76	175.34	1.74	2.08	41.80
Average	160.23	0.55	4128.84	163.09	164.79	1.78	2.85	36.16

5.3. Value of speed optimization

In this section, we evaluate the potential benefits of speed optimization by mainly considering two key performance indicators: cost and time. To this end, alongside the cost minimization model presented in Section 3.3, we introduce an additional time minimization model, detailed in Appendix B, which aims to minimize the system's service completion time.

Table 3 Performance comparison under the cost minimization objective with different speed strategies

Customers	Speed Strategy	Cost($)	#Cus$_d$	Avg Energy(kJ)	Avg Radius(m)	Max Radius(m)
20	Optimization	**57.38**	**8.8**	**243.57**	**6163.73**	**8674.05**
	Fixed 15 m/s	57.41	**8.8**	256.25	**6163.73**	**8674.05**
	Fixed 20 m/s	57.45	**8.8**	273.50	**6163.73**	**8674.05**
	Fixed 25 m/s	59.88	7.6	306.80	4700.27	6685.21
	Fixed 30 m/s	69.82	4.8	340.28	3916.15	5031.18
30	Optimization	**80.17**	**10.0**	**239.51**	**5613.97**	**8320.01**
	Fixed 15 m/s	80.23	**10.0**	247.13	5465.46	8103.48
	Fixed 20 m/s	80.26	**10.0**	260.36	5465.46	8103.48
	Fixed 25 m/s	81.15	**10.0**	325.66	5280.20	7402.95
	Fixed 30 m/s	89.93	7.0	338.43	3807.47	5325.19
40	Optimization	**97.11**	9.2	296.73	**6626.24**	**10454.52**
	Fixed 15 m/s	97.28	9.4	274.00	5926.12	8888.91
	Fixed 20 m/s	97.32	**9.6**	**242.62**	4832.69	8214.84
	Fixed 25 m/s	97.53	**9.6**	296.44	4597.44	7669.29
	Fixed 30 m/s	101.00	9.4	325.06	3532.51	4989.09
60	Optimization	**128.32**	**6.0**	292.11	**6622.78**	**8840.80**
	Fixed 15 m/s	128.47	**6.0**	**290.09**	6100.20	8301.98
	Fixed 20 m/s	128.49	5.8	299.07	5959.38	8301.98
	Fixed 25 m/s	129.16	5.2	336.54	4956.69	6498.10
	Fixed 30 m/s	131.18	4.2	336.73	3545.74	4331.89

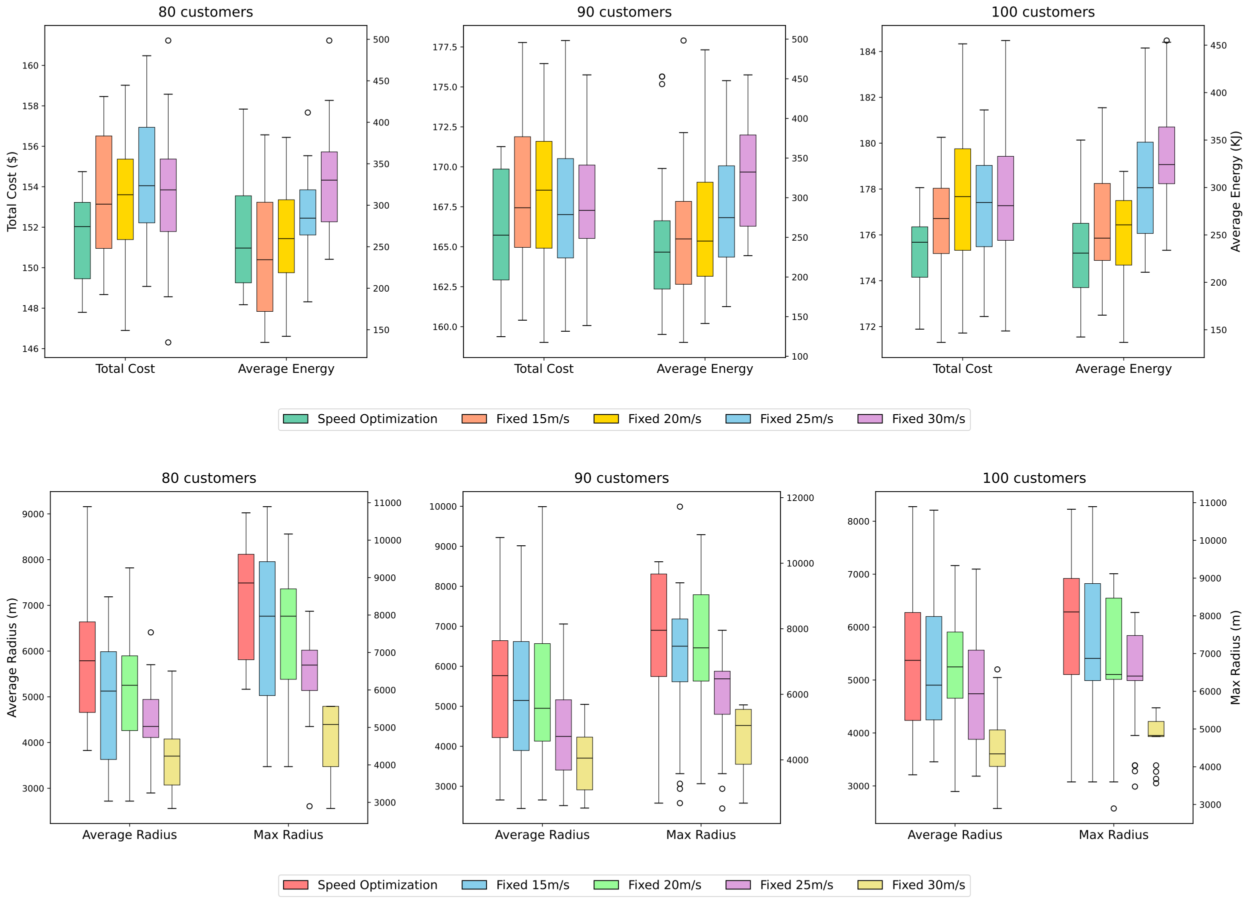


Figure 5 Performance comparison of different speed strategies for instances with 80–100 customers

5.3.1. Cost performance analysis. To assess the impact of speed optimization on cost, we solve the model in Section 3.3 for instances with 20 to 100 customers. The results are benchmarked against fixed-speed strategies in which drone speeds are set to 15, 20, 25, and 30 m/s, respectively. Table 3 reports the results for instances with 20–60 customers, solved using Gurobi. Figure 5 provides box plot comparisons for instances with 80–100 customers, based on 5 independent runs of the ALNS algorithm. In addition to the total cost, we report several drone-related performance indicators, including the number of customers served by drones (denoted as $\#Cus_d$), average energy consumption, average coverage radius, and maximum coverage radius. The reported values are averaged over five instances for each customer size. In the table, the best-performing value for each performance indicator is highlighted in bold.

As shown in Table 3, speed optimization achieves the minimum total cost for all tested instance sizes. For example, in the 30-customer case, it reduces the total cost to 80.17 $, compared to 89.93 $ under the fixed speed of 30 m/s. In terms of drone-related indicators, the speed optimization strategy also provides promising results. Specifically, it consumes less energy for 20- and 30-customer instances. Moreover, it enables drones to achieve broader service coverage, as shown in the last two columns of the table. A similar pattern is also observed in Figure 5, where the speed optimization strategy consistently yields the lowest total costs and the highest service coverage.

To further elucidate the impact, Figure 6 depicts the truck and drone routes for a 20-customer instance. The speed-optimized solution exhibits two key distinctions from its fixed-speed counterparts. First, the decisions regarding station activation and vehicle routing can differ significantly. For instance, three stations are activated when the drone speed is fixed at 30 m/s, whereas only two are deployed in other scenarios. Moreover, even with identical station activations, the resulting truck and drone routes may vary. As an example, under the optimized strategy, stations 21 and 23 serve nine customers, while the same stations serve only six customers when the drone speed is fixed at 25 m/s. Second, even with identical station and routing decisions (e.g., the 15 and 20 m/s cases), the speed-optimization method assigns heterogeneous speeds to individual customer trips, thereby reducing total costs.

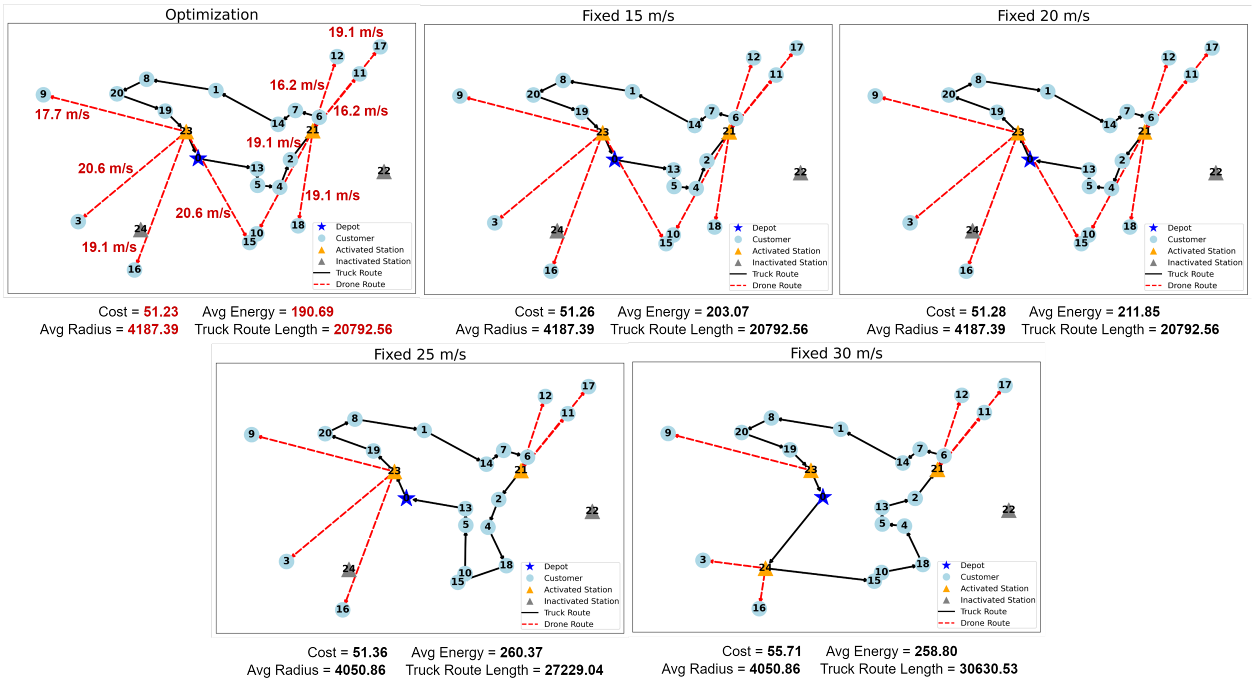

Optimization
19.1 m/s
16.2 m/s
16.2 m/s
17.7 m/s
19.1 m/s
20.6 m/s
19.1 m/s
20.6 m/s
19.1 m/s
Cost = 51.23 Avg Energy = 190.69
Avg Radius = 4187.39 Truck Route Length = 20792.56
Fixed 15 m/s
Cost = 51.26 Avg Energy = 203.07
Avg Radius = 4187.39 Truck Route Length = 20792.56
Fixed 20 m/s
Cost = 51.28 Avg Energy = 211.85
Avg Radius = 4187.39 Truck Route Length = 20792.56
Fixed 25 m/s
Cost = 51.36 Avg Energy = 260.37
Avg Radius = 4050.86 Truck Route Length = 27229.04
Fixed 30 m/s
Cost = 55.71 Avg Energy = 258.80
Avg Radius = 4050.86 Truck Route Length = 30630.53
Depot
Customer
Activated Station
Inactivated Station
Truck Route
Drone Route

Figure 6 **Routes under the cost minimization objective with different speed strategies for a 20-customer instance**

5.3.2. Time performance analysis. This section analyzes the impact of different speed strategies on the system's service completion time using the model in Appendix B. Since the time minimization model includes more variables and constraints than the cost minimization model, our analysis is restricted to instances with 10 to 20 customers, for which Gurobi can produce optimal or near-optimal solutions (with an optimality gap within 1%) within the 5000-second time limit. The results

Table 4 **Performance comparison under the time minimization objective with varying speed strategies**

Customers	Speed Strategy	Total Time(s)	Time Saving(%)	Total Cost($)	Cost Saving(%)
10	Optimization	1494.60	0.00	46.38	0.00
	Fixed 15 m/s	1742.17	14.15	46.75	0.80
	Fixed 20 m/s	1631.97	8.36	45.95	-0.92
	Fixed 25 m/s	1740.31	14.06	51.27	9.54
	Fixed 30 m/s	2402.56	37.75	66.62	30.39
15	Optimization	1156.18	0.00	59.58	0.00
	Fixed 15 m/s	1588.40	27.21	64.88	8.17
	Fixed 20 m/s	1378.43	16.12	63.40	6.02
	Fixed 25 m/s	1264.02	8.53	60.28	1.16
	Fixed 30 m/s	2682.11	56.89	80.96	26.41
20	Optimization	1453.92	0.00	73.63	0.00
	Fixed 15 m/s	1822.99	20.25	74.92	1.73
	Fixed 20 m/s	1732.95	16.10	74.59	1.29
	Fixed 25 m/s	1484.89	2.09	79.23	7.07
	Fixed 30 m/s	2666.19	45.47	94.62	22.19

are reported in Table 4, where $Saving$ is computed as $(fixed - opt)/fixed \times 100$, with $fixed$ and opt representing the cost or time under the fixed-speed and optimized-speed strategies, respectively.

Table 4 demonstrates that the speed optimization strategy yields substantial time savings across all instance sizes. For example, in the 15-customer instance, it achieves a service completion time of 1156.18 seconds—compared to 1588.40 seconds with the fixed 15 m/s strategy—a 27.21% reduction. This suggests that although lower speeds reduce energy consumption and extend drone endurance, excessively slow flights hinder delivery efficiency. In contrast, the fixed 30 m/s strategy results in a completion time of 2682.11 seconds, which the speed optimization reduces by 56.89%. This highlights that while higher speeds shorten individual delivery times, they greatly increase energy consumption, shrinking the drone's service range and forcing greater reliance on less time-efficient truck deliveries for remote customers. From the cost perspective, the speed optimization strategy generally achieves more effective cost control than fixed-speed approaches, despite the objective function not explicitly targeting cost minimization. In most scenarios, it yields lower overall costs with substantial savings; for example, compared to the fixed 30 m/s strategy, cost reductions exceeding 20% are observed across all instances. In the 10-customer case with a fixed speed of 20 m/s, however, the strategy incurs a slightly higher cost, resulting in a marginal negative saving of -0.92%. Nonetheless, this discrepancy is minor and does not outweigh the overall benefits.

To summarize, the results presented in Sections 5.3.1–5.3.2 collectively demonstrate that optimizing drone speed leads to better performance in both cost and service completion time, compared to fixed-speed approaches.

5.4. Impact of customer geographical distribution

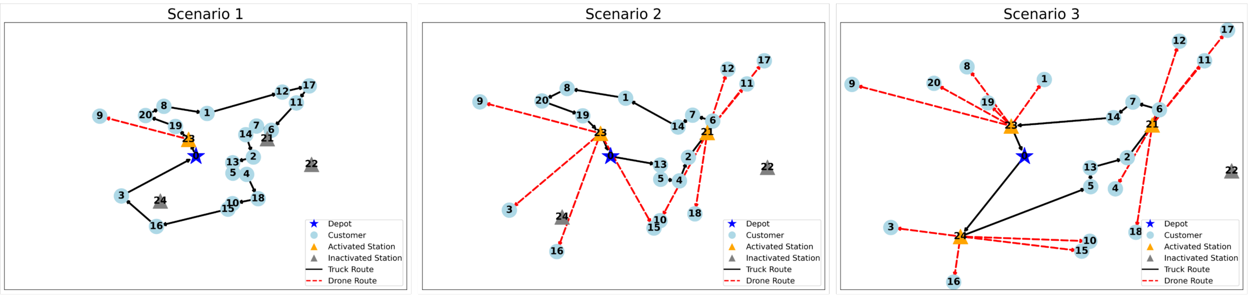
In a delivery system involving both a truck and drones, the distance between customers influences the choice of delivery mode, which in turn affects the system's total cost. To investigate this relationship, we analyze route choice preferences under varying spatial distributions by adjusting the inter-customer distances to simulate different customer densities. Specifically, we consider three scenarios: the original distance setting used in previous experiments; 0.5 times the original distance, representing dense urban areas; and 3 times the original distance, representing sparse rural areas. For convenience, we label these configurations as Scenario 1, Scenario 2, and Scenario 3, respectively.

The average results of instances with 20–60 customers are presented in Table 5. *Activated Stations* denotes the number of activated drone stations. *Drone Tasks* and *Truck Tasks* represent the number of customers assigned to drones and trucks, respectively. *Task Ratio* is calculated as *Drone Tasks*/ *Customers* × 100. *Truck Cost Ratio* denotes the proportion of the truck cost in the total cost. *Normalized Route Length* is obtained by dividing the truck's total route length by the distance factor, ensuring fair comparisons across different spatial scales.

Table 5 Impact of customer geographical distribution

			Drone			Truck		
Customers	Distance Factor	Activated Stations	Drone Tasks	Task Ratio(%)	Avg Radius(m)	Truck Tasks	Truck Cost Ratio(%)	Normalized Route Length(m)
20	0.5	1.0	2.8	14.0	3732.75	17.2	78.9	44748.63
	1.0	1.4	8.8	44.0	6163.73	11.2	57.9	25939.34
	3.0	2.6	9.0	45.0	7861.47	11.0	86.8	42524.03
30	0.5	1.2	2.2	7.3	3365.57	27.8	87.4	64850.44
	1.0	1.2	10.0	33.3	5613.97	20.0	65.7	41173.32
	3.0	3.2	14.0	46.7	7942.42	16.0	83.8	52165.46
40	0.5	0.6	1.2	3.0	2565.44	38.8	94.1	78776.54
	1.0	1.2	9.2	23.0	6626.24	30.8	73.8	56025.80
	3.0	4.0	22.4	56.0	7447.51	17.6	75.3	49131.54
60	0.5	0.4	0.6	1.0	1778.59	59.4	97.4	101094.37
	1.0	1.2	6.0	10.0	6622.78	54.0	87.1	87315.43
	3.0	6.6	33.2	55.3	6870.75	26.8	70.2	56277.66

Table 5 shows that when customers are more clustered, as in Scenario 1, the number of activated stations is minimal, and drone utilization is at its lowest. In this case, the truck serves the majority of customers, accounting for the dominant portion of the total cost. As the distance between customers increases, more stations are activated, a larger share of delivery tasks is handled by drones, and the average coverage radius of drones also expands. Although one might expect the truck route to shorten as more deliveries are assigned to drones, the 20- and 30-customer instances in Scenario 3 actually have longer truck routes than those in Scenario 2. This results from the activation of additional stations—some located in remote areas to exploit drones' coverage advantage—causing truck detours. Figure 7 illustrates this effect by displaying the truck and drone routes for a 20-customer instance. For larger-sized instances with 80–100 customers, Figure 8 shows that as the inter-customer distance increases, the number of drone tasks rises while the truck route length decreases. Across the three distance settings, Scenario 1 consistently produces the longest truck routes.



Activated Stations = **1** Avg Radius = **2013.15**
Drone Tasks = **1** Normalized Route Length = **42347.60**

Activated Stations = **2** Avg Radius = **4187.39**
Drone Tasks = **9** Normalized Route Length = **20792.56**

Activated Stations = **3** Avg Radius = **7964.77**
Drone Tasks = **14** Normalized Route Length = **20926.81**

Figure 7 Illustration of routes under different customer spatial distributions for a 20-customer instance

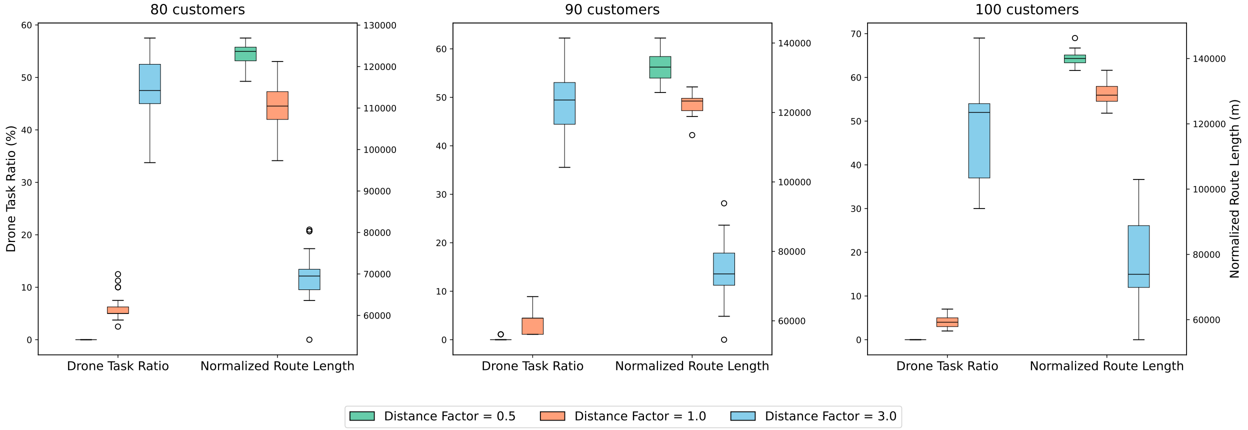


Figure 8 Performance comparison of different distance scenarios for instances with 80–100 customers

These results indicate that when customers are concentrated, the truck achieves high efficiency by visiting multiple customers per route, thereby limiting the relative advantage of point-to-point drone deliveries. In contrast, when customers are more dispersed, drones can efficiently serve remote customers to control the system's overall cost. These observations suggest that the optimal choice of delivery mode is highly dependent on the spatial distribution of customer locations.

6. Conclusions

This paper introduces a novel last-mile delivery mode in which a truck and multiple drones, docked at drone stations, collaborate to deliver packages to customers. Unlike previous studies that typically treat drone speed as a fixed parameter in drone facility location problems, we incorporate drone speeds as decision variables. Furthermore, we adopt a comprehensive energy consumption model that accounts for all phases of drone flight. The objective is to determine the truck route, activation of drone stations, drone-to-customer assignments, and drone speeds, to minimize the overall cost. For the problem, we construct an MILP model and develop a tailored ALNS algorithm featuring problem-specific destroy and repair operators along with local search strategies. Using instances based on real-world geographical data from Istanbul, Turkey, numerical experiments are conducted. The results show that, compared to Gurobi, our ALNS algorithm can efficiently generate high-quality solutions for large-sized instances with 100 customers within 50 seconds. Moreover, compared to fixed-speed strategies, the speed optimization approach shows superior performance regarding total cost and service completion time, along with drone-related indicators such as energy consumption and coverage radius. A sensitivity analysis on customer geographical distribution indicates that the deployment of drone stations and the optimization of drone speeds are particularly beneficial in sparsely populated rural areas.

This research could be extended by incorporating heterogeneous drones with varying battery capacities and payload limits, enabling more flexible drone operations. For example, rural areas may benefit from drones with larger battery capacities to support extended service coverage. In contrast, urban environments may favor lighter, more agile models to enable higher delivery frequencies. Furthermore, integrating customer time window constraints would enhance the model's applicability, particularly in scenarios where delivery timing and service level requirements are critical.

References

Alverhed, Elin, Simon Hellgren, Hanna Isaksson, Lisa Olsson, Hanna Palmqvist, Jonas Flodén. 2024. Autonomous last-mile delivery robots: a literature review. *European Transport Research Review* **16**(1) 4.

Amazon. 2025. Amazon's drones deliver items in 60 minutes or less—here's how we simplified the process. Accessed July 24, 2025, https://www.aboutamazon.com/news/transportation/amazon-drone-deliver-package.

Amorosi, Lavinia, Paolo Dell'Olmo, Justo Puerto, Carlos Valverde. 2025. A time space network model for a truck and multi-drone delivery system with battery recharging and variable speeds. *Omega* 103399.

Aurelia Aerospace. 2024. Aurelia X6 Standard. Accessed July 19, 2025, https://aurelia-aerospace.com/products/aurelia-x6-standard.

Bektaş, Tolga, Gilbert Laporte. 2011. The pollution-routing problem. *Transportation Research Part B: Methodological* **45**(8) 1232–1250.

Bruni, Maria Elena, Sara Khodaparasti, Guido Perboli. 2023. Energy efficient uav-based last-mile delivery: A tactical-operational model with shared depots and non-linear energy consumption. *IEEE Access* **11** 18560–18570.

Campuzano, Giovanni, Eduardo Lalla-Ruiz, Martijn Mes. 2023. The drone-assisted variable speed asymmetric traveling salesman problem. *Computers & Industrial Engineering* **176** 109003.

Chauhan, Darshan, Avinash Unnikrishnan, Miguel Figliozzi. 2019. Maximum coverage capacitated facility location problem with range constrained drones. *Transportation Research Part C: Emerging Technologies* **99** 1–18.

Cheng, Chun, Yossiri Adulyasak, Louis-Martin Rousseau. 2020. Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B: Methodological* **139** 364–387.

Cheng, Chun, Peng Yang, Mingyao Qi, Louis-Martin Rousseau. 2017. Modeling a green inventory routing problem with a heterogeneous fleet. *Transportation Research Part E: Logistics and Transportation Review* **97** 97–112.

ChooseEnergy. 2024. Electricity Rates by State. Accessed July 19, 2025, https://www.chooseenergy.com/electricity-rates-by-state/.

DHL Taiwan Press. 2019. DHL express launches its first regular fully-automated and intelligent urban drone delivery service Accessed August 7, 2025, https://www.dhl.com/tw-en/home/press/press-archive/2019/dhl-express-launches-its-first-regular-fully-automated-and-intelligent-urban-drone-delivery-service.html.

Dodge, M., S. A. MirHassani, F. Hooshmand. 2024. A modelling and solution approach for wind-affected drone-truck routing problem under uncertainty. *Expert Systems with Applications* **257** 124996.

Dorling, Kevin, Jordan Heinrichs, Geoffrey G Messier, Sebastian Magierowski. 2016. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **47**(1) 70–85.

Dukkanci, Okan, James F Campbell, Bahar Y Kara. 2024a. Facility location decisions for drone delivery: A literature review. *European Journal of Operational Research* **316**(2) 397–418.

Dukkanci, Okan, James F Campbell, Bahar Y Kara. 2024b. Facility location decisions for drone delivery with riding: A literature review. *Computers & Operations Research* **167** 106672.

Dukkanci, Okan, Bahar Y Kara, Tolga Bektaş. 2021. Minimizing energy and cost in range-limited drone deliveries with speed optimization. *Transportation Research Part C: Emerging Technologies* **125** 102985.

Dynamic Logistix. 2024. Trucking Freight Rates. Accessed July 19, 2025, https://dynamiclogistix.com/trucking-freight-rates/.

Huang, Hailong, Andrey V Savkin. 2020. A method of optimized deployment of charging stations for drone delivery. *IEEE Transactions on Transportation Electrification* **6**(2) 510–518.

Kaiyuan Securities. 2024. Industry Report on Logistics Drones in Power Equipment Sector. Accessed July 19, 2025, https://data.eastmoney.com/report/zw_industry.jshtml?encodeUrl=M9gDZh+mIjCUzqd6sUgxh51WHtRgB4AosvTEtlEMeA0=.

Kilci, Fatih, Burcin Bozkaya, Bahar Yetis Kara, et al. 2018. Kartal district of istanbul. Online dataset. Available at https://doi.org/10.17632/xhz8m74pp7.1.

Kim, Sungwoo, Ilkyeong Moon. 2018. Traveling salesman problem with a drone station. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **49**(1) 42–52.

Kirschstein, Thomas. 2020. Comparison of energy demands of drone-based and ground-based parcel delivery services. *Transportation Research Part D: Transport and Environment* **78** 102209.

Kloster, Konstantin, Mahdi Moeini, Daniele Vigo, Oliver Wendt. 2023. The multiple traveling salesman problem in presence of drone-and robot-supported packet stations. *European Journal of Operational Research* **305**(2) 630–643.

Liu, Zhilong, Raja Sengupta, Alex Kurzhanskiy. 2017. A power consumption model for multi-rotor small unmanned aircraft systems. *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 310–315.

Logistics Research. 2025. SF Express Drone Operations Data Statistics. Accessed July 20, 2025, https://logresearch.logclub.com/articleInfo/NzQwNjA=.

MAVLink. 2020. MAVLink Developer Guide. Accessed August 7, 2025, https://mavlink.io/en/.

Meituan. 2024. Meituan Drone Delivery Service Progress Report. Accessed July 20, 2025, https://www.meituan.com/news/NN241218063008769.

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

Meng, Shanshan, Xiuping Guo, Dong Li, Guoquan Liu. 2023. The multi-visit drone routing problem for pickup and delivery services. *Transportation Research Part E: Logistics and Transportation Review* **169** 102990.

Meskar, Mahla, Amir Ahmadi-Javid. 2025. Integrated speed optimization and multivisit drone routing: a mathematical programming approach. *International Transactions in Operational Research* .

Murray, Chase C, Amanda G Chu. 2015. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies* **54** 86–109.

Pei, Zhi, Wuhao Pan, Kebiao Weng, Tao Fang. 2024. A branch-and-price-and-cut algorithm for the unmanned aerial vehicle delivery with battery swapping. *International Journal of Production Research* **62**(19) 7030–7055.

Pinto, Roberto, Alexandra Lagorio. 2022. Point-to-point drone-based delivery network design with intermediate charging stations. *Transportation Research Part C: Emerging Technologies* **135** 103506.

Raj, Ritwik, Dowon Lee, Seunghan Lee, Jose Walteros, Chase Murray. 2021. A branch-and-price approach for the parallel drone scheduling vehicle routing problem. *Available at SSRN 3879710* .

Raj, Ritwik, Chase Murray. 2020. The multiple flying sidekicks traveling salesman problem with variable drone speeds. *Transportation Research Part C: Emerging Technologies* **120** 102813.

Ropke, Stefan, David Pisinger. 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* **40**(4) 455–472.

Sarasola, Briseida, Karl F Doerner. 2020. Adaptive large neighborhood search for the vehicle routing problem with synchronization constraints at the delivery location. *Networks* **75**(1) 64–85.

Schermer, Daniel, Mahdi Moeini, Oliver Wendt. 2019. The traveling salesman drone station location problem. *World Congress on Global Optimization*. Springer, 1129–1138.

Shavarani, Seyed Mahdi. 2019. Multi-level facility location-allocation problem for post-disaster humanitarian relief distribution. *Journal of Humanitarian Logistics and Supply Chain Management* **9**(1) 70–81.

Tamke, Felix, Udo Buscher. 2023. The vehicle routing problem with drones and drone speed selection. *Computers & Operations Research* **152** 106112.

The Spoon. 2024. Valqari Launching Single Family Drone Mailboxes Later This Year, Targeting $1500 Price Point. Accessed July 19, 2025, https://thespoon.tech/valqari-launching-single-family-drone-mailboxes-later-this-year-targeting-1500-price-point/.

Urban Air Mobility. 2020. U.S. Chicago start-up launches its drone delivery station. Accessed August 7, 2025, https://www.urbanairmobilitynews.com/uam-infrastructure/u-s-chicago-start-up-launches-its-drone-delivery-station/.

Voigt, Stefan. 2025. A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems. *European Journal of Operational Research* **322**(2) 357–375.

Yang, Ying, Jiaxin Liu, Shuaian Wang. 2025. A bi-objective optimisation model for the drone scheduling problem in island delivery. *International Journal of Production Research* 1–22.

Zeng, Yong, Jie Xu, Rui Zhang. 2019. Energy minimization for wireless communication with rotary-wing uav. *IEEE Transactions on Wireless Communications* **18**(4) 2329–2345.

Zhang, Juan, James F Campbell, Donald C Sweeney II, Andrea C Hupman. 2021. Energy consumption models for delivery drones: A comparison and assessment. *Transportation Research Part D: Transport and Environment* **90** 102668.

A. Parameter setting of ALNS algorithm

The ALNS algorithm is configured as follows. The adaptive mechanism uses a score decay factor of 0.9 and updates operator weights every 25 iterations. Rewards are assigned based on solution quality changes, with positive rewards for improvements and negative for deteriorations. Reward magnitude is scaled by the logarithm of the cost change (coefficient: 0.5), with a minimum threshold of 0.1 to ensure operator diversity; all operators start with a score of 1.0. A phased weighting strategy is applied: in the first half of iterations, weights range from 0.15 to 0.75 to promote exploration, while in the second half, they narrow to 0.20–0.60 to emphasize performance. Simulated annealing starts at a temperature of 200 with a cooling rate of 0.98. The algorithm runs for up to 500 iterations, with a restart triggered after 50 consecutive iterations without improvement. An elite pool of up to five solutions supports restarts. Local search is applied every 20 iterations, and the `randomRemove` operator has a removal rate of 0.25.

B. Time minimization model

To formulate the time minimization model, we introduce an additional parameter and three types of decision variables. Let v_t denote the truck's driving speed. The continuous variable t_i represents the service start time at node $i \in \mathcal{N}$, and T denotes the overall service completion time. The binary variable $o_{di}^{sj} = 1$ if and only if drone $d \in \mathcal{D}$ from station $s \in \mathcal{N}_s$ serves customer $j \in \mathcal{N}_c$ after customer $i \in \mathcal{N}_c$.

The model is constructed as follows:

$$\min T \tag{B.1}$$

$$\text{s.t. } (5)\text{–}(16),$$

$$T \geq t_i \qquad \forall i \in \mathcal{N}_c, \tag{B.2}$$

$$t_j \geq t_i + \frac{c_{ij}}{v_t} - M(1 - x_{ij}) \qquad \forall i \in \mathcal{N}^+, j \in \mathcal{N}', i \neq j, \tag{B.3}$$

$$t_i \geq t_s + \frac{h}{v_v^t} + \frac{h}{v_v^l} + \sum_{r \in \mathcal{R}} \frac{\bar{c}_{si}}{v_r'} w_{di}^{sr} - M(1 - y_{di}^s) \quad \forall i \in \mathcal{N}_c, d \in \mathcal{D}, s \in \mathcal{N}_s, \tag{B.4}$$

$$o_{di}^{sj} \leq y_{di}^s \qquad \forall i, j \in \mathcal{N}_c, d \in \mathcal{D}, s \in \mathcal{N}_s, i \neq j, \tag{B.5}$$

$$o_{di}^{sj} \leq y_{dj}^s \qquad \forall i, j \in \mathcal{N}_c, d \in \mathcal{D}, s \in \mathcal{N}_s, i \neq j, \tag{B.6}$$

$$o_{di}^{sj} + o_{dj}^{si} \geq y_{di}^s + y_{dj}^s - 1 \qquad \forall i, j \in \mathcal{N}_c, d \in \mathcal{D}, s \in \mathcal{N}_s, i \neq j, \tag{B.7}$$

$$t_j \geq t_i + \left(\frac{h}{v_v^t} + \frac{h}{v_v^l} + \sum_{r \in \mathcal{R}} \frac{\bar{c}_{si}}{v_r'} w_{di}^{sr} \right) + \left(\frac{h}{v_v^t} + \frac{h}{v_v^l} + \sum_{r \in \mathcal{R}} \frac{\bar{c}_{sj}}{v_r'} w_{dj}^{sr} \right) - M(1 - o_{di}^{sj})$$
$$\forall i, j \in \mathcal{N}_c, d \in \mathcal{D}, s \in \mathcal{N}_s, i \neq j, \tag{B.8}$$

$$t_i \geq 0 \qquad \forall i \in \mathcal{N}, \tag{B.9}$$

$$o_{di}^{sj} \in \{0,1\} \qquad \forall i, j \in \mathcal{N}_c, d \in \mathcal{D}, s \in \mathcal{N}_s, i \neq j. \tag{B.10}$$

The objective function (B.1) minimizes the system's service completion time. Constraint (B.2) defines it as the latest arrival time across all customers. Constraint (B.3) ensures time consistency for truck operations. Constraint (B.4) specifies service start times for drone-served customers, accounting for takeoff, landing, and flight durations. Constraints (B.5)–(B.6) link ordering variables to drone assignments, while constraint (B.7) enforces unique orderings among customers served by the same drone. Constraint (B.8) ensures temporal feasibility for sequential drone deliveries. Finally, constraints (B.9) and (B.10) define variable domains.