sustainability

MDPI

Article

Arc Routing for Drone Road Inspection with Dynamic Battery Swapping: A Case Study of Heshan City

Wenxue Cai, Qifeng Li and Yimiao Gu *

Department of Electronic Business, South China University of Technology, Guangzhou 510006, China; ctwxcai@scut.edu.cn (W.C.); 202420159292@mail.scut.edu.cn (Q.L.)
* Correspondence: guymcheers@scut.edu.cn

Abstract

Electric drones have been widely applied in highway infrastructure inspection as a low-carbon maintenance solution, but the limited battery endurance severely restricts large-scale continuous operations. While ground vehicles as mobile battery swapping platforms can alleviate this issue, they bring challenges including multi-mode flight selection for drones and complex spatiotemporal coupling constraints between drones and vehicles. This study focuses on the Drone Road Inspection Arc Routing Problem (ARP) with dynamic battery swapping. With the goal of minimizing total inspection time, a mixed-integer programming model integrating a three-layer network structure and virtual node extension is established. Through rational selection of drone flight modes and coordinated planning of vehicle–drone paths, the non-inspection flight time of drones is effectively reduced, which also helps cut unnecessary energy consumption. To address the challenge of solving large-scale instances, an Adaptive Large Neighborhood Search (ALNS) Algorithm with battery endurance-oriented operators is proposed. Numerical experiments show that the algorithm exhibits great performance in both solution efficiency and quality. The case study of Heshan City verifies that the proposed hybrid flight strategy delivers notable benefits in both path optimization and energy consumption reduction, and the sensitivity analysis clarifies the optimization directions of key parameters, providing technical reference for sustainable road maintenance.

Keywords: road inspection; vehicle–drone arc routing problem; adaptive large neighborhood search

1. Introduction

Against the backdrop of global carbon neutrality goals and sustainable transportation infrastructure development, promoting the low-carbon transformation of highway maintenance has become a core demand for road management authorities. Traditional fuel-powered inspection vehicles suffer from high carbon emissions and insufficient coverage of blind areas, which can no longer meet the requirements of green and sustainable road operation and maintenance.

In recent years, drone inspection has been increasingly adopted as an effective solution for highway infrastructure inspection. With low-altitude flight capability and access to hard-to-reach areas, drones are deployed to detect defects in the blind spots of traditional vehicle inspection, such as road slopes and bridge-tunnel transition sections. On the one hand, replacing fuel-powered inspection vehicles with electric drones directly reduces carbon emissions from maintenance operations; on the other hand, efficient and accurate

Check for updates

Academic Editor: Carlos Llopis-Albert

Received: 5 August 2026
Revised: 8 October 2026
Accepted: 13 October 2026
Published: 25 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

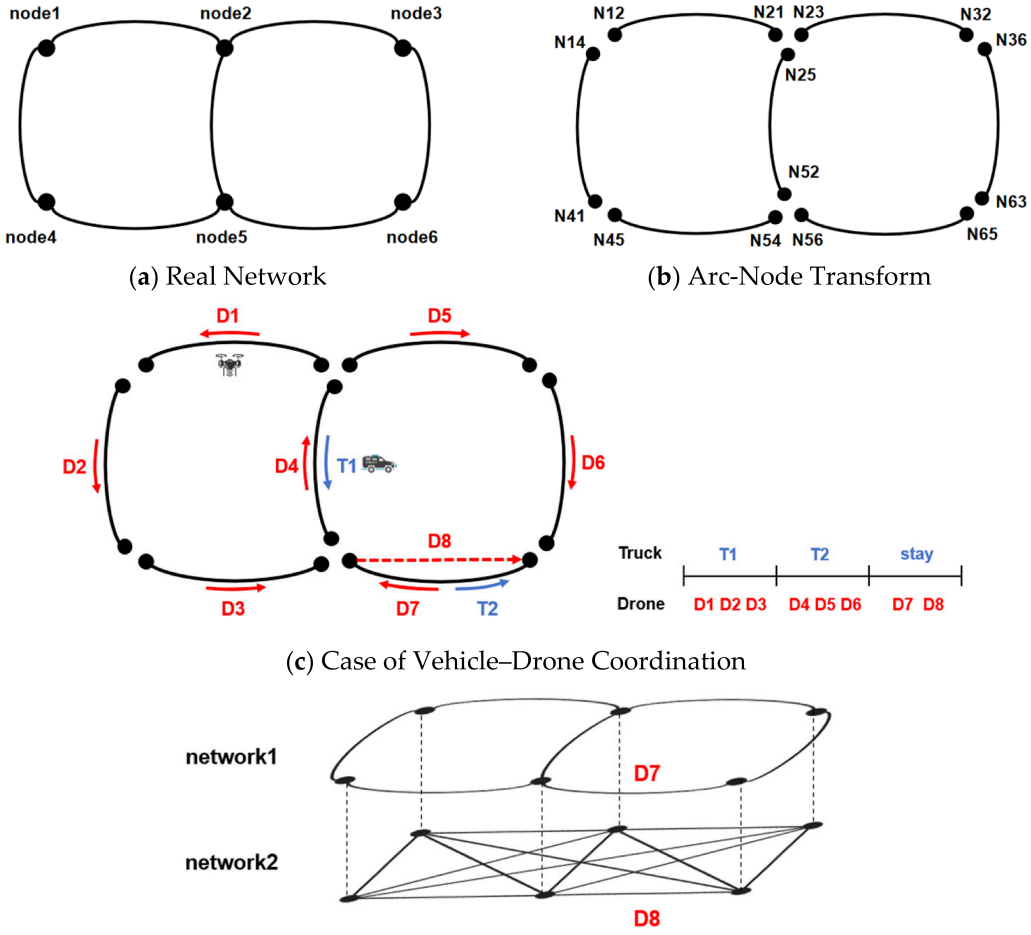
defect detection helps extend the service life of road facilities and reduce resource consumption from large-scale overhauls, which aligns with the goal of sustainable infrastructure development. However, the critical limitation of drones is insufficient endurance, which fails to support continuous inspection over large-scale road networks. Existing battery replenishment strategies for drones mainly include fixed replenishment [1] and vehicle-based battery replenishment [2]. The fixed charging station mode is severely restricted by dynamic inspection demands, difficult site selection, and high operation and maintenance costs, which greatly limits its practical applicability. Thus, using vehicles as mobile replenishment carriers to provide endurance support for inspection drones has become the mainstream solution for large-scale road network inspection.

In this mode, drones are responsible for inspecting vehicle-inaccessible areas such as road slopes. Vehicles provide takeoff, landing, and battery replacement services for drones to ensure their long-term operation. This study addresses the core problem: under the premise of ensuring effective traversal of all inspection demand arcs in the road network, how to optimize the full-process flight paths of drones and simultaneously match the travel and battery-swapping stop strategies of vehicles, so as to minimize the total inspection time of drones. In this problem, the total inspection time of drones can be divided into two parts: inspection flight time and non-inspection transfer flight time. The inspection flight time is determined by the mileage of predefined demand arcs and is relatively fixed. In contrast, the non-inspection flight time has significant optimization potential. Therefore, reducing the non-inspection flight time of drones is the core optimization focus of this study, which also helps to further cut unnecessary energy consumption and improve the low-carbon level of inspection operations.

Reasonable switching of drone flight modes is an effective way to reduce non-inspection flight time. On the one hand, drones can ignore the restrictions of road travel direction, and the shortest path distance between nodes is relatively shorter than that of vehicles. Wei et al. [3] accounted for this characteristic and defined separate shortest path distance matrices for drones and vehicles. On the other hand, the core advantage of drones in road inspection scenarios is that their flight path selection is not bound by the road network; they can fly in a straight line between any two nodes in the road network. This means that drones fly along the actual road curve during the inspection phase and fly in a straight line during the non-inspection phase. Wu et al. [4], Campbell, J. F. et al. [5], and Liu, X. F. et al. [6] all considered the combination of drones' straight-line flight and curved-line flight modes. Through the arc-node transform, they defined that when an arc is a demand arc, its distance is the actual path distance; when an arc is a non-demand arc, its distance is the shortest path distance. This definition effectively resolves the inconsistency of time-sequence variables when passing through the same node repeatedly in the Rural Postman Problem (RPP). However, this definition ignores the vehicle–drone path coupling relationship in the vehicle–drone inspection process: when the vehicle cannot reach the rendezvous node in time due to its slower speed than the drone, the drone may pass through the same arc segment multiple times to meet the vehicle.

As shown in Figure 1, for network structure (a), assume that every arc in the network is a demand arc, and the drone inspection path is (node 2, node 1, node 4, node 5, node 2, node 3, node 6, node 5). Converting network structure (a) to network structure (b) via the arc-node transform can solve the repeated traversal problem of node 2 and node 5. Based on real-world conditions, we assume that the vehicle speed is lower than the drone flight speed, and the drone does not hover at nodes to wait for the vehicle. As shown in case (c), the vehicle and drone depart simultaneously from node 2, pass through rendezvous nodes node 5 and node 6 successively, and finally end the inspection at node 6. In this case, the drone needs to pass through the same arc segment (N56, N65) twice, and fly straight from

node N56 to node N65 to rendezvous with the vehicle. The distance matrix definition method of Liu, X.F. et al. [6] may not fully capture such vehicle–drone coupling scenarios. The core reason is that this method integrates multiple flight modes of drones into a single road network structure, which exhibits limited adaptability when spatiotemporal constraints are tightened. Therefore, this paper adopts the virtual node replication method to extend the road network structure [7]. As shown in (d), the actual road network (a) is converted into the curved flight network 1 and the direct flight network 2. The model in this paper comprehensively considers the different flight modes of drones in the inspection and non-inspection phases, as well as the differences between the drone and vehicle travel networks. It defines the drone inspection network, drone direct flight network, and vehicle travel network, and realizes vehicle–drone path coupling according to inspection task constraints and battery endurance constraints. This reduces unnecessary non-inspection flight distances of drones caused by real-world constraints such as vehicle mobility, and optimizes the total inspection time on this basis.



(**a**) Real Network (**b**) Arc-Node Transform

(**c**) Case of Vehicle–Drone Coordination

(**d**) Road Network Expansion Structure

Figure 1. Schematic diagram of the topology.

According to different task objectives, existing research on the drone routing problem can be divided into node-based routing problems (VRP) and arc-based routing problems (ARP). VRP research is widely applied in power inspection [8], customer delivery [9], and emergency logistics [10]. ARP research mainly focuses on application scenarios including traffic patrol and area coverage. Xu et al. [11] designed the RVND-SA algorithm to solve the coordinated monitoring problem of target road segments by ground vehicles and multiple drones in urban traffic patrol tasks. Momeni et al. [12] adopted the Benders decomposition

algorithm to address the coordinated patrol problem of trucks and drones in early forest fire detection. However, ARP research focusing on the road inspection scenario remains relatively scarce in the field of vehicle–drone coordinated routing.

In terms of solution methods for drone routing problems, scholars have proposed various solutions in recent years. From the perspective of solution strategies, these methods can be categorized into two main classes: exact algorithms based on mathematical models and heuristic algorithms based on rule design. For exact algorithms, traditional methods such as DP [13], B&B [14], and BP [15] are widely used, while emerging algorithms including CG [16] and BD [17] have also been extensively applied. Research on heuristic algorithms has been flourishing, with applications of multi-stage heuristics [18–24], SA [25], ACO [7], GRASP [2], VNS [26], MA [27,28] and RL [29].

The Adaptive Large Neighborhood Search (ALNS) algorithm has been widely applied to vehicle–drone coordinated routing problems in recent years. In terms of the global design of ALNS, Chen et al. [30] generated diverse initial solutions through multiple heuristic strategies, providing abundant search starting points for ALNS. Laporte et al. [31] decomposed routing costs into deterministic route costs and expected remedial costs, and propagated dynamic demands to subsequent road segments via probability functions. Mulumba et al. [32] introduced a forgetting factor into the ALNS algorithm to intelligently adjust the weight iteration of various operators, preventing historical performance from dominating excessively. Liu et al. [33] integrated battery scheduling decisions with routing decisions such as truck routes and drone takeoff/landing points, ensuring the continuity of drone missions under limited battery resources. In terms of detailed design of ALNS, the time window similarity-based destruction strategies proposed by Zhang et al. [34] and Meng et al. [35], as well as the robust feasibility verification mechanism developed by Li et al. [36], have achieved excellent search optimization effects. To solve large-scale instances of real road networks, this paper designs a solution algorithm based on ALNS. We consider the battery endurance utilization rate of drones in each inspection flight during operator design to maximize battery utilization and reduce unnecessary charging trips.

The relevant literature is summarized in Table 1.

Table 1. Summary of related studies.

Reference	Problem Type	Routing Unit	Charging Strategy [1]	Depot Position	Multiple Flight Modes [2]	Algorithm
Cai et al. [1]	LRP	Arc	Fixed	Fixed	No	LBBD
Liu et al. [33]	VRP	Node	Fixed	Fixed	No	ALNS
Chen et al. [30]	VRP	Node	Mobile	Fixed	No	ALNS
Wu et al. [4]	ARP	Arc	Mobile	Fixed	No	ALNS
Xu et al. [11]	ARP	Arc	Mobile	Fixed	No	RVND-SA
Momeni et al. [12]	ARP	Arc	Mobile	Fixed	No	BD
Liu, X. et al. [6]	ARP	Arc	Mobile	Fixed	Yes	ALNS
This study	ARP	Arc	Mobile	Flexible	Yes	ALNS

[1] Charging Strategy: Fixed indicates that the drone is recharged at fixed charging stations; mobile indicates that the drone is recharged at mobile charging stations (typically vehicles). [2] Multiple flight modes: indicates whether the study simultaneously defines two flight modes for the drone: flying along road curves and flying straight between nodes.

The main contributions of this study are summarized as follows:

1. A three-layer network framework is constructed to address the vehicle–drone coordinated arc routing problem with dynamic battery swapping, which fully considers the differentiated travel characteristics of drones and vehicles and improves the model's applicability in real road inspection scenarios.

2. Based on the vehicle–drone coordinated road inspection scenario, an Adaptive Large Neighborhood Search Algorithm is designed, and a case study based on the real road network of Heshan City is conducted to verify the optimization effect of the direct-flight strategy, providing a practical reference for the on-site deployment of drone inspection systems.
3. The routing optimization scheme proposed in this paper can effectively reduce the energy consumption of inspection operations, which provides decision support for the low-carbon transformation of road maintenance and the sustainable development of transportation infrastructure.

2. Problem Formulation

2.1. Model Assumptions

The proposed model is based on the following assumptions:

1. Drones are responsible for drone road inspection tasks, and vehicles provide takeoff and landing platforms as well as battery swapping services for drones;
2. The model only allows vehicles and drones to rendezvous and swap batteries at road network nodes, and prohibits mid-road rendezvous for battery swapping. In addition, vehicles must arrive at rendezvous nodes earlier than drones, and drones are not allowed to hover and wait for vehicles at nodes;
3. The length of a single inspection demand arc does not exceed the maximum battery endurance distance of the drone.

2.2. Problem Statement

The vehicle–drone road inspection mode is shown in Figure 2. In the road inspection model, the drone and vehicle start simultaneously from a node in the road network. The drone takes off from the vehicle with a fully charged battery to begin inspecting road slopes, while the vehicle travels to the next rendezvous node. During the inspection phase, the drone performs tasks along the actual road network routes, flying in a curved path. When the endpoint of the next demand arc is not directly connected to that of the current demand arc, the drone can break through road network constraints and fly in a straight line between nodes to reach the nearest demand arc and continue the inspection. During the battery swapping phase, since the vehicle speed is lower than the drone flight speed and the vehicle is restricted by road direction, it cannot always arrive at the rendezvous node earlier than the drone. Therefore, when the drone's battery endurance is nearly exhausted, it decides whether to fly straight to a node close to the vehicle for battery swapping based on the vehicle's reachability, as illustrated in Figure 2. For the first battery swap, the vehicle arrives on time, and the two rendezvous at chargeNode1. For the second battery swap, the vehicle cannot reach the rendezvous point in time, so the drone flies straight along D7 to chargeNode2 to meet the vehicle. After battery swapping at the rendezvous point, the drone takes off again to continue inspecting the remaining demand arcs. This process repeats until all demand arcs are covered, and the inspection is completed when the drone lands back on the vehicle for retrieval.

Based on the example in Figure 2, Figure 3 presents the vehicle travel network (a), drone inspection network (b), and drone direct flight network (c). The core of the coordination in this paper lies in systematically integrating these three interrelated arc sets through a mathematical model to construct a unified vehicle–drone road inspection routing model. This model needs to address two key problems: first, how to optimize the drone's flight mode switching strategies under the premise of satisfying all road inspection requirements; second, how to realize the spatiotemporal coordination constraints between vehicles and drones in terms of battery endurance through reasonable rendezvous node planning.

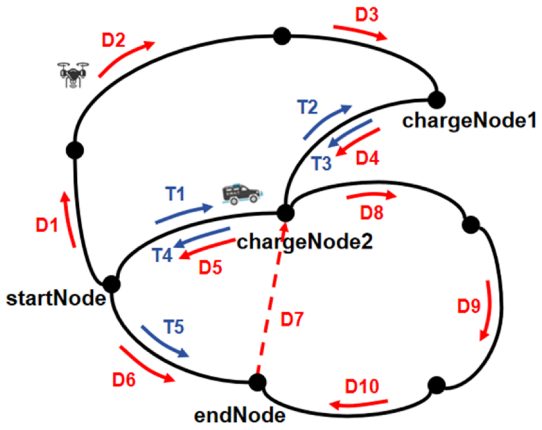


Figure 2. Case of vehicle–drone road inspection.

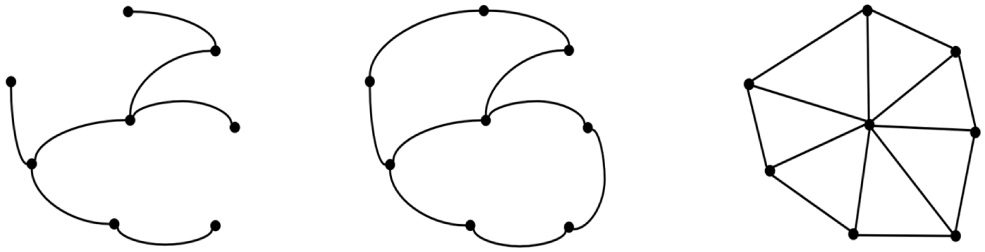


Figure 3. Arc set of vehicle and drone. (**a**) shows the vehicle travel network. (**b**) shows the drone inspection network, while (**c**) shows the drone direct flight network.

2.3. Model Formulation

The notations and variable definitions of the proposed Vehicle–Drone Road Inspection Routing Model are listed in Table 2. The road inspection task is formulated on the graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ denotes the set of all road nodes in the actual road network, and n is the total number of road nodes. Specifically, E_1 denotes the set of arcs traversable by vehicles in the actual road network, and E_2 denotes the set of arcs traversable by drones.

In the ARP, a drone may visit the same node multiple times to traverse all demand arcs. If cumulative flight time is defined on physical nodes, a node visited twice would be associated with two arrival times, which causes logical inconsistencies in the battery-endurance constraints. To accommodate repeated traversal without artificially restricting the number of visits, we generalize the virtual-node technique of Jia et al. [7] and let the replica count of each node be determined by the demand network. Let $\overline{E}_2 = \{\{u, v\} \subseteq V : (u, v) \in E_2 \text{ or } (v, u) \in E_2\}$ be the set of demand segments underlying E_2, where the two arcs of a segment are identified, and let $\delta(v) = \left|\{u \in V : \{u, v\} \in \overline{E}_2\}\right|$ denote the number of demand segments incident to node v. Since every time the route "meets" node v accompanies the traversal of a distinct demand segment incident to v, the drone visits v at most $\delta(v)$ times; it may additionally start, end, or undergo battery exchange at v. Each node v is therefore assigned $r_v = \delta(v) + 1$ replicas $v^{(1)}, \ldots, v^{(r_v)}$, and the total node set is defined as $V_{total} = \bigcup_{v \in V} \left\{v^{(1)}, \ldots, v^{(r_v)}\right\}$. Let $\pi: V_{total} \to V$ return the physical node of a replica. For a demand segment between k and l, traversing from a replica of k to a replica of l reproduces the traversal of that segment. Accordingly, to simplify the formulation of constraints, the extended arc sets are defined as $E_1^{ext} = \{(i, j) \in V_{total} \times V_{total} : (\pi(i), \pi(j)) \in E_1\}$ and $E_2^{ext} = \{(i, j) \in V_{total} \times V_{total} : (\pi(i), \pi(j)) \in E_2\}$. The model requires that all demand arcs are traversed at least once. Traversing the segment (k,l) in the physical network is equivalent to traversing any arc (i,j) in E_2^{ext} whose endpoints i and j are replicas of k and l, respectively.

Table 2. Mathematical notation.

Symbol	Definition
Sets	
V	Set of nodes in the road network, $V = \{1,2,\ldots,n\}$
V_{total}	Total node set, $V_{total} = \bigcup_{v \in V} \left\{ v^{(1)}, \ldots, v^{(r_v)} \right\}$
$\delta(v)$	The number of demand arcs associated with node v
E_1	Set of vehicle-traversable arcs in the road network
E_2	Set of drone-traversable arcs in the road network
E_2'	Set of drone direct-flight arcs based on node set V
E_1^{ext}	Extended set of vehicle-traversable arcs based on V_{total}
E_2^{ext}	Extended set of drone-traversable arcs based on V_{total}
E_2^{str}	Extended set of drone direct-flight arcs based on V_{total}
Parameters	
d_{ij}^{car}	Distance from node i to node j for the vehicle
d_{ij}^{drone}	Distance from node i to node j for the drone along the road
d'^{drone}_{ij}	Direct flight distance from node i to node j for the drone
v_{car}	Constant speed of the vehicle
v_{drone}	Constant speed of the drone
t_{ij}^{car}	Travel time from node i to node j for the vehicle
t_{ij}^{drone}	Travel time from node i to node j for the drone along the road
t'^{drone}_{ij}	Direct flight time from node i to node j for the drone
t_c	Operation time for battery replacement of the drone
e_{max}	Maximum battery endurance of the drone
Variables	
x_{ij}	1 if the drone flies from node i to node j along the road, 0 otherwise
x'_{ij}	1 if the drone flies directly from node i to node j, 0 otherwise
y_{ij}	1 if the vehicle travels from node i to node j, 0 otherwise
C_{ij}	1 if the drone swaps battery at node j and the previous battery swapping node is i, 0 otherwise
S_i	1 if the drone and vehicle start the inspection task at node i, 0 otherwise
F_i	1 if the drone and vehicle end the inspection task at node i, 0 otherwise
t_i^{drone}	Arrival time of the drone at node i
t_i^{car}	Arrival time of the vehicle at node i

During the battery swapping phase, drones can choose to fly straight to rendezvous nodes for battery replacement. Therefore, we introduce the drone direct flight arc set E_2' based on the node set V. As shown in Figure 4, direct flight routes only exist between nodes within the drone's battery endurance limit. Correspondingly, analogous to E_2^{ext}, we introduce the direct flight extended arc set E_2^{str} based on the total node set V_{total} as the extension of E_2'.

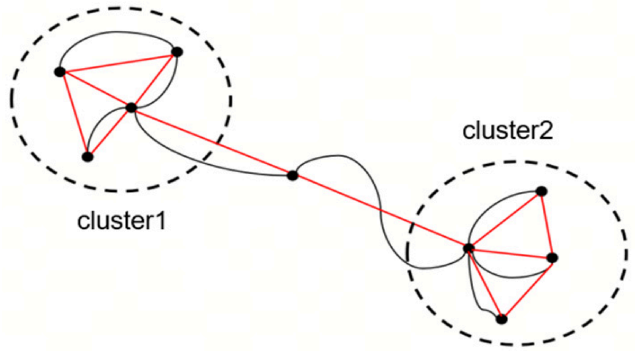


Figure 4. Schematic diagram of the drone direct flight arc set.

The distance matrices between any two nodes in the graph for vehicles and drones are defined as follows.

$$d_{ij}^{drone} = \begin{cases} d_{act}(i,\ j) & i,j \in V_{total}\ and\ (i,j) \in E_2^{ext} \\ M & i,j \in V_{total}\ and\ (i,j) \notin E_2^{ext} \end{cases}$$

$$d_{ij}^{car} = \begin{cases} d_{SP}(i,\ j) & i,j \in V_{total}\ and\ (i,j) \in E_1^{ext} \\ M & i,j \in V_{total}\ and\ (i,j) \notin E_1^{ext} \end{cases}$$

$$d'^{drone}_{ij} = \begin{cases} d_{Euc}(i,\ j) & i,j \in V_{total}\ and\ (i,j) \in E_2^{str} \\ M & i,j \in V_{total}\ and\ (i,j) \notin E_2^{str} \end{cases}$$

The travel distances of vehicles and drones in the road network extended sets E_1^{ext} and E_2^{ext} are denoted by d_{ij}^{car} and d_{ij}^{drone}, where $d_{act}(i, j)$ represents the actual path length of arc (i, j), and $d_{SP}(i, j)$ represents the shortest path distance between node i and node j, which is computed using the Floyd–Warshall algorithm based on the adjacency matrix of the actual road network. The travel distance of drones in the direct flight extended arc set E_2^{str} is denoted by d'^{drone}_{ij}, where $d_{Euc}(i, j)$ represents the Euclidean distance between node i and node j. M denotes a sufficiently large positive constant. Correspondingly, the travel time between any two nodes for vehicles and drones is given by: $t_{ij}^{car} = d_{ij}^{car}/v_{car}$, $t_{ij}^{drone} = d_{ij}^{drone}/v_{drone}$ and $t'^{drone}_{ij} = d'^{drone}_{ij}/v_{drone}$ where v_{car} and v_{drone} are the constant travel speed of the vehicle and the constant flight speed of the drone.

The operation time for replacing the drone's battery is also a key factor affecting the path results of the model solution and the frequency of battery replacement for the drone. Therefore, we introduce the operation time for battery replacement of the drone as t_c, which includes the take-off and landing time of the unmanned aerial vehicle at the vehicle, the battery replacement operation time, and the safety inspection time. We introduce C_{ij} to represent the two adjacent charging nodes as i and j. By combining the endurance constraint, the paths of the drone and the vehicle are coupled. This further leads to the conclusion that the time of battery replacements in the drone inspection path is $\sum C_{ij} - 1$.

The location of the start and end nodes is also a key factor affecting the total inspection time. The problem studied in this paper is essentially a variant of the RPP [5] with vehicle-based battery replacement constraints, whose core is to achieve full coverage traversal of all demand arcs. When the inspection road network does not satisfy the Eulerian circuit condition, the difference in in-degree and out-degree of different start and end nodes will directly determine the unnecessary detour mileage in the route, and thus have a significant impact on the total inspection time. Most existing studies set the start and end points of vehicles and drones as fixed nodes, as demonstrated by the start and end point settings in Yin et al. [17] and Mulumba et al. [19]. Similarly, Momeni et al. [12] used S and F to represent the start point and end recovery point, respectively, which also essentially fixes the start and end points. Different from previous studies, the proposed model does not predetermine the start and end points of vehicle–drone inspection. Instead, binary variables S_i and F_i are used to indicate whether node i is the inspection start point or end point. This flexible setting further eliminates unnecessary deadhead miles, which is conducive to reducing energy consumption in inspection operations. The definitions of other notations in this paper are listed as follows.

The proposed vehicle–drone road inspection routing model is formulated as follows.

Objective Function: The objective of this model is to minimize the total inspection time of the drone, subject to the satisfaction of all drone inspection requirements and the vehicle–drone battery endurance coupling constraints. Constraint (1) represents the total

inspection time, which includes the total flight time of the drone and the operation time for replacing the battery of the drone.

$$\min \sum_{\substack{i,\, j \in V_{total} \\ i \neq j}} \left(x_{ij} t_{ij}^{drone} + x'_{ij} t'^{drone}_{ij} + t_c C_{ij} \right) \tag{1}$$

Inspection Task: Constraint (2) ensures that all demand arcs in the road network are traversed at least once by the drone. Each arc in the drone arc set E_2 corresponds to multiple copies in the extended arc set E_2^{ext}, and all copies represent the same physical arc. So, one valid coverage of any of these copies is required.

$$\sum_{i\,:\pi(i)=k} \sum_{j\,:\pi(j)=l} \left(x_{ij} + x_{ji} \right) = 1 \; \forall \; (k,l) \in E_2 \tag{2}$$

Start and End Point Constraints: Constraints (3) to (5) define the constraints on the number of start and end points for vehicle–drone coordinated road inspection. Specifically, Constraint (3) specifies that there is exactly one start point for the inspection, Constraint (4) specifies that there is exactly one end point, and Constraint (5) states that no node within the defined range of V_{total} can serve as both the start point and the end point simultaneously.

$$\sum_{i \in V_{total}} S_i = 1 \tag{3}$$

$$\sum_{i \in V_{total}} F_i = 1 \tag{4}$$

$$S_i + F_i \leq 1 \quad \forall \, i \in V_{total} \tag{5}$$

Flow Balance: Constraints (6) to (8) impose constraints on the in-degree, out-degree, and flow balance of each node for the drone flight sets E_2^{ext} and E_2^{str}. Constraint (6) is the start and end point constraint for drones. Constraints (7) and (8) specify that any node can be traversed by the drone at most once via any flight mode. Constraints (9) to (11) impose constraints on the in-degree, out-degree, and flow balance of each node for the vehicle travel set E_1^{ext}, which follow the same logic as Constraints (6) to (8).

$$\left(\sum_{j:(i,j) \in E_2^{ext}} x_{ij} + \sum_{j:(i,j) \in E_2^{str}} x'_{ij} \right) - \left(\sum_{j:(j,i) \in E_2^{ext}} x_{ji} + \sum_{j:(j,i) \in E_2^{str}} x'_{ji} \right) = S_i - F_i \quad \forall i \in V_{total} \tag{6}$$

$$\sum_{j:(i,j) \in E_2^{ext}} x_{ij} + \sum_{j:(i,j) \in E_2^{str}} x'_{ij} \leq 1 \quad \forall i \in V_{total} \tag{7}$$

$$\sum_{j:(j,i) \in E_2^{ext}} x_{ji} + \sum_{j:(j,i) \in E_2^{str}} x'_{ji} \leq 1 \quad \forall i \in V_{total} \tag{8}$$

$$\sum_{j:(i,j) \in E_1^{ext}} y_{ij} - \sum_{j:(j,i) \in E_1^{ext}} y_{ji} = S_i - F_i \quad \forall i \in V_{total} \tag{9}$$

$$\sum_{j:(i,j) \in E_1^{ext}} y_{ij} \leq 1 \quad \forall i \in V_{total} \tag{10}$$

$$\sum_{j:(j,i) \in E_1^{ext}} y_{ji} \leq 1 \quad \forall i \in V_{total} \tag{11}$$

Cumulative Time: Constraints (12) to (15) impose constraints on the cumulative arrival time of vehicles and drones. Constraints (12) and (13) ensure that the arrival time of the drone and vehicle at node j is greater than or equal to the arrival time at the previous node i

plus the travel time of the current arc segment, thus guaranteeing the logical consistency of cumulative time. Constraints (14) and (15) ensure that the cumulative arrival time is 0 when a node is the start point. Meanwhile, these constraints also serve as subtour elimination constraints to prevent the formation of subtours.

$$t_j^{drone} - t_i^{drone} \geq x_{ij} t_{ij}^{drone} + x'_{ij} t'^{drone}_{ij} - \left(1 - x_{ij} - x'_{ij}\right) M \quad \forall i,\ j \in V_{total} \tag{12}$$

$$t_j^{car} - t_i^{car} \geq y_{ij} t_{ij}^{car} - \left(1 - y_{ij}\right) M \quad \forall\ (i,\ j) \in E_1^{ext} \tag{13}$$

$$t_i^{drone} \leq M(1 - S_i) \quad \forall i \in V_{total} \tag{14}$$

$$t_i^{car} \leq M(1 - S_i) \quad \forall i \in V_{total} \tag{15}$$

Battery Endurance: Constraint (16) ensures that the cumulative flight time of the drone between two consecutive battery swapping nodes is less than the maximum battery endurance e_{max}. Constraint (17) limits the travel time of the vehicle between two consecutive battery swapping nodes to be less than the flight time of the drone, which satisfies the fundamental assumption that vehicles must arrive at battery swapping nodes earlier than drones. Constraints (18) and (19) connect the adjacent battery swapping nodes of the drone and vehicle in series to form a sequential relationship.

$$t_j^{drone} - t_i^{drone} \leq e_{max} + \left(1 - C_{ij}\right) M \quad \forall i,\ j \in V_{total} \tag{16}$$

$$t_j^{drone} - t_i^{drone} \geq t_j^{car} - t_i^{car} - \left(1 - C_{ij}\right) M \quad \forall i,\ j \in V_{total} \tag{17}$$

$$t_j^{drone} - t_i^{drone} \geq -\left(1 - C_{ij}\right) M \quad \forall i,\ j \in V_{total} \tag{18}$$

$$t_j^{car} - t_i^{car} \geq -\left(1 - C_{ij}\right) M \quad \forall i,\ j \in V_{total} \tag{19}$$

Battery Swapping Constraints: Constraints (20) to (23) establish the correlation between battery swapping nodes and the routes of drones and vehicles. Constraints (24) to (26) are the flow balance constraints for battery swapping variables. Together with Constraints (18) and (19), they jointly constrain the coupling relationship between the vehicle and the drone's battery swapping chain, ensuring the integrity of the route.

$$\sum_{k:(i,k) \in E_2^{ext}} x_{ik} + \sum_{k:(i,k) \in E_2^{str}} x'_{ik} \geq C_{ij} \quad \forall i,\ j \in V_{total} \tag{20}$$

$$\sum_{k:(k,j) \in E_2^{ext}} x_{kj} + \sum_{k:(k,j) \in E_2^{str}} x'_{kj} \geq C_{ij} \quad \forall i,\ j \in V_{total} \tag{21}$$

$$\sum_{k:(i,k) \in E_1^{ext}} y_{ik} \geq C_{ij} \quad \forall i,\ j \in V_{total} \tag{22}$$

$$\sum_{k:(k,j) \in E_1^{ext}} y_{kj} \geq C_{ij} \quad \forall i,\ j \in V_{total} \tag{23}$$

$$\sum_{j \in V_{total}} C_{ij} \leq 1 \quad \forall i \in V_{total} \tag{24}$$

$$\sum_{j \in V_{total}} C_{ji} \leq 1 \quad \forall i \in V_{total} \tag{25}$$

$$\sum_{j \in V_{total} \setminus \{i\}} C_{ij} - \sum_{j \in V_{total}\ \{\mathrm{i}\}} C_{ji} = S_i - F_i \quad \forall i \in V_{total} \tag{26}$$

Variable Type: Constraints (27) to (29) specify the value ranges of all variables.

$$x_{ij},\ x'_{ij},\ y_{ij} \in \{0,\ 1\} \quad \forall i \in V_{total},\ j \in V_{total} \tag{27}$$

$$C_{ij} \in \{0, 1\} \qquad \forall\, i, j \in V_{total}, i \neq j \tag{28}$$

$$S_i, F_i \in \{0, 1\} \qquad \forall i \in V_{total} \tag{29}$$

3. Adaptive Large Neighborhood Search

The proposed vehicle–drone road inspection routing model can be regarded as a variant of the Rural Postman Problem. The RPP has been proven to be an NP-hard problem by Lenstra and Kan [37], and thus the problem studied in this paper is also NP-hard. When the number of roads to be inspected by drones increases to a certain extent, the model cannot obtain an exact solution in polynomial time. To solve larger-scale road inspection routing problems, we develop an Adaptive Large Neighborhood Search algorithm. The ALNS algorithm was first applied to solve the capacitated ARP by Laporte et al. [31], and has been widely extended in subsequent research. The core principle of ALNS lies in its destroy-and-repair framework, and its main process is shown in Algorithm 1.

Algorithm 1: Pseudocode for ALNS

Input: E, DS, RS, N_{max}, C_{max}, T_0, T_{min}
Output: R_{best}, W_{best}

Generate initial total route R_0 from E via greedy principle (Section 4.1)
$R_{current} \leftarrow R_0$, $R_{best} \leftarrow R_{current}$
Calculate initial total route cost W_0 based on R_0 (Section 4.2)
$W_{current} \leftarrow W_0$, $W_{best} \leftarrow W_{current}$
Initialize destroy operators probability P_d^0 and repair operators probability P_r^0
$E \leftarrow 1$, $C \leftarrow 1$, $T \leftarrow T_0$
while $(N < N_{max})$ and $(C < C_{max})$ and $(T > T_{min})$ **do**
    Select destroy operator $d \in DS$ with probability P_d^N
    Select repair operator $r \in RS$ with probability P_r^N
    Apply d then r to $R_{current}$ to obtain new route R_{new}
    Calculate new total cost W_{new} based on R_{new} (Section 4.2)
    if $W_{new} < W_{current}$ or $\varepsilon < e^{-(W_{new}-W_{current})/T}$ **then**
        $R_{best} \leftarrow R_{new}$, $W_{current} \leftarrow W_{new}$
        if $W_{new} < W_{best}$ **then**
            $R_{best} \leftarrow R_{new}$, $W_{best} \leftarrow W_{new}$, $C \leftarrow 0$
    else
        $C \leftarrow C + 1$
    Update operators probability P_d^N and P_r^N
    $T \leftarrow hT$
    $N \leftarrow N + 1$

At the initial stage of the algorithm, taking the demand arc set E as the inspection target, an initial total route is generated according to the greedy principle (detailed in Section 3.1). In each iteration, operators are selected probabilistically from the destroy operator set DS and the repair operator set RS, and a portion of demand arcs are removed and reinserted to generate a new solution. The route costs $W_{current}$ and W_{new} are calculated using the method described in Section 3.2. The new solution is accepted based on the Metropolis criterion. Specifically, the new route is accepted if its cost is lower than the current route cost; otherwise, it is accepted with a probability of $e^{-(W_{new}-W_{current})/T}$, where

T denotes the current temperature of the simulated annealing process, which is gradually cooled at a rate h in each iteration.

The algorithm adopts three termination criteria: the maximum number of iterations, the maximum number of consecutive iterations without improvement, and the simulated annealing temperature. A standard adaptive weight adjustment mechanism is employed: operator scores are updated based on solution quality, and the operator weights are calculated as $\omega_{i,j+1} = (1 - \alpha)\omega_{i,j} + \alpha \times avg_score$, where $\omega_{i,j}$ denotes the weight of operator i in the j-th iteration, avg_score represents the average score of operator i within the current update window, and α is the decay rate used to balance the historical and current performance of operators. The selection probability of operator i is updated after the j-th iteration according to the following equation:

$$P_i^{j+1} = \frac{\omega_{i,j+1}}{\sum_{i' \in DS \text{ or } RS} \omega_{i',j+1}} \tag{30}$$

3.1. Initial Solution Generation

The generation method of the initial solution is presented in Algorithm 2.

Algorithm 2: Generation of the initial solution

Input: arc_dict, floyd_matrix, e_{max}
Output: initial_solution

$R_{total} \leftarrow \emptyset$
for arc $\in$ arc_dict **do**
    unvisited_arc_list $\leftarrow$ arc_dict
    unvisited_arc_list $\leftarrow$ unvisited_arc_list$\backslash${arc}
    $i \leftarrow$ arc.node_i, $j \leftarrow$ *arc.node_j*, *subroute* $\leftarrow$ [*i*,*j*]
    $R_{current} \leftarrow \emptyset$
    while unvisited_arc_list $\neq \emptyset$ **do**
        while calculate_route_cost(subroute) $<$ e_{max} **do**
            if unvisited_arc_list = $\emptyset$ **then**
                break
            end if
            next_arc $\leftarrow \underset{\alpha \in \text{unvisited_arc_list}}{\text{argmin}} d_{floyd}(\text{end}(R_{current}), \alpha)$
            Add the nearer node of next_arc to subroute, then the other
            unvisited_arc_list $\leftarrow$ unvisited_arc_list$\backslash${next_arc}
        end while
        $R_{total} \leftarrow R_{total} \cup$ {subroute}
        subroute $\leftarrow\emptyset$
    end while
    $R_{total} \leftarrow R_{total} \cup \{R_{current}\}$
    Get the other direction(j,i) of arc, subroute $\leftarrow$ [j, i]
    Repeat steps 6-19
end for
initial_solution $\leftarrow \underset{R \in R_{total}}{\text{argmin}}(W(R))$

The drone inspection demand arc set is $E = \{e_1,e_2,\ldots,e_n\}$, and the road network node set is $V = \{v_1,v_2,\ldots,v_m\}$. The two endpoints of each demand arc correspond to two road network nodes. Assuming that the nodes corresponding to the two ends of demand arc e_k are v_i and v_j, the endpoints of demand arc e_k are denoted as s_{ij} and s_{ji}. Therefore,

each demand arc has two travel directions, represented as (s_{ij}, s_{ji}) and (s_{ji}, s_{ij}) respectively. Furthermore, the calculation method of the distance matrix between endpoints is as follows.

$$d(s_{ij}, s_{kl}) = \begin{cases} 0 & \text{if } (i,j),(k,l) \in E \text{ and } (i,j) = (k,l) \\ d_{act}(i, j) & \text{if } (i,j),(k,l) \in E \text{ and } (i,j) = (l,k) \\ d_{Euc}(i, k) & \text{if } (i,j),(k,l) \notin E \text{ and drone flies directly} \\ d_{SP}(i, k) & \text{if } (i,j),(k,l) \notin E \text{ and drone flies along the road} \end{cases} \quad (31)$$

Based on the above settings, the initial solution is generated using a greedy strategy: each demand arc is traversed as the starting arc, and two travel directions are considered for each arc. For each direction, starting from the starting arc, the uninspected arc closest to the end of the current route is selected each time, and its proximal endpoint is preferentially added to the route. A new sub-route is created when the cumulative flight cost of the sub-route reaches the maximum battery endurance of the drone, until all demand arcs are traversed. Each starting arc corresponds to two total routes, resulting in a total of $2n$ candidate routes, and the one with the minimum cost is selected as the initial solution.

3.2. Encoding and Decoding

After each destruction and repair operation, the total route is encoded as a set of sub-routes $R = \{r_1, r_2, \ldots, r_u\}$. Each sub-route corresponds to a single fully charged flight of the drone. The total route cost is taken as the minimum value of two connection methods: $W = min\{W_1, W_2\}$, where the sequential connection cost W_1 is calculated as follows:

$$W = \sum_{k=1}^{u} f(r_k) + \sum_{i=1}^{u-1} f(r_i, r_{i+1}), \quad (32)$$

where $f(r_k)$ denotes the flight cost of sub-route r_k, which includes the path cost of each demand arc within the sub-route and the direct flight distance cost between adjacent demand arcs. $f(r_i, r_{i+1})$ denotes the direct flight distance cost between two consecutive sub-routes.

The direct flight connection cost W_2: Within the maximum battery endurance limit, adjust the order of sub-routes and select the optimal direct flight path to generate a new route, then calculate the cost using the above formula.

3.3. Validation of Vehicle–Drone Coordination Logic and Penalty Mechanism

Feasibility verification of vehicle–drone coordination logic is required during operations such as initial solution generation, demand arc removal by destroy operators, and demand arc insertion by repair operators to ensure that the obtained route solutions comply with real-world operational logic. The feasibility verification mainly focuses on whether the route satisfies two core constraints: "the drone's flight time per mission cannot exceed its maximum battery endurance" and "the vehicle must arrive at the battery swapping node earlier than the drone". The verification of battery endurance feasibility and vehicle reachability must consider the impact of direct flight segments on the drone's battery endurance. Assume a route solution is $R = \{r_1, r_2, r_3\}$, where $r_1 = \{1, 2, 3\}$, $r_2 = \{3, 4, 5, 6, 9\}$, and $r_3 = \{9, 8, 7\}$. The drone flies directly between sub-routes r_2 and r_3, and the following battery endurance constraints must be satisfied: $T_{drone}(1, 2, 3) < e_{max}$, $T_{drone}(3, 4, 5, 6) + t_{drone}(6, 9) < e_{max}$, $T_{drone}(9, 8, 7) < e_{max}$. From the above example, the general battery endurance constraint can be summarized as follows:

$$T_{drone}(r_n) + t_{drone}\left(r_n^{end}, r_{n+1}^{start}\right) < e_{max}, \quad (33)$$

where $T_{drone}(r_n)$ denotes the time required for the drone to fly sub-route r_n along the shortest path, and $t_{drone}\left(r_n^{end}, r_{n+1}^{start}\right)$ denotes the direct flight time of the drone from the end point of sub-route r_n to the start point of sub-route r_{n+1}. When the drone does not choose to fly directly between sub-routes, the end point of r_n is exactly the start point of r_{n+1}, and thus $t_{drone}\left(r_n^{end}, r_{n+1}^{start}\right) = 0$.

In addition, the timely reachability of the vehicle is also an essential step in the feasibility verification of vehicle–drone coordination logic. In the above example, the following constraints must be satisfied to ensure timely battery swapping for the drone:

$$T_{car}\left(r_n^{start}, r_{n+1}^{start}\right) < T_{drone}(r_n) + t_{drone}\left(r_n^{end}, r_{n+1}^{start}\right), \tag{34}$$

where $T_{car}\left(r_n^{start}, r_{n+1}^{start}\right)$ denotes the time required for the vehicle to travel from the start point of the current sub-route to the start point of the next sub-route along the shortest path.

During the route repair and the route solution cost calculation phase, the algorithm introduces penalty costs for two types of infeasible situations: when a sub-route exceeds the drone's maximum battery endurance, and when the vehicle arrives at the battery swapping node later than the drone. A penalty is imposed whenever the obtained new route solution violates the vehicle–drone coordination logic, so as to avoid additional energy consumption caused by infeasible coordination schemes.

3.4. Destroy Operators

3.4.1. Random Removal

Randomly remove q% of the demand arcs from the existing total route of demand arcs, and add the removed demand arcs to the removed demand arc list.

3.4.2. Cluster Removal

Demand arcs are removed based on their similarity degree. The similarity between demand arcs is measured in both spatial and temporal dimensions [6]. Specifically, first randomly remove one demand arc e_1 from the total route. Then calculate the similarity between each remaining demand arc and e_1, and remove the demand arc e_2 with the highest similarity. Subsequently, remove the demand arc e_3 with the highest similarity to e_2. This process continues until q% of the demand arcs are removed.

3.4.3. Worst Cost Removal

The cost difference before and after removing a demand arc is defined as $\Delta W_{remove} = W(R) - W(R_{-e})$, where $W(R)$ denotes the route cost of the total route R, and $W(R_{-e})$ denotes the route cost of the total route R after removing demand arc e. The removal cost ΔW_{remove} of each demand arc is calculated and sorted in descending order, and the top q% of demand arcs are removed.

3.4.4. Battery Usage Rate Oriented Removal

The total route R consists of multiple sub-routes r, and the flight time of each sub-route does not exceed the maximum battery endurance. Therefore, the battery endurance utilization rate of the drone in each sub-route is defined as $UR(r_k) = T_{drone}(r_k)/e_{max}$, where $T_{drone}(r_k)$ denotes the flight time of the drone in sub-route r_k. The endurance utilization rate UR of each sub-route is calculated and sorted in descending order. For sub-routes with higher utilization rates, demand arcs are removed according to the principle of descending removal cost. The number of demand arcs removed from each sub-route is subject to a certain limit.

3.5. Repair Operators

This paper introduces a penalty cost mechanism to simultaneously optimize route efficiency and coordination feasibility. Within each repair operator, the battery endurance feasibility and battery swapping constraint rationality of the repaired route solution are incorporated into the cost calculation. When a repair operator inserts a removed demand arc, whether the new route solution after insertion at a preset position complies with real-world operational assumptions is used as the basis for penalty cost assessment. The penalty calculation method is defined as follows. The $C(i)$ denotes the cost increment after inserting demand arc e at position i, ΔW_{insert} denotes the total route cost increment after inserting arc e, and $\rho\Delta V(i)$ is the penalty term. The penalty cost $\Delta V(i)$ consists of two components: the total delay time $\Delta DT(i)$ when the vehicle arrives later than the drone, and the total overtime $\Delta OT(i)$ when the flight distance exceeds the drone's maximum battery endurance. ρ is the penalty coefficient.

$$C(i) = \Delta W_{insert} + \rho\Delta V(i) \tag{35}$$

$$\Delta W_{insert} = W(R_{+e}) - W(R) \tag{36}$$

$$\Delta V(i) = \Delta DT(i) + \Delta OT(i) \tag{37}$$

3.5.1. Random Insertion

For each demand arc e to be inserted, enumerate all candidate insertion positions. Calculate the cost $C(i)$ for each position i, and map the cost to a selection probability using the weight function $w(i) = 1/(1 + C(i))$. Finally, randomly select a number of positions equal to the number of removed demand arcs according to the probability distribution to insert the demand arcs.

3.5.2. Greedy Insertion

For each demand arc e to be inserted, always select the position with the minimum penalty cost $C(i)$ for insertion until all demand arcs to be inserted are processed.

3.5.3. Battery Usage Rate Oriented Insertion

Contrary to the endurance utilization-based removal operator, calculate the battery endurance utilization rate $UR(r_k)$ of each sub-route. Sort the sub-routes in ascending order of endurance utilization rate, and select sub-routes with lower utilization rates to insert a limited number of demand arcs. In each step, select the position with the minimum insertion cost $C(i)$ for insertion until all demand arcs are inserted.

4. Numerical Experiments

This chapter presents a series of numerical experiments to verify the effectiveness of the proposed ALNS algorithm. Section 4.1 first describes the instance generation rules and parameter settings. Then, the ALNS algorithm is compared with exact solution results on randomly generated instances to validate its effectiveness and solution stability. Section 4.2 conducts a case study based on the road network of Heshan City, Guangdong Province. The optimization performance of the drone direct-flight strategy on inspection routes is analyzed, and several key factors affecting the objective value of the model are investigated. The implementations in this chapter are based on Gurobi12.0.3 and python3.9.2.

4.1. Algorithm Performance Analysis

4.1.1. Test Instances and Parameter

Existing benchmark instances for the ARP only include node coordinates and arc lengths of a single road network. They do not distinguish between feasible arc sets for

vehicles and drones, nor do they consider the direct flight characteristics of drones, and thus cannot meet the research requirements of this paper. Therefore, this paper randomly generates test instances with reference to the specifications of ARP benchmark instances: the instance size is defined by the number of arcs; drone arc lengths are limited to 5–20 km to comply with their battery endurance constraints; 30% of arcs are removed from the drone arc set as the vehicle infeasible arc set to simulate the real-world scenario where vehicles are subject to road access restrictions; the direct flight distance of drones between nodes is generated by multiplying the corresponding arc length by a random reduction factor, which ranges from 0.7 to 0.9 and decreases as the arc length increases, to reflect the direct flight advantage of drones over long distances.

In the experimental solution of this section, we set the constant speeds of the drone and vehicle to 40 km/h and 30 km/h, respectively, which satisfies the real-world situation that the flight speed of the drone is higher than the travel speed of the vehicle. Considering the risk of thermal runaway at high temperatures [38,39], the drone's maximum endurance is taken as the full-charge nominal endurance multiplied by the thermal-safety SOC cap, set to 60 min. The maximum number of iterations is 1000, the maximum number of consecutive iterations without improvement is 100, the proportion of arcs removed by destroy operators q = 30%, the initial weights of all destroy-and-repair operators are 1, the weight score values (Φ_1,Φ_2,Φ_3) = (0.6, 0.3, 0.1), and the weight decay rate α = 20%. The parameter settings above are determined based on preliminary tuning experiments on small and medium-sized instances.

4.1.2. Computational Results

To verify the effectiveness of the ALNS algorithm, we compared Gurobi's exact solution, the GVNS algorithm [40], the classic ALNS algorithm [41], and the ALNS algorithm designed in this paper on different-sized randomly generated instances. All instances are solved for a limited time of 3600 s. The results are shown in Table 3. The performance of the solution was measured using the relative error index:

$$\begin{aligned} Gap_1 &= \frac{W - W_{best}}{W_{best}} \times 100\%,\ Gap_2 = \frac{W_{GV} - W_{best}}{W_{best}} \times 100\% \\ Gap_3 &= \frac{W_{CA} - W_{best}}{W_{best}} \times 100\%,\ Gap_4 = \frac{W_A - W_{best}}{W_{best}} \times 100\% \end{aligned} \tag{38}$$

In the table, W_{best}, W and T denote the best lower bound, final solution, and solving time obtained by Gurobi, respectively; W and T with subscripts GV, CA, and A denote the solution values and solving times of GVNS, classic ALNS, and the proposed ALNS, respectively.

The data in Table 3 indicate that in smaller-scale instances, the ALNS algorithm proposed in this paper can obtain better solutions. As the scale of the instances continues to increase, Gap_4 increases within a reasonable range. Compared with the GVNS algorithm and the classic ALNS algorithm, the solution values and solution efficiency of the ALNS algorithm proposed in this paper have a slight advantage.

To further verify the solution stability of the ALNS algorithm, six new instances with increasing scales are randomly generated. Each instance is run independently 30 times under the same parameter settings, and the best, worst, mean, and standard deviation of the total inspection time obtained over the 30 runs are reported in Table 4. The report shows that the solution of ALNS is stable under small-scale instances, and the solution results fluctuate within a reasonable range with the increase in the size of the instance.

In order to further verify the effectiveness of the battery-oriented operator proposed in this paper for the inspection scene, we compare the classical ALNS algorithm and conduct 20 experiments under different scales of instances to obtain the experimental results (total

inspection time) in Table 5. The results show that compared with the classical ALNS, the total inspection time is further optimized after adding the customized operator. The customized operators demonstrate a favorable optimization effect.

Table 3. Comparison of solution results for the example cases.

Instance	E	V	Gurobi				GVNS			C-ALNS			ALNS		
			W_{best}	W	Gap_1	T	W_{GV}	Gap_2	T_G	W_{CA}	Gap_3	T_{CA}	W_A	Gap_4	T_A
D20-V14	10	8	3.23	3.23	0.0%	1.1	3.23	0.0%	1.2	3.23	0.0%	1.2	3.23	0.0%	1.2
D30-V21	15	10	5.03	5.03	0.0%	2.4	5.03	0.0%	2.4	5.05	0.4%	3.3	5.03	0.0%	3.2
D40-V28	20	13	6.15	6.15	0.0%	12.6	6.17	0.2%	12.7	6.17	0.2%	8.7	6.17	0.2%	10.5
D50-V35	25	16	7.40	7.40	0.0%	48.7	7.40	0.0%	27.1	7.45	0.7%	30.6	7.45	0.7%	36.1
D60-V42	30	19	9.60	9.60	0.0%	51.5	9.63	0.3%	35.4	9.63	0.3%	31.8	9.60	0.0%	39.5
D70-V49	35	21	11.05	11.15	0.9%	77.7	11.25	1.8%	69.0	11.15	0.9%	71.6	11.16	1.0%	62.2
D80-V56	40	24	13.13	13.13	0.0%	95.1	13.39	2.0%	82.4	13.43	2.3%	83.6	13.23	0.8%	79.9
D90-V63	45	25	14.93	14.93	0.0%	135.4	15.45	3.5%	107.3	15.56	4.2%	148.2	15.35	2.8%	102.5
D100-V70	50	29	16.80	16.92	0.7%	189.2	17.65	5.0%	145.3	17.53	4.3%	129.2	17.16	2.1%	112.8
D110-V77	55	31	18.80	18.80	0.0%	328.4	19.99	6.3%	164.0	20.24	7.7%	175.5	19.37	3.0%	184.7
D120-V84	60	32	20.10	20.10	0.0%	487.7	21.48	6.8%	228.6	21.13	5.1%	211.5	20.21	0.5%	207.9
D130-V91	65	36	21.53	21.62	0.4%	611.3	22.35	3.8%	307.2	22.40	4.1%	252.9	22.24	3.3%	261.5
D140-V98	70	39	23.25	23.25	0.0%	1213.6	23.34	0.4%	401.8	23.35	0.4%	384.2	23.75	2.2%	374.2
D150-V105	75	41	24.80	24.94	0.6%	1474.5	26.22	5.7%	381.7	25.95	4.6%	362.8	25.76	3.9%	364.7
D160-V112	80	43	26.88	26.93	0.2%	2125.4	28.98	7.8%	529.3	28.17	4.8%	489.2	28.01	4.2%	415.3
D170-V119	85	46	28.28	28.65	1.3%	2851.7	30.18	6.7%	588.5	29.85	5.6%	624.4	28.75	1.7%	625.1
D180-V126	90	48	30.40	—	—	>3600	32.51	7.0%	676.3	32.60	7.2%	615.6	31.23	2.7%	591.9
D190-V133	95	50	33.18	—	—	>3600	33.90	2.2%	825.9	36.14	8.9%	737.5	35.07	5.7%	667.3
D200-V140	100	53	34.22	—	—	>3600	35.60	4.0%	834.3	35.42	3.5%	786.2	35.24	3.0%	717.5
D210-V147	105	56	—	—	—	>3600	40.25	—	905.2	40.35	—	814.1	39.53	—	708.7
D220-V154	110	58	—	—	—	>3600	48.25	—	886.3	47.85	—	802.6	45.49	—	789.7
D230-V161	115	60	—	—	—	>3600	44.85	—	1016.2	44.22	—	883.4	43.60	—	816.7
D240-V168	120	62	—	—	—	>3600	47.68	—	1058.4	46.78	—	912.2	44.93	—	845.6

Table 4. Stability analysis results.

Instance	E	Best	Worst	Mean	Std	CV [1]
D20-V14b	10	3.625	3.625	3.625	0.000	0.00%
D60-V42b	30	13.625	14.273	13.964	0.142	1.017%
D100-V70b	50	21.075	24.770	22.532	0.846	3.755%
D140-V98b	70	26.650	32.786	28.107	1.636	5.821%
D180-V126b	90	34.178	42.025	38.225	2.687	7.029%
D220-V154b	110	42.728	50.567	46.357	4.275	9.222%

[1] CV: denotes the coefficient of variation, defined as the standard deviation divided by the mean.

Table 5. Comparison of ablation experiment results.

Instance	E = 20	E = 30	E = 40	E = 50
(i) Standard ALNS [1]	9.242	15.973	21.433	28.032
(ii) Standard + customized destroy	9.037 (−2.22)	15.950 (−0.14)	20.863 (−2.66)	27.817 (−0.77)
(iii) Standard + customized repair	9.092 (−1.62)	15.645 (−2.05)	20.670 (−3.56)	27.965 (−0.24)
(iv) Complete proposed ALNS	8.910 (−3.60)	15.050 (−5.78)	19.705 (−8.06)	27.315 (−2.56)

[1] The standard ALNS adopts the classical operators—random, cluster, and worst-cost removal, together with random and greedy insertion.

4.2. Case Study

4.2.1. Case Description

Heshan City, Guangdong Province, has carried out drone inspection work on its roads. This section takes the real road network of Heshan City as the empirical research object, and selects the main provincial roads, county roads, and some township roads of the city as the inspection scope. According to statistics, the total length of inspection demand arc segments is 383.51 km, including 140.55 km of provincial roads, 226.95 km of county

roads, and 16.01 km of township roads. The inspection demand road network is shown in Figure 5.

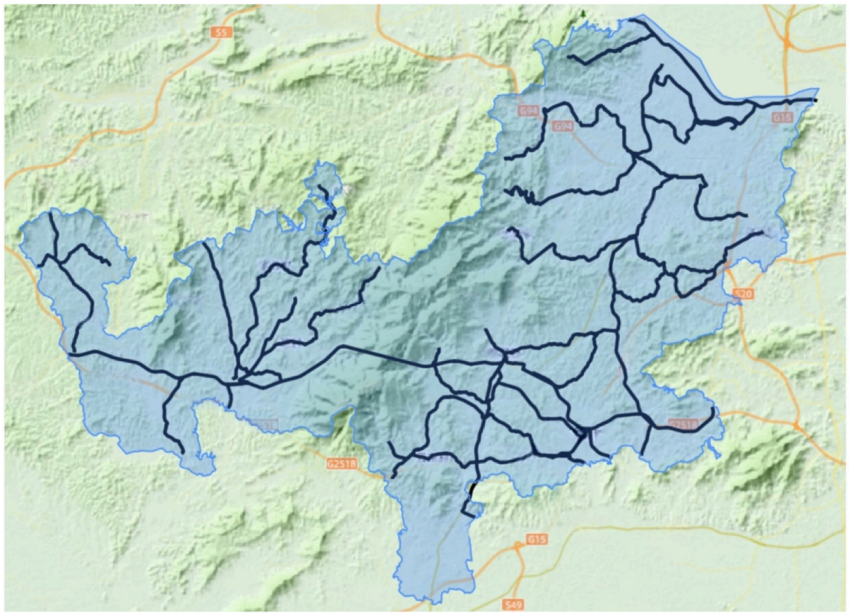
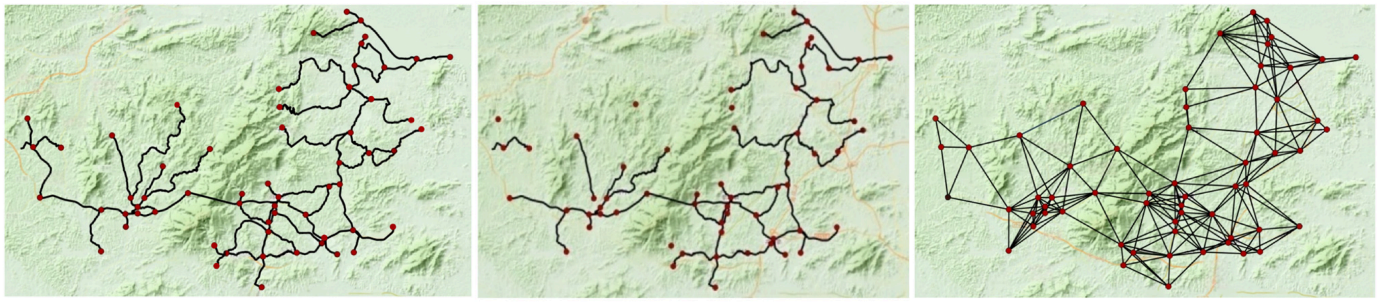


Figure 5. Road network of Heshan City.

4.2.2. Experimental Analysis

The preprocessing steps for the Heshan road network experiment are as follows: select road intersections and ends as potential battery swapping nodes, divide overlong roads into short arc segments that meet the drone's battery endurance requirements, and obtain the drone inspection network E1 (Figure 6a) with a total inspection demand path of 383.51 km; retain the sections accessible to the inspection vehicles to obtain the vehicle travel network E2 (Figure 6b); add direct flight arcs between nodes with a straight-line distance less than 10 km to generate the drone direct flight network E3 (Figure 6c), based on the average distance (typically 8–12 km) of direct flight nodes along the Heshan road network.



(**a**) Drone inspection road network (**b**) Vehicle travel road network (**c**) Drone direct flight network

Figure 6. Vehicle–drone inspection networks.

The existing vehicle–drone inspection mode is the drone-synchronized mode, in which the drone flies synchronously with the inspection vehicle at a fixed relative position. In this mode, the drone flies at the speed of the vehicle and cannot adopt the direct flight strategy. This mode retains no routing freedom and reduces exactly to a Chinese postman problem [42], which we solve to provable optimality. This paper compares the proposed vehicle–drone cooperative mode with the drone-synchronized mode to quantify the improvement of the inspection mode proposed in this paper. Based on the road network data of Heshan, the results obtained by solving the two inspection modes are as

follows. Figure 7 shows the solution path of the vehicle–drone inspection mode in this paper. The blue curve represents the vehicle's travel trajectory, and the red straight line represents the direct flight trajectory of the drone. From the figure, it can be seen that the vehicle mainly travels along the main roads of the road network and collaborates with the drone to complete the battery replacement task. The comparison results (Table 6) indicate that compared with the drone-synchronized mode, the proposed hybrid flight mode reduces the non-inspection path mileage of the drone by 61.32% and the vehicle's travel distance by 75.04%, significantly shortening the redundant flight distance and enhancing the inspection efficiency.

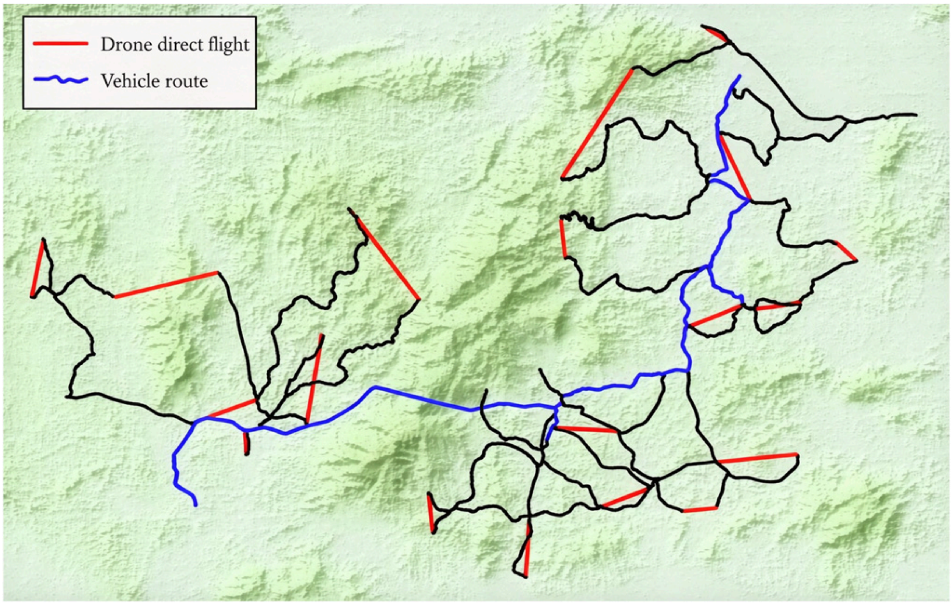


Figure 7. Schematic diagram of the overall route for vehicle–drone collaborative inspection.

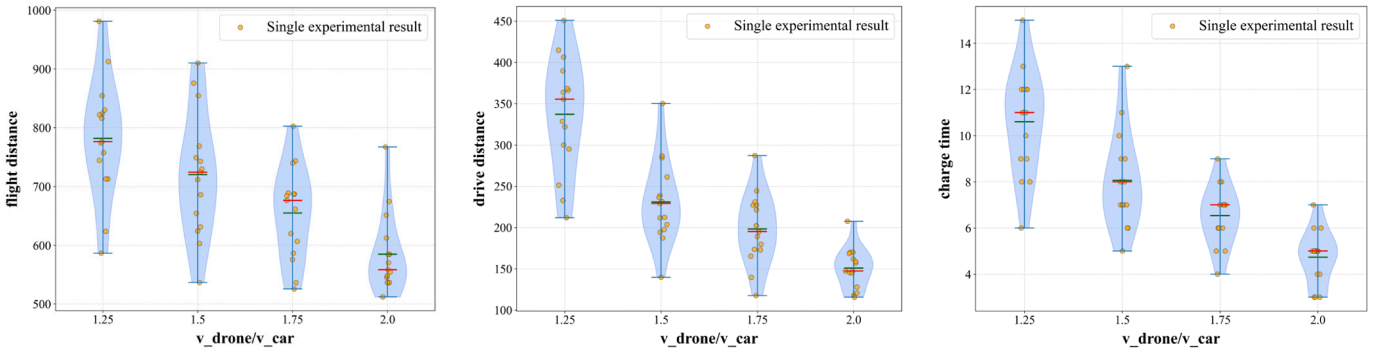
Table 6. Comparison of routing schemes.

Inspection Mode	Drone Inspection Path Distance (km)	Drone Non-Inspection Path Distance (km)	Total Drone Flight Distance (km)	Vehicle Travel Distance (km)	Total Inspection Time (h)
Drone-synchronized mode	383.51	278.83	662.34	662.34	23.18
Drone hybrid flight mode	383.51	107.84	491.36	165.34	12.98
Comparison	-	61.32%	25.81%	75.04%	43.98%

4.2.3. Analysis of the Key Features

Drone-to-Vehicle Speed Ratio

This section analyzes the impact of drone flight speed, vehicle travel speed, and their ratio on inspection indicators. Four groups of speed ratios are set in the experiment: 1.25, 1.5, 1.75, and 2. The drone speed ranges from 30 to 70 km/h, and the vehicle speed ranges from 20 to 40 km/h. All experimental groups ensure that the drone flight speed is higher than the vehicle travel speed. Multiple repeated experiments are conducted under each speed ratio. The distribution of drone travel distance, vehicle travel distance, and battery swapping times corresponding to different speed ratios is shown in Figure 8, and the mean values of each indicator are shown in Table 7.



(**a**) Impact of Speed Ratio on Drone Flight Distance

(**b**) Impact of Speed Ratio on Car Travel Distance

(**c**) Impact of Speed Ratio on Charge Times

Figure 8. Numerical distribution under different speed ratios.

Table 7. Mean values of indicators under different speed ratios.

Speed Ratio	Average Drone Travel Distance (km)	Average Vehicle Travel Distance (km)	Average Battery Swapping Times
1.25	781.94	337.23	11
1.5	720.21	231.10	8
1.75	654.88	198.37	7
2	584.87	150.82	5

The results show that as the drone–vehicle speed ratio increases, the drone travel distance, vehicle travel distance, and battery swapping times all show a downward trend; among them, the decrease is most significant when the speed ratio increases from 1.25 to 1.5, and the optimization effect gradually diminishes as the speed ratio continues to increase. The core reason is that the increase in speed ratio expands the drone's activity range, increases the choice of direct flight paths, and improves battery utilization; however, when the speed difference is too large, the vehicle's reachability becomes the main constraint, limiting the path optimization space of the drone. Each indicator presents a distribution characteristic of first dispersion and then convergence as the speed ratio increases. The above results indicate that matching a reasonable drone–vehicle speed ratio can effectively reduce invalid travel mileage, which is an important way to cut energy consumption and carbon emissions in inspection operations.

Drone Endurance

Keeping other parameters unchanged, multiple groups of drone maximum battery endurance values are set for repeated experiments, and the mean values are taken to obtain the variation trends of each indicator (Figure 9). On the whole, the drone battery endurance is negatively correlated with the total inspection time, drone travel distance, vehicle travel distance, and battery swapping times. When the drone battery endurance is less than 2 h, each indicator decreases significantly with the increase in endurance; increasing endurance can expand the drone's path selection range and reduce path redundancy and battery swapping times. When the battery endurance is greater than 5 h, the problem reduces to the classic Rural Postman Problem. The drone and vehicle travel distances approach the minimum travel distance of the equivalent RPP, the total inspection time converges to 11 h, and the battery swapping times approach 0.

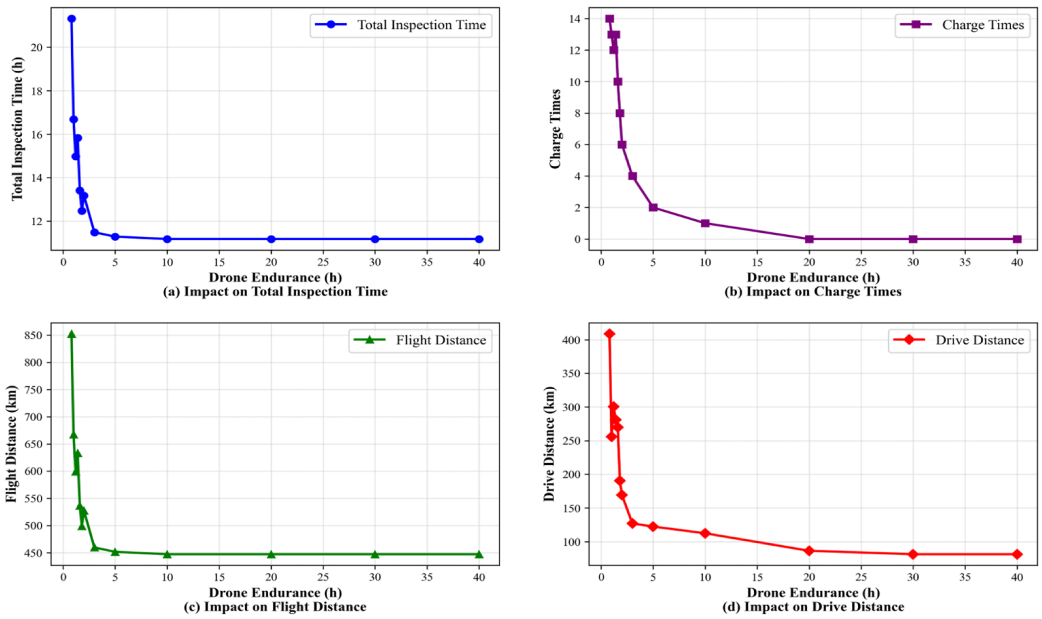
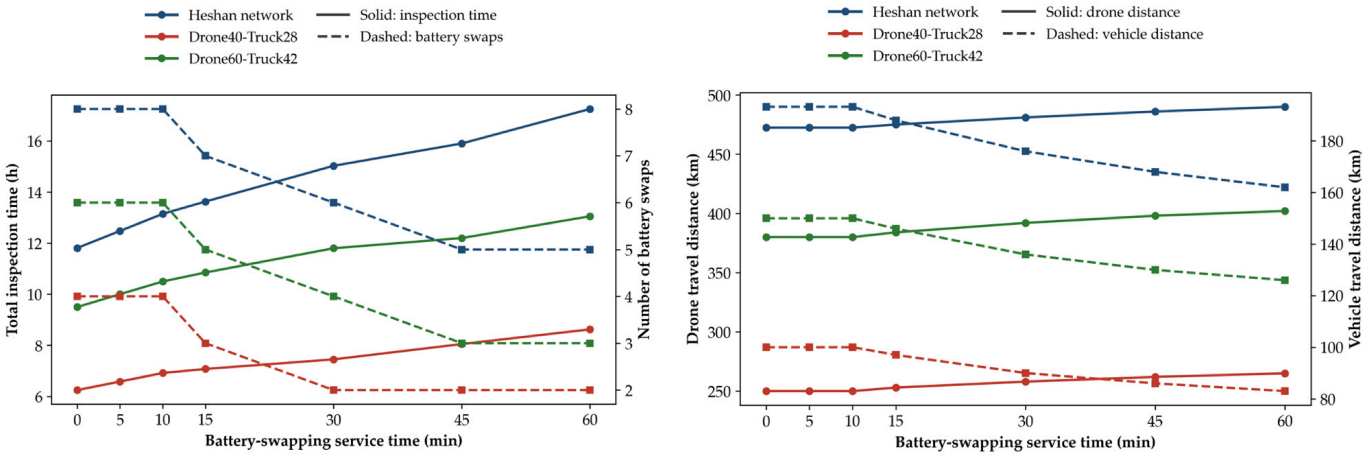


Figure 9. Impact of range on performance.

Charging Operation Time

To evaluate the impact of the battery swapping service time t_c, we conducted a sensitivity test based on the Heshan network and two instances. Figure 10 shows the influence of the battery swapping time on the path results. When t_c is within the normal operation time range (0–10 min), the number of battery swaps, the distance traveled by the drone, and vehicle remain basically unchanged, while the total inspection time increases linearly, indicating that within this range, the battery swapping time does not change the optimal route structure. As the battery swapping time further increases, the number of battery swaps shows a stepwise decrease; the flight distance of the drone slightly increases due to the longer flight segments, and the vehicle travel distance decreases owing to fewer rendezvous points, which slows down the growth of the total inspection time.



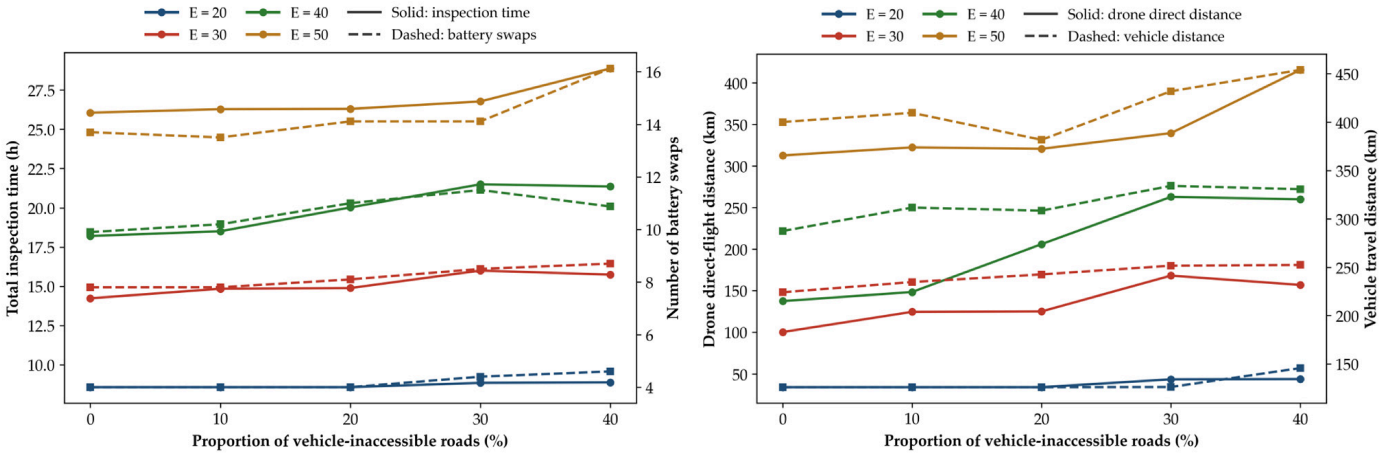
(**a**) Effect on Inspection Time and Swapping Times (**b**) Effect on Drone and Vehicle Travel Distance

Figure 10. Impact of battery-swapping service time.

Vehicle-Inaccessible Ratio

In the network of Heshan City, the proportion of roads where inspection vehicles cannot pass is 30%. In order to further explore the impact of the proportion of feasible road sections for vehicles on the total inspection time, we conducted multiple experiments for

different-sized instances under different proportions of inaccessibility for vehicles, and obtained the results shown in Figure 11. The experiment shows that as the number of inaccessible segments increases, the drone requires more direct flights, resulting in longer total inspection time and more battery swap times. Additionally, the vehicles need to travel a greater distance for detouring. The impact of the spatial distribution pattern of vehicle-inaccessible roads on collaborative inspection is presented in Appendix A.



(**a**) Effect on Inspection Time and Swapping Times (**b**) Effect on Drone and Vehicle Travel Distance

Figure 11. Impact of vehicle-inaccessible ratio.

Road Network Structure

The routing conclusions above rest on two settings related to road-network structure. First, battery swapping is confined to road-network nodes. Second, the synthetic instances follow a single generation rule with a different random seed. These settings will significantly affect the topological structure of the road network; we performed two additional analyses. First, we allow battery swapping at a position along a road. Second, we conduct a comparative analysis on grid, sparse, clustered, and heterogeneous networks. The experimental results (Appendix C.1) show that allowing drones and vehicles to rendezvous and swap batteries on the road does not produce a systematic improvement. In different topological road networks, compared with other algorithms, the ALNS proposed in this paper still maintains its performance advantage (Appendix C.2).

4.3. Energy Consumption Assessment

To further quantify the change in energy consumption under the inspection mode proposed in this paper, this section converts the model from a time account into an energy account, considering both the energy consumed by the drone in its two flight modes and the energy consumed by the electric inspection vehicle.

The parameters are set as follows: during the inspection, the drone flies at a speed of 20 km/h due to imaging and other operational requirements; during non-inspection periods, it maintains a fixed cruising speed of 40 km/h. The electric vehicle travels at a constant speed of 30 km/h. The propulsion power of the rotorcraft drone when flying at a speed of v is given by the closed-form model derived by Zeng et al. [43] and experimentally validated with about 12,000 power–speed measurements by Gao et al. [44]:

$$P(v) = P_0 + P_i g(v) + c_5 v^3 \tag{39}$$

$$g(v) = \left[\sqrt{1+u^2} - u\right]^{1/2},\ u = \frac{v^2}{2v_0^2},\ c_5 = \frac{1}{2} d_0 \rho s A \tag{40}$$

where P_0 and P_i are the blade-profile and induced powers in hover, v_0 is the mean induced velocity in hover, and the remaining symbols are listed in Appendix B.1. Therefore, the energy consumption per unit distance is:

$$\alpha(v) = P(v)/v \tag{41}$$

Evaluating (41) at the two operating speeds gives the two energy coefficients that the model actually uses, which are summarized in Table 8.

Table 8. Energy coefficients at the two operating speeds.

Flight Mode	Speed (km/h)	Power (W)	α (kWh/km)
Inspection	$v_1 = 20$	627	$\alpha_p = 0.031$
non-inspection	$v_2 = 40$	533	$\alpha_c = 0.013$

For the electric inspection vehicle, the speed is constant, so its energy consumption per kilometer is a constant as well, $\alpha_v = 0.2$ kWh/km. Its derivation from the road-load model and its cross-check against the manufacturer's certified consumption are given in Appendix B.2. Based on the above unit mileage energy consumption, we transformed the objective function into the following form:

$$\min E_{total} = \alpha_p \sum_{i,\, j \in V_{total}} d_{ij}^{drone} x_{ij} + \alpha_c \sum_{i,\, j \in V_{total}} d'^{drone}_{ij} x'_{ij} + \alpha_v \sum_{i,\, j \in V_{total}} d_{ij}^{car} y_{ij} \tag{42}$$

For a single flight in the inspection scenario, the energy consumption of the drone does not exceed the maximum energy capacity of its battery, as follows. The equivalent capacity of the battery is calculated based on the inspection speed v_1 according to the dominant flight mode of the task.

$$E_{leg} = \alpha_p \sum_{leg} d_{ij}^{drone} + \alpha_c \sum_{leg} d'^{drone}_{ij} \leq C_{bat} \tag{43}$$

$$C_{bat} = P(v_1) e_{max} \tag{44}$$

By applying the transformed energy consumption model to the data of Heshan, we can obtain the energy consumption differences under the two inspection modes (Table 9). The data indicate that, under the goal of minimizing the total energy consumption of the inspection process, the drone's flight path during the inspection adopts more direct flights in order to minimize the vehicle's travel distance (as the unit energy consumption of the vehicle is much higher than that of the drone). The hybrid flight mode proposed in this paper can reduce 81.7% of energy consumption in the Heshan inspection scenario.

Table 9. Energy consumption under two inspection modes.

Inspection Mode	Total Drone Flight Distance (km)	Drone Energy Consumption (kWh)	Vehicle Travel Distance (km)	Vehicle Energy Consumption (kWh)	Total Energy Consumption (kWh)
Drone-synchronized mode	662.34	17.30	662.34	132.47	149.76
Drone hybrid flight mode	570.56	14.32	65.35	13.07	27.39

5. Conclusions

Against the background of low-carbon transformation and sustainable development of transportation infrastructure, this paper addresses the limitations of low inspection efficiency and incomplete coverage of blind areas such as roadside slopes in conventional road maintenance inspection. A drone routing optimization problem for road inspection with dynamic battery swapping strategies is formulated, whose core objective is to minimize the total inspection time under the prerequisite that all required demand arcs are fully traversed. Key constraints of the formulated problem consist of limited drone flight endurance, road-network-based travel restrictions for ground vehicles, and temporal-spatial coordination requirements at battery swapping nodes. The primary challenge lies in reducing the non-inspection flight distance of drones via route optimization while coordinating the coupled routing relationship between vehicles and drones.

To tackle the above problem, a mixed-integer programming model is established. By comprehensively distinguishing flight characteristics between the inspection and non-inspection phases of drones, as well as disparate travel networks for vehicles and drones, a three-layer network framework is constructed, including the drone inspection network, drone direct-flight network, and vehicle travel network. Virtual nodes are introduced to eliminate conflicts in endurance constraints originating from repeated arc traversal, and binary decision variables are adopted to flexibly determine the start and end locations of inspection routes. To overcome the difficulty of exact model solution on large-scale road networks, an Adaptive Large Neighborhood Search (ALNS) Algorithm is developed with four destroy operators and three repair operators embedded. An optimization strategy oriented by endurance utilization rate is integrated into the algorithm, and a penalty mechanism is incorporated to guarantee the logical coupling of vehicle–drone routes. Numerical experiments verify the efficiency and stability of the proposed ALNS across instances of diverse sizes. Compared with the exact solver, the GVNS algorithm, and the classical ALNS algorithm, the proposed algorithm drastically shortens computational time for large-scale test cases, with the objective gap controlled within 6%. A practical case study based on the real road network of Heshan City validates the applicability of the constructed model; the adoption of a drone direct-flight strategy cuts non-inspection drone flight mileage by 61.32% and reduces vehicle travel mileage by 75.04%, achieving an 81.7% reduction in total system energy consumption. The scheme delivers significant low energy consumption benefits while improving inspection efficiency. In addition, parametric analysis confirms that a reasonable vehicle–drone speed ratio and appropriately improved drone endurance can effectively improve overall inspection efficiency.

The findings of this study have the following policy implications for promoting the new energy application and sustainable development of road maintenance:

1. Road management authorities should promote the vehicle–drone cooperative inspection mode and replace traditional fuel-powered inspection fleets with electric vehicles and drones. Through the optimization of the inspection mode, it helps the road maintenance transition to low-carbon.
2. The application of drones in road maintenance inspection should be expanded to achieve full coverage of blind areas such as road slopes. Early detection of structural defects reduces the demand for large-scale overhauls and extends the service life of road infrastructure.
3. Technical research on new energy power batteries should be further strengthened. Longer battery endurance can reduce battery swapping times and invalid flight mileage, delivering more substantial carbon reduction benefits for inspection operations.

https://doi.org/10.3390/su182110842

Although the proposed modeling framework and optimized algorithm provide feasible solutions for vehicle–drone cooperative road inspection, potential extensions remain for further research. Future work can introduce new energy fixed charging stations integrated with photovoltaic power supply to remedy the limitation that ground vehicles may fail to arrive at rendezvous points on time, transforming the original problem into an integrated routing and facility location optimization problem with low-carbon objectives. Moreover, subsequent research can incorporate cost variations induced by limited battery inventory, thereby converting the problem into a multi-objective optimization problem balancing efficiency, economy, and sustainability. Relevant explorations will further enhance the engineering applicability and sustainable development value of the routing schemes.

Author Contributions: Conceptualization, W.C. and Q.L.; methodology, Q.L.; software, Q.L.; validation, Q.L. and Y.G.; resources, W.C. and Y.G.; data curation, Q.L. and W.C.; writing—original draft preparation, Q.L.; writing—review and editing, Y.G. and W.C.; visualization, Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received funding support from the Fundamental Research Funds for the Central Universities (grant number CXTD202407).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request. The data are not publicly available due to privacy concerns.

Acknowledgments: The authors gratefully acknowledge the reviewers for their comments to improve the quality of this paper and would also like to thank the editors for their help with this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

In the generation of the test instances, the vehicle travel network is obtained by removing 30% of the arcs from the drone inspection network at random. To further examine whether the spatial distribution of these inaccessible roads affects the coordinated inspection, four accessibility patterns are compared on the same randomly generated road networks: the inaccessible roads are either sampled uniformly at random, or concentrated around one, two, and three mutually distant centers. Only the spatial distribution differs between the patterns, while the drone inspection arc set is left unchanged. Tables A1 and A2 report the summary and the detailed results, respectively. Only the number of battery swaps shows a slight trend with the degree of concentration, which is nevertheless not statistically significant, while the other indicators are essentially unaffected. These results indicate that, for the test instances, the spatial pattern of the vehicle-inaccessible roads has no significant effect on the inspection performance.

Table A1. Effect of the accessibility pattern on each indicator.

Indicator	ρ	t
Total inspection time	0.04	0.26
Number of battery swaps	−0.22	−1.34
Drone direct-flight distance	0.12	0.76
Vehicle travel distance	−0.17	−1.38

Note: ρ is the Spearman rank correlation coefficient between the degree of concentration of the inaccessible roads and the indicator; t is the test statistic.

Table A2. Detailed results per road network and accessibility pattern.

Road Network	Pattern	Total Inspection Time (h)	Battery Swaps	Direct-Flight Distance (km)	Vehicle Distance (km)
g1-30	k = 1	15.643	8.5	153.7	254
g1-30	k = 2	14.773	8.4	119.3	200.6
g1-30	k = 3	15.252	7.9	140.5	209.9
g1-30	Random	15.645	8.5	153.8	228.6
g2-30	k = 1	13.152	6.8	92.9	197.7
g2-30	k = 2	13.805	7.2	117.4	224.8
g2-30	k = 3	13.963	7.4	122.9	194.3
g2-30	Random	13.512	7.3	105.3	211.7
g3-30	k = 1	11.117	5.8	63.5	154.6
g3-30	k = 2	11.162	5.9	64.9	159.3
g3-30	k = 3	11.122	5.9	63.3	159.3
g3-30	Random	10.787	5.6	51.1	139.3
g4-30	k = 1	11.682	5.9	82.7	164.9
g4-30	k = 2	11.467	5.9	74.1	168.6
g4-30	k = 3	11.255	6.1	64.8	159.3
g4-30	Random	11.55	6.2	76.2	165.8
g5-30	k = 1	11.578	5.9	74.5	162.6
g5-30	k = 2	11.502	6.1	70.7	168.1
g5-30	k = 3	11.725	6.2	79.2	167.6
g5-30	Random	11.883	6.2	85.5	178.4
g1-40	k = 1	21.075	11.5	246	332.2
g1-40	k = 2	21.275	11.6	253.6	327.1
g1-40	k = 3	21.837	11.6	276.1	355
g1-40	Random	20.67	11	231.8	313.4
g2-40	k = 1	17.905	9.3	168	265.5
g2-40	k = 2	17.317	9.3	144.5	253
g2-40	k = 3	17.308	8.8	146.1	255.1
g2-40	Random	17.335	9.4	144.8	277.1
g3-40	k = 1	16.565	8.7	134.8	248
g3-40	k = 2	16.345	8.8	125.6	227.7
g3-40	k = 3	16.392	8.5	128.7	243.1
g3-40	Random	16.445	9	128.8	251.1
g4-40	k = 1	16.185	8.8	138.2	234.2
g4-40	k = 2	15.953	8.2	131.3	232.6
g4-40	k = 3	15.877	8.7	126.3	249.1
g4-40	Random	15.93	8.6	128.8	236.4
g5-40	k = 1	16.585	8.9	137.8	230.6
g5-40	k = 2	16.74	8.7	144.8	235.8
g5-40	k = 3	15.768	8.1	108.3	211.5
g5-40	Random	16.753	8.9	144.5	254
g1-3-50	k = 1	24.558	13.07	282.7	374.3
g1-3-50	k = 2	25.774	13.77	330.4	379.4
g1-3-50	k = 3	25.68	13.48	323	381
g1-3-50	Random	25.752	13.82	325.6	395.9
g4-5-50	k = 1	22.496	12.26	259.9	327.8
g4-5-50	k = 2	22.093	11.75	245.7	323.9
g4-5-50	k = 3	22.7	12.15	268.4	333.9
g4-5-50	Random	22.304	11.84	253.6	310.3

Note: each value is the mean over 20 independent runs for every road network.

Appendix B

Appendix B.1

The current inspection drone used in Heshan is the DJI Matrice 350 RTK. The parameters of the drone power consumption model are shown in Table A3.

Table A3. Drone power model parameters.

Symbol	Meaning	Value
A	Total rotor disk area	0.894 m^2
v_0	Mean induced velocity in hover	5.384 m/s
P_i	Induced power in hover	375.9 W
c_5	Parasite power coefficient	0.0164 $W{\cdot}s^3/m^3$
P_0	Blade-profile power in hover	332.5 W
ρ	Air density	1.225 kg/m^3
d_0,s,δ	Fuselage drag ratio, solidity, profile drag coefficient	0.6, 0.05, 0.012

Appendix B.2

The current inspection vehicles are equipped with the Geely Radar RD6 series. The energy consumption model is based on the research of Guzzella et al. [45] and Wasserfall et al. [46] is as follows.

$$P_{veh} = \frac{m_v g C_{rr} v_{car} + \frac{1}{2}\rho C_d A_f v_{car}^3}{\eta} + P_{aux}, \quad \alpha_v = \frac{P_{veh}}{v_{car}} \tag{A1}$$

The parameters are shown in Table A4.

Table A4. Vehicle power model parameters.

Symbol	Meaning	Value
m_v	Working mass	2390 kg
C_{rr}	Rolling resistance coefficient	0.012
C_d	Aerodynamic drag coefficient	0.4
A_f	Frontal area	2.7 m^2
η	Drivetrain efficiency	0.85
P_{aux}	Auxiliary load (HVAC, lighting, instruments)	1.5 kW
v_{car}	Vehicle speed	30 km/h

Appendix C

Appendix C.1

Based on the instances and the Heshan case shown in Table A5, we adopted the midpoint splitting method for the arc segments and added candidate charging nodes along the road. The operation is controlled: it does not change the required inspection mileage (the two halves sum to the original road length), nor the optimal tour of the vehicle network—subdividing an edge changes neither shortest-path distances nor vertex parities (the new node has degree 2). The only difference between the two versions is therefore the relaxation of the swapping-point assumption. The experimental results are shown in Table A5. It shows that relaxing the restriction that battery swapping must occur at road-network nodes does not yield a systematic improvement: across the ten synthetic instances, the average change (+2.1%) is smaller than one standard error, and the sign reverses repeatedly between adjacent scales—indicating that the difference is governed by the geometry of the individual network rather than by its scale. The Heshan network is likewise not improved (+8.2% on the best value).

Table A5. Effect of battery-swapping point location.

Instance	E	N	Best Result	Mean Value
E20	20 → 26	13 → 19	8.38 → 8.63 (2.99%)	8.68 → 9.18 (5.79%)
E30	30 → 38	19 → 27	13.10 → 12.88 (−1.72%)	13.62 → 13.19 (−3.16%)
E40	40 → 49	24 → 33	15.88 → 15.78 (−0.63%)	16.69 → 17.21 (3.13%)
E50	50 → 61	29 → 40	19.98 → 20.78 (4.01%)	20.75 → 22.18 (6.92%)
E60	60 → 72	32 → 44	22.73 → 24.80 (9.13%)	23.13 → 28.18 (21.81%)
E70	70 → 84	39 → 53	28.58 → 27.33 (−4.37%)	30.07 → 27.81 (−7.52%)
E80	80 → 102	43 → 65	31.88 → 33.38 (4.71%)	35.83 → 35.61 (−0.59%)
E90	90 → 113	48 → 71	36.63 → 36.43 (−0.55%)	39.10 → 38.31 (−2.02%)
E100	100 → 125	53 → 78	39.93 → 42.63 (6.76%)	41.96 → 47.45 (13.09%)
E110	110 → 135	58 → 83	43.43 → 43.73 (0.69%)	45.95 → 44.13 (−3.98%)
Heshan	57 → 69	51 → 63	12.98 → 14.05 (8.20%)	14.07 → 15.02 (6.79%)

Appendix C.2

To examine whether the performance of the proposed method depends on how the instances are constructed, the algorithmic comparison is repeated on synthetic networks with different topological structures. The four networks are generated under identical scale and parameter settings: the same number of required arcs and physical nodes, arc lengths drawn from the same distribution. They differ only in the spatial arrangement of nodes, which yields four characteristic forms. In the grid network, nodes are placed on a regular lattice and connectivity is uniform. In the sparse network, nodes are distributed along a narrow corridor with relatively few junctions. In the clustered network, nodes concentrate in several mutually separated groups that are dense internally and connected by a small number of long arcs. In the heterogeneous network, one half of the nodes forms a dense core while the remainder is evenly dispersed, producing an uneven degree distribution. This construction leaves topology as the only varying factor. The experimental results show that the proposed ALNS algorithm has advantages over the classical ALNS algorithm and the GVNS algorithm in different topologies, achieving average improvements of 3.66% and 0.74%. In addition, we find that with the grid topology, it is easier to find the inspection path, and with the heterogeneous topology, it is relatively more difficult to find the optimal solution.

Table A6. Effect of network topology.

Instance	Topology	C-ALNS		GVNS		ALNS	
		Mean	Runtime	Mean	Runtime	Mean	Runtime
E30	Grid	9.82	35.90	9.83	28.40	9.80	37.10
	Sparse	11.71	42.40	11.33	48.80	11.33	44.60
	Clustered	11.64	36.90	11.19	36.80	11.22	37.40
	Heterogeneous	11.33	44.60	11.38	43.20	11.28	40.30
E40	Grid	13.18	63.50	12.68	69.00	12.66	63.60
	Sparse	14.85	72.00	14.56	77.00	14.56	75.10
	Clustered	16.34	72.00	14.45	75.30	14.48	72.07
	Heterogeneous	16.12	78.10	15.78	86.70	15.65	80.80
E50	Grid	17.49	127.30	17.55	122.30	17.33	118.50
	Sparse	18.31	139.40	18.18	196.30	17.91	173.90
	Clustered	17.96	157.40	17.92	174.10	17.90	130.90
	Heterogeneous	19.12	167.60	18.96	169.30	18.84	164.50
E60	Grid	24.95	214.70	21.10	203.00	20.73	210.50
	Sparse	21.42	228.10	20.87	253.30	20.82	228.90
	Clustered	21.03	207.30	20.92	186.30	20.88	232.20
	Heterogeneous	23.45	263.40	23.18	275.60	22.14	269.30

References

1. Cai, L.; Li, J.; Wang, K.; Luo, Z.; Qin, H. Optimal allocation and route design for station-based drone inspection of large-scale facilities. *Omega* **2025**, *130*, 103172. [CrossRef]
2. Ding, Y.; Wan, J.; Lin, W.; Wang, K. Coordinated last-mile deliveries with trucks and drones: A comparative study of operational modes. *J. Air Transp. Res. Soc.* **2024**, *3*, 100025. [CrossRef]
3. Wei, S.; Fan, H.M.; Ren, X.X.; Diao, X.L. Time-Dependent Vehicle Routing Problem with Drones Under Vehicle Restricted Zones and No-Fly Zones. *Appl. Sci.* **2025**, *15*, 2207. [CrossRef]
4. Wu, G.H.; Zhao, K.X.; Cheng, J.Q.; Ma, M.H. A Coordinated Vehicle-Drone Arc Routing Approach Based on Improved Adaptive Large Neighborhood Search. *Sensors* **2022**, *22*, 3702. [CrossRef] [PubMed]
5. Campbell, J.F.; Corberán, A.; Plana, I.; Sanchis, J.M.; Segura, P. Solving the length constrained K-drones rural postman problem. *Eur. J. Oper. Res.* **2021**, *292*, 60–72. [CrossRef]
6. Liu, X.F.; Chung, S.H.; Kwon, C. An adaptive large neighborhood search method for the drone-truck arc routing problem. *Comput. Oper. Res.* **2025**, *176*, 106959. [CrossRef]
7. Jia, Y.H.; Mei, Y.; Zhang, M.J. Confidence-Based Ant Colony Optimization for Capacitated Electric Vehicle Routing Problem with Comparison of Different Encoding Schemes. *IEEE Trans. Evol. Comput.* **2022**, *26*, 1394–1408. [CrossRef]
8. Cao, F.; Cui, S.; Jing, Z.; Yao, B. Vehicle Route Planning Method for Cooperative Power Inspection by Inspection Vehicles and Drones. *J. Transp. Eng. Inf.* **2022**, *20*, 95–104. [CrossRef]
9. Guo-hua, W.; Ni, M.; Bin-jie, X.; Zhong-qiang, M.; Ke-xin, Z.; Zhi-meng, L. The cooperative delivery of multiple vehicles and multiple drones basedon adaptive large neighborhood. *Control Decis.* **2023**, *38*, 201–210. [CrossRef]
10. Changshi, L.; Zhang, W.; Yufeng, Z.; Liang, L.; Xiancheng, Z.; Xiaoliang, X. The Optimization of Dynamic and Cooperative Truck-Drone Distribution Routes for Emergency Supplies Delivery in Pandemic Areas. *J. Syst. Sci. Math. Sci.* **2022**, *42*, 3027–3043. [CrossRef]
11. Xu, B.; Zhao, K.; Luo, Q.; Wu, G.; Pedrycz, W. A GV-drone arc routing approach for urban traffic patrol by coordinating a ground vehicle and multiple drones. *Swarm Evol. Comput.* **2023**, *77*, 101246. [CrossRef]
12. Momeni, M.; Soleimani, H.; Shahparvari, S.; Afshar-Nadjafi, B. Coordinated routing system for fire detection by patrolling trucks with drones. *Int. J. Disaster Risk Reduct.* **2022**, *73*, 102859. [CrossRef]
13. Dienstknecht, M.; Boysen, N.; Briskorn, D. The traveling salesman problem with drone resupply. *OR Spectr.* **2022**, *44*, 1045–1086. [CrossRef]
14. Dell'Amico, M.; Montemanni, R.; Novellani, S. Algorithms based on branch and bound for the flying sidekick traveling salesman problem. *Omega* **2021**, *104*, 102493. [CrossRef]
15. Wang, Z.; Sheu, J.-B. Vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **2019**, *122*, 350–364. [CrossRef]
16. Zhen, L.; Gao, J.-J.; Tan, Z.-Y. Routing problem of multiple trucks and drones cooperative delivery. *J. Manag. Sci. China* **2024**, *27*, 1–18. [CrossRef]
17. Yin, Y.; Qing, L.; Wang, D.; Cheng, T.C.E.; Ignatius, J. Exact solution method for vehicle-and-drone cooperative delivery routing of blood products. *Comput. Oper. Res.* **2024**, *164*, 106559. [CrossRef]
18. Murray, C.C.; Chu, A.G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 86–109. [CrossRef]
19. Mulumba, T.; Diabat, A. Optimization of the drone-assisted pickup and delivery problem. *Transp. Res. Part E-Logist. Transp. Rev.* **2024**, *181*, 103377. [CrossRef]
20. Ramasamy, S.; Reddinger, J.P.F.; Dotterweich, J.M.; Childers, M.A.; Bhounsule, P.A. *Cooperative Route Planning of Multiple Fuel-Constrained Unmanned Aerial Vehicles with Recharging on an Unmanned Ground Vehicle*; IEEE: New York, NY, USA, 2021; pp. 155–164.
21. Saleu, R.G.M.; Deroussi, L.; Feillet, D.; Grangeon, N.; Quilliot, A. An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks* **2018**, *72*, 459–474. [CrossRef]
22. El-Adle, A.M.; Ghoniem, A.; Haouari, M. The cost of carrier consistency: Last-mile delivery by vehicle and drone for subscription-based orders. *J. Oper. Res. Soc.* **2024**, *75*, 821–840. [CrossRef]
23. Gu, Q.C.; Fan, T.J.; Han, W.K. Optimization of hybrid delivery by vehicle and drones. *Electron. Commer. Res. Appl.* **2024**, *66*, 101411. [CrossRef]
24. Wang, Y.; Duan, J.C.; Zhang, Q.; Ding, Y. A Three-Phase Heuristic for Optimizing Truck-Drone Collaborative Delivery. *Appl. Sci.* **2026**, *16*, 2016. [CrossRef]
25. Huachen, L.; Wen, Z.; Xudong, N.; Kangsheng, H.; Oianqian, W.; Zhibo, Z. Cooperative inspection Path Planning Method for Multi-Objective Joint Power inspection by Vehicle-Mounted UAV. *South. Power Syst. Technol.* **2023**, *17*, 138–147. [CrossRef]
26. de Freitas, J.C.; Penna, P.H.V. A Randomized Variable Neighborhood Descent Heuristic to Solve the Flying Sidekick Traveling Salesman Problem. *Electron. Notes Discret. Math.* **2018**, *66*, 95–102. [CrossRef]

27. Mara, S.T.W.; Sarker, R.; Essam, D.; Elsayed, S. Electric Vehicle-Drone Routing Problem with Optional Drone Availability. In *Proceedings of the IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 24–28 September 2023*; IEEE: New York, NY, USA, 2023; pp. 827–832.
28. Mara, S.T.W.; Sarker, R.; Essam, D.; Elsayed, S. An Adaptive Memetic Algorithm for a Cost-Optimal Electric Vehicle-Drone Routing Problem. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 19619–19632. [CrossRef]
29. Bogyrbayeva, A.; Dauletbayev, B.; Meraliyev, M. Reinforcement Learning for Efficient Drone-Assisted Vehicle Routing. *Appl. Sci.* **2025**, *15*, 2007. [CrossRef]
30. Chen, E.; Zhou, Z.; Li, R.; Chang, Z.; Shi, J. The multi-fleet delivery problem combined with trucks, tricycles, and drones for last-mile logistics efficiency requirements under multiple budget constraints. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *187*, 103573. [CrossRef]
31. Laporte, G.; Musmanno, R.; Vocaturo, F. An Adaptive Large Neighbourhood Search Heuristic for the Capacitated Arc-Routing Problem with Stochastic Demands. *Transp. Sci.* **2010**, *44*, 125–135. [CrossRef]
32. Mulumba, T.; Najy, W.; Diabat, A. The drone-assisted pickup and delivery problem: An adaptive large neighborhood search metaheuristic. *Comput. Oper. Res.* **2024**, *161*, 106435. [CrossRef]
33. Liu, Z.; Yu, B.; Chen, T.; Zhang, L. Integrating battery-related decisions into truck-drone tandem delivery problem with limited battery resources. *Transp. Res. Part C Emerg. Technol.* **2025**, *174*, 105082. [CrossRef]
34. Zhang, X.; Zeng, S. The drone-assisted simultaneous pickup and delivery problem with time windows. *Comput. Oper. Res.* **2025**, *178*, 106996. [CrossRef]
35. Meng, S.; Chen, Y.; Li, D. The multi-visit drone-assisted pickup and delivery problem with time windows. *Eur. J. Oper. Res.* **2024**, *314*, 685–702. [CrossRef]
36. Li, X.; Luo, H.; Wang, G.; Song, Z.; Gou, Q.; Meng, F. Optimizing multi-drone patrol path planning under uncertain flight duration: A robust model and adaptive large neighborhood search with simulated annealing. *Appl. Soft Comput.* **2025**, *176*, 113107. [CrossRef]
37. Lenstra, J.K.; Kan, A.H.G.R. On general routing problems. *Networks* **1976**, *6*, 273–280. [CrossRef]
38. Wang, J.; Yue, Y.; Yu, K.; Han, C.; Yang, J.; Zhang, L.; Zhang, Y.; Wang, Z.; Huang, Y. Thermal runaway behaviors of lithium iron phosphate battery with various capacity and state of charge: Characteristic comparison and safety assessment. *Appl. Therm. Eng.* **2026**, *284*, 129170. [CrossRef]
39. Chen, Y.; Zhu, M.; Chen, M. Comprehensive experimental research on wrapping materials influences on the thermal runaway of lithium-ion batteries. *Emerg. Manag. Sci. Technol.* **2025**, *5*, e007. [CrossRef]
40. Hansen, P.; Mladenović, N.; Todosijević, R.; Hanafi, S. Variable neighborhood search: Basics and variants. *EURO J. Comput. Optim.* **2017**, *5*, 423–454. [CrossRef]
41. Ropke, S.; Pisinger, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **2006**, *40*, 455–472. [CrossRef]
42. Campbell, J.F.; Corberán, Á.; Plana, I.; Sanchis, J.M. Drone arc routing problems. *Networks* **2018**, *72*, 543–559. [CrossRef]
43. Zeng, Y.; Xu, J.; Zhang, R. Energy Minimization for Wireless Communication with Rotary-Wing UAV. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2329–2345. [CrossRef]
44. Gao, N.; Zeng, Y.; Wang, J.; Wu, D.; Zhang, C.Y.; Song, Q.H.; Qian, A.C.; Jin, S. Energy Model for UAV Communications: Experimental Validation and Model Generalization. *China Commun.* **2021**, *18*, 253–264. [CrossRef]
45. Guzzella, L.; Sciarretta, A. *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*; Springer: Berlin/Heidelberg, Germany, 2007.
46. Wasserfall, D.; Botha, S.; Booysen, M.J. An Electric Vehicle Conversion for Rural Mobility in Sub-Saharan Africa. *Energies* **2025**, *18*, 6625. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.