

Article

Multi-Objective Dynamic Scheduling for Heterogeneous Emergency Fleets with Breakdown-Resilient Rescheduling

Zhuang Cai ^{1,*} and Cong Xiao ²¹ Jiangxi Provincial Key Laboratory of Intelligent Information Processing and Affective Computing, School of Artificial Intelligence, Jiangxi Normal University, Nanchang 330022, China² Financial Department, Jiangxi Normal University, Nanchang 330022, China; xiaocong@jxnu.edu.cn

* Correspondence: caizhuang@jxnu.edu.cn

Abstract

In post-disaster relief operations, emergency fleets typically consist of vehicles with varying load capacities, travel speeds, and operating costs. These heterogeneous vehicles are prone to unexpected breakdowns during delivery, which can severely disrupt supply chains and delay urgent aid. Existing scheduling approaches, however, rarely account for fleet heterogeneity, real-time breakdowns, and the trade-off between delivery speed and cost within a unified framework. This paper addresses this gap by formulating the dynamic scheduling of heterogeneous emergency fleets as a two-stage mixed-integer programming model, where total transportation time and cost are simultaneously minimized. The key algorithmic contribution is a fuzzy robust adaptive multi-objective hybrid algorithm (FR-AMOH) with three interconnected design components. First, a fuzzy evaluation-based pre-matching strategy uses entropy-weighted multi-criteria assessment to generate high-quality initial solutions. Second, a failure-resilient rescheduling module freezes system state upon breakdown detection and selects recovery plans via multi-dimensional resilience scoring to prevent cascading failures. Third, a Pareto-guided adaptive neighborhood search dynamically adjusts operator selection to balance time and cost optimization. Tests on 40 real-world instances with 50 to 1000 demand nodes show that FR-AMOH achieves optimal inverted generational distance values on 17 out of 40 instances, improves hypervolume by 15% to 35% on average compared with other metaheuristics, and keeps computation times between 40 and 250 s, which is within acceptable limits for emergency decision-making. FR-AMOH outperforms Gurobi and six leading metaheuristics in solution quality with comparable computational cost.

Keywords: emergency fleet scheduling; heterogeneous vehicles; breakdown resilience; multi-objective optimization; dynamic rescheduling



MSC: 68W50

Academic Editor: Frank Werner

Received: 15 June 2026

Revised: 2 July 2026

Accepted: 5 July 2026

Published: 14 July 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Delivering emergency supplies to disaster-affected areas in a timely and cost-effective manner is one of the most pressing logistical challenges in post-disaster response [1]. At the heart of this challenge lies the vehicle routing problem (VRP), which directly governs how quickly relief supplies reach those in need and how efficiently limited resources are deployed [1,2]. Post-disaster environments face multiple concurrent disruptions. Vehicle breakdowns are especially damaging because they directly reduce transport capacity in an

already strained system. By contrast, road interruptions can often be bypassed via detours, demand surges can be partially absorbed through load reallocation, and communication delays mainly affect coordination rather than physical delivery. When a vehicle breaks down mid-route, its cargo must be redistributed among the remaining fleet, creating a cascading burden that worsens the impact of other disaster-related disruptions. In practice, emergency fleets are rarely homogeneous; they usually include three categories of vehicles with distinct roles in rescue operations. Heavy-duty trucks (10–20 ton capacity) carry bulk supplies from depots to distribution points, medium vans (3–8 ton capacity) handle flexible regional delivery, and light specialized vehicles (1–2 ton capacity) such as four-wheel-drive ambulances and small utility vehicles can traverse damaged roads to reach isolated communities. Each type differs significantly in load capacity, fuel range, speed, and per-kilometer operating cost. When a mixed fleet operates under the harsh, unpredictable conditions of a disaster zone, including flooded roads, debris-blocked routes, and mechanical failures under heavy use, vehicle breakdowns occur frequently and are difficult to avoid. Without prompt rescheduling of the remaining fleet, supply interruptions cascade through the network, leaving affected communities without essential aid and undermining the entire rescue effort [3,4]. Related research on disruption recovery in logistics and networked systems has also examined resilience to failures, providing useful theoretical perspectives on maintaining service continuity under unexpected disruptions [5–7].

Many studies have investigated emergency vehicle routing from both modeling and algorithmic perspectives. On the modeling side, mixed-integer programming (MIP) is widely used to capture problem characteristics across various disaster types. Representative efforts include robust multi-period models for resource allocation under demand uncertainty [2], scenario-based recovery formulations for delivery disruptions [5], drone-truck collaborative logistics for flood response [1], and heterogeneous fleet models with time-window constraints [3,8]. On the algorithmic side, metaheuristics have become the dominant solution approach. Parallel genetic algorithms hybridized with variable neighborhood search for green and emergency routing [4,9], adaptive large neighborhood search (ALNS) for multi-objective routing [10,11], and improved multi-objective evolutionary algorithms [12,13] all show that heuristic strategies produce high-quality solutions within reasonable computation times.

However, several gaps remain. In terms of modeling, many studies optimize only a single objective, typically cost, while neglecting the equally important dimension of delivery timeliness. We recognize that life-saving priority and service equity are paramount in emergency relief. Even so, treating transportation cost as an equal objective alongside time is warranted for three practical reasons. First, post-disaster relief operations often extend over weeks or months, and sustained delivery capacity depends on maintaining operational budgets; focusing too heavily on speed alone can quickly deplete limited resources, making it harder to serve all affected communities fairly over the course of relief operations. Second, cost minimization naturally promotes efficient route design, which cuts unnecessary travel and indirectly shortens response times by freeing vehicles for additional trips. Third, emergency managers often need to balance speed against resource consumption, and a bi-objective framework provides the Pareto front needed to make context-appropriate choices instead of forcing a single solution that may not work well in all situations. Those that do address breakdowns tend to focus on ordinary logistics settings, where the urgency and operational constraints of emergency scenarios are not fully captured. Most importantly, no existing work integrates heterogeneous fleet characteristics, dynamic breakdown handling, bi-objective optimization, and reliable rescheduling into a coherent two-stage framework. The two-stage structure of initial planning followed by post-breakdown rescheduling still works when disaster environments evolve continuously,

because it works on a rolling horizon. The initial plan provides a baseline schedule that is continuously monitored. Whenever a breakdown occurs, the system state is frozen and a new rescheduling problem is formulated with the updated vehicle positions, remaining cargo, and unserved demands. This event-driven approach readily handles continuous evolution by treating each state change as a potential rescheduling trigger, rather than trying to predict all future disturbances in advance. On the algorithmic side, classical multi-objective algorithms such as the non-dominated sorting genetic algorithm II (NSGA-II) [14] and the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [15] often have problems with poor initial solution quality and slow convergence, while the local search components of improved variants cannot adapt well enough to handle the dual objectives of time and cost simultaneously. Demand nodes in real post-disaster scenarios vary not only in quantity demanded but also in supply category requirements, affected population size, and rescue priority. Our model incorporates priority weights in the pre-matching stage (criterion I_4) to ensure that high-priority nodes receive earlier service, while demand quantity itself directly drives load allocation across vehicles. Supply category differentiation is handled implicitly through the heterogeneous fleet composition, where different vehicle types are assigned to transport different categories of relief materials based on their capacity and accessibility characteristics.

This paper addresses these gaps with an integrated modeling and algorithmic framework. We formally define the multi-objective dynamic scheduling problem for heterogeneous emergency fleets subject to real-time vehicle breakdowns (DHEVRP-VB) and build a two-stage MIP model that minimizes total transportation time and cost. On the algorithmic side, we develop a tailored solution method, the fuzzy robust adaptive multi-objective hybrid algorithm (FR-AMOH), which addresses the shortcomings of existing approaches through three coordinated mechanisms, namely high-quality initial population generation, failure-resilient dynamic rescheduling, and adaptive multi-neighborhood local search.

The main contributions of this paper are as follows.

- A formal definition of DHEVRP-VB together with a two-stage MIP model that captures heterogeneous vehicle attributes (load capacity, cruising range, speed, cost), real-time breakdown detection, and the joint minimization of transportation time and cost. The model provides a useful decision-making framework for post-disaster fleet operations that directly considers the interplay between delivery urgency and resource sustainability.
- The design of FR-AMOH, which incorporates a heterogeneous vehicle-demand node pre-matching mechanism (HVDNPM). By combining entropy-weighted objective assessment with fuzzy comprehensive evaluation, HVDNPM generates initial populations that are both task-adaptive and diverse, significantly accelerating convergence compared with random initialization. Fuzzy evaluation is chosen over greedy construction or regret-based heuristics because it handles the multi-criteria character of vehicle–node matching, where load capacity, speed, range, and priority must be balanced at the same time. Greedy approaches often over-optimize for a single criterion and produce short-sighted assignments, while regret-based methods cannot easily handle the fuzzy boundaries between good and poor matches in heterogeneous fleet contexts.
- Two further algorithmic enhancements. (a) A resilience-oriented rescheduling mechanism (ROM-MDR) activates upon breakdown detection, reconstructs the scheduling problem in real time, and evaluates candidate plans using multi-dimensional resilience metrics to guard against cascading failures. (b) A Pareto-dominance-based adaptive local search (ALM-PDR) dynamically selects among four complementary neighborhood operators according to their historical contribution to each objective, which balances

exploration and exploitation. These three modules are designed to address three key challenges of the problem: poor initial solution quality from naive assignment, vulnerability to cascading failures after breakdowns, and the difficulty of optimizing two conflicting objectives at the same time with a single fixed neighborhood structure.

The remainder of this paper is organized as follows. Section 2 presents the problem formulation and reviews related work. Section 3 details the FR-AMOHFA framework and its components. Section 4 reports computational experiments on 40 real-world instances. Section 5 concludes the paper and outlines future research directions.

2. Problem Definition and Related Work

This section formalizes the DHEVRP-VB and then surveys the relevant literature on mathematical modeling and solution algorithms for emergency vehicle routing.

2.1. Problem Definition

The DHEVRP-VB captures a realistic post-disaster relief scenario in which a fleet of heterogeneous vehicles, differing in load capacity, cruising range, travel speed, and unit cost, must deliver supplies from a depot to multiple affected locations. Under normal conditions, the task is to plan routes that minimize total transportation time and cost simultaneously (Figure 1). The situation becomes more complex when one or more vehicles break down during operation. The remaining fleet must be rescheduled promptly to take over unfinished deliveries, while still respecting vehicle capacity constraints and the urgency of emergency service. In this context, resilience describes the ability of the distribution system to maintain continuous delivery service and withstand vehicle breakdowns without triggering widespread supply disruptions. From an operational perspective, a resilient schedule provides sufficient redundancy in vehicle load capacity, driving range, and time buffers to handle unexpected disruptions and smoothly reallocate tasks among the remaining vehicles. The practical value of resilience appears in rescue outcomes: shorter service delays for affected populations, fewer abandoned delivery tasks, and consistent completion rates even when multiple vehicles fail during operations. This problem combines an initial planning stage with a real-time reactive stage, requiring solutions that are both efficient and resilient to operational disruptions.

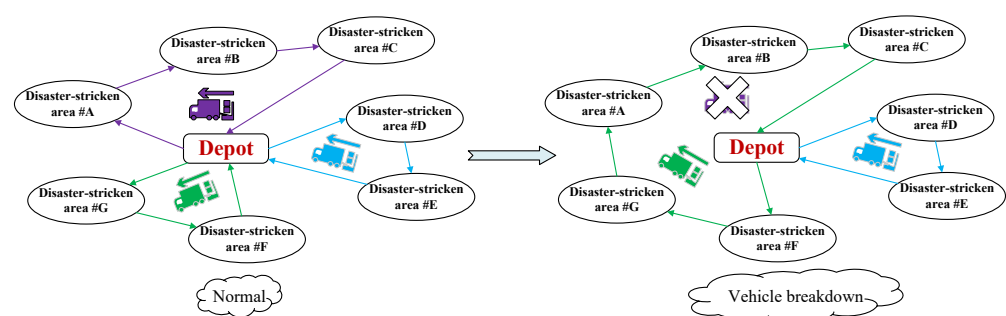


Figure 1. A schematic diagram of DHEVRP-VB.

To precisely define the DHEVRP-VB, we introduce the following notation.

2.1.1. Sets

- N denotes the set of all nodes, where $N = \{0\} \cup C$, where 0 represents the emergency material depot, and $C = \{1, 2, \dots, n_C\}$ denotes the set of affected points, with $n_C = |C|$ being the number of affected points.

- V denotes the set of heterogeneous emergency vehicles, $V = \{1, 2, \dots, n_V\}$, where $n_V = |V|$ is the total number of vehicles. Vehicles are classified by attributes, denoted by the set $K = \{1, 2, \dots, n_K\}$, with $n_K = |K|$ being the number of vehicle types.
- T denotes the set of time periods, used to characterize the temporal randomness of breakdowns, where $T = \{1, 2, \dots, n_T\}$ and n_T is the maximum number of time steps.
- E denotes the set of feasible paths between nodes, $E = \{(i, j) | i, j \in N, i \neq j\}$, indicating that vehicles can travel from node i to node j .

2.1.2. Parameters

- q_j denotes the material demand of affected point j , $j \in C$.
- Q_v denotes the maximum load capacity of vehicle v , $v \in V$ (a key attribute of heterogeneous vehicles, with different vehicle types having different load capacities).
- R_v denotes the maximum cruising range of vehicle v , $v \in V$ (a key attribute of heterogeneous vehicles, with different vehicle types having different cruising ranges).
- s_v denotes the average travel speed of vehicle v , $v \in V$ (a key attribute of heterogeneous vehicles, with different vehicle types having different speeds).
- c_{ijv} denotes the unit transportation cost for vehicle v to travel from node i to node j , $v \in V, (i, j) \in E$.
- d_{ij} denotes the distance between node i and node j , $(i, j) \in E$.
- $t_{tr,ijv}$ denotes the travel time for vehicle v to travel from node i to node j , $t_{tr,ijv} = d_{ij}/s_v$, $v \in V, (i, j) \in E$.
- $t_{un,j}$ denotes the unloading time of materials at affected point j , $j \in C$.
- α_{vt} is a 0-1 parameter: $\alpha_{vt} = 1$ if vehicle v breaks down at time t , and $\alpha_{vt} = 0$ if vehicle v operates normally at time t , $v \in V, t \in T$.
- M denotes a sufficiently large positive number, used for constraint relaxation.

2.1.3. Decision Variables

- x_{ijvt} is a 0-1 variable: $x_{ijvt} = 1$ if vehicle v travels from node i to node j at time t ; otherwise, $x_{ijvt} = 0$, $v \in V, (i, j) \in E, t \in T$.
- y_{jvt} is a 0-1 variable: $y_{jvt} = 1$ if vehicle v delivers materials to affected point j at time t ; otherwise, $y_{jvt} = 0$, $v \in V, j \in C, t \in T$.
- z_{vt} is a 0-1 variable: $z_{vt} = 1$ if vehicle v participates in delivery (operates normally and is not faulty) at time t ; otherwise, $z_{vt} = 0$, $v \in V, t \in T$.
- w_{vjt} is a continuous variable representing the quantity of materials delivered by vehicle v to affected point j at time t , $v \in V, j \in C, t \in T$.
- u_{vt} is a continuous variable representing the remaining load capacity of vehicle v at time t , $v \in V, t \in T$.
- l_{vt} is a continuous variable representing the remaining cruising range of vehicle v at time t , $v \in V, t \in T$.
- TT denotes the total transportation time (objective function variable).
- TC denotes the total transportation cost (objective function variable).

2.1.4. Objective Functions

This is a bi-objective optimization problem in which both total transportation time (TT) and total transportation cost (TC) are minimized.

Objective function 1 (minimizing total transportation time).

$$\min TT = \sum_{v \in V} \sum_{t \in T} \sum_{(i,j) \in E} x_{ijvt} \cdot t_{tr,ijv} + \sum_{v \in V} \sum_{t \in T} \sum_{j \in C} y_{jvt} \cdot t_{un,j} \quad (1)$$

Note. Total transportation time comprises two components: vehicle travel time between nodes and unloading time at each affected point.

Objective function 2 (minimizing total transportation cost).

$$\min TC = \sum_{v \in V} \sum_{t \in T} \sum_{(i,j) \in E} x_{ijvt} \cdot c_{ijv} \cdot d_{ij} \quad (2)$$

Note. Total transportation cost aggregates the product of per-unit-distance cost and travel distance over all vehicle movements, directly reflecting fleet heterogeneity since different vehicle types incur different unit costs.

The model is subject to the following constraints.

Demand satisfaction. Every affected point must receive its full material demand, and each point is served by exactly one vehicle to avoid redundant deliveries in time-sensitive scenarios,

$$\begin{cases} \sum_{v \in V} \sum_{t \in T} w_{vjt} = q_j, & \forall j \in C \\ \sum_{v \in V} \sum_{t \in T} y_{jvt} = 1, & \forall j \in C \end{cases} \quad (3)$$

Load capacity. The actual cargo on any vehicle must not exceed its rated capacity at any time, and the remaining capacity must suffice for subsequent deliveries,

$$\begin{cases} u_{vt} = Q_v - \sum_{j \in C} w_{vjt}, & \forall v \in V, t \in T \\ u_{vt} \geq 0, & \forall v \in V, t \in T \end{cases} \quad (4)$$

Cruising range. A vehicle's remaining range must at all times cover its planned travel distance, preventing mid-route fuel or battery exhaustion,

$$\begin{cases} l_{vt} = R_v - \sum_{(i,j) \in E} x_{ijvt} \cdot d_{ij}, & \forall v \in V, t \in T \\ l_{vt} \geq 0, & \forall v \in V, t \in T \end{cases} \quad (5)$$

Breakdown exclusion. A broken-down vehicle is immediately withdrawn from service, and its fault state and delivery state are mutually exclusive,

$$\begin{cases} z_{vt} + \alpha_{vt} \leq 1, & \forall v \in V, t \in T \\ \sum_{(i,j) \in E} x_{ijvt} \leq z_{vt} \cdot M, & \forall v \in V, t \in T \\ \sum_{j \in C} y_{jvt} \leq z_{vt} \cdot M, & \forall v \in V, t \in T \end{cases} \quad (6)$$

Path continuity. Any vehicle arriving at a node must subsequently depart from that node (or return to the depot), ensuring no route segment is left dangling,

$$\sum_{j \in N, j \neq i} x_{ijvt} = \sum_{j \in N, j \neq i} x_{jivt}, \quad \forall v \in V, i \in N, t \in T \quad (7)$$

Delivery-path coupling. A vehicle can deliver materials to an affected point only if it actually visits that point, so delivery actions are thus tied to the physical route,

$$y_{jvt} \leq \sum_{i \in N, i \neq j} x_{ijvt}, \quad \forall v \in V, j \in C, t \in T \quad (8)$$

The binary variables $x_{ijvt}, y_{jvt}, z_{vt} \in \{0, 1\}$ and continuous variables $w_{vjt}, u_{vt}, l_{vt} \geq 0$ follow their respective domain definitions given above.

The model follows a two-stage structure. In the first stage (initial planning), routes are planned for the entire fleet under the assumption that all vehicles operate normally.

This stage covers material allocation, node assignment, and route sequencing. In the second stage (reactive rescheduling), a breakdown is triggered whenever $\alpha_{vt} = 1$, meaning vehicle v fails at time t . The failed vehicle is then taken out of service by setting $z_{vt} = 0$, and the remaining operational vehicles ($z_{vt} = 1$) are re-routed to complete any outstanding deliveries, ensuring that no affected point goes unserved.

2.2. Related Work

We review two bodies of work most closely related to this paper: mathematical modeling of emergency vehicle routing, and the algorithms developed to solve such problems.

(1) Problem modeling. Emergency vehicle routing has been studied extensively across various disaster contexts, including floods, earthquakes, wildfires, pandemics, and droughts [16]. Formulations range from two-echelon logistics networks under urban pandemic lockdowns [17] and fire engine scheduling for wildfire suppression [12] to post-earthquake road damage scenarios [4], truck–drone collaboration in flood zones [1], large-scale drought relief water distribution [16], and emergency vehicle priority coordination in connected vehicle environments [18]. Distributionally robust and scenario-based approaches handle demand and travel time uncertainty in post-disaster relief distribution [2], and researchers have also developed robust optimization frameworks for truck–drone humanitarian logistics under travel time uncertainty [12]. Most of these studies use mixed-integer programming frameworks to capture scenario-specific features, such as road network accessibility, demand uncertainty, and transport capacity. Beyond standard routing problems, researchers have studied more complex settings, including multi-depot multi-trip UAV operations for post-disaster surveying [19], two-echelon truck–drone delivery with communication recovery [20], integrated location-routing with inter-depot transshipment [21], and coupled UAV-ground vehicle scheduling [22]. Researchers have also incorporated emergency-specific constraints, such as soft and hard time windows and service priority rules [3,18], along with integrated production-distribution models for public health emergencies [8] and multi-depot heterogeneous fleet formulations with adaptive large neighborhood search [10].

More recent modeling work has focused on dynamic and robust formulations. Researchers have developed disruption recovery models for vehicle breakdowns, road congestion, and node failures [5,6], while distributionally robust and scenario-based approaches handle uncertain demand and travel times [2,23,24]. Multi-objective models now go beyond cost minimization to include delivery timeliness, service equity, and loss reduction [1,21], and heterogeneous fleet attributes, such as load capacity, cruising range, and operating cost, are increasingly reflected in objective formulations [3,10,12]. However, how breakdown-induced uncertainty interacts with routing decisions has received less attention [25].

(2) Solving algorithms. Multi-objective VRPs are NP-hard, and most solution approaches in the literature are based on metaheuristics [10,14,15,26,27]. Early bi-objective work used scatter search to approximate Pareto fronts for small and medium instances [28]. Researchers have since combined classical scalarization techniques, such as weighted sum and ϵ -constraint methods, with ALNS, variable neighborhood search, and simulated annealing, applying them to problems from green logistics to multi-period perishable goods delivery [9,11,29]. Path relinking, multidirectional local search, and ϵ -dominance mechanisms help maintain population diversity while exploiting local neighborhoods [12,30–32], and 3D k -means clustering with multi-thread parallel computing has improved scalability for large instances.

More recent algorithmic work has explored data-driven and adaptive approaches. Evolutionary algorithms enhanced with machine learning use historical routing data to capture decision-maker preferences and time-dependent traffic patterns, leading to better

solution quality and faster convergence [13,33,34]. For heavily constrained problems, infeasibility-guided strategies use information from infeasible solutions to broaden the search and handle complex constraints such as dual time windows and hierarchical demand priorities [35,36]. Hyper-heuristic memetic frameworks and MIP-metaheuristic hybrids have been applied to truck–drone coordination and post-disaster humanitarian logistics, where they balance competing objectives effectively under dynamic uncertainty [37,38].

(3) Summary. Although researchers have made significant progress in both modeling and algorithm design for emergency vehicle routing, two important gaps remain. As summarized in Table 1, on the modeling side, no existing framework simultaneously integrates heterogeneous fleet adaptation, dynamic breakdown handling, bi-objective optimization, and robust rescheduling. These four components are all necessary for realistic post-disaster operations. On the algorithmic side, current multi-objective methods can approximate Pareto fronts but have three key limitations. Specifically, they do not adequately ensure solution robustness under breakdown conditions, they often produce low-quality initial populations, and they struggle to scale to large instances while maintaining convergence speed.

Table 1. Positioning of this work relative to representative emergency VRP studies. A checkmark (✓) indicates that the corresponding feature is addressed by the study, while a blank cell indicates that the feature is not considered. The last row (this work) is highlighted in bold to distinguish the proposed framework from existing literature.

Reference	Heterogeneous Fleet	Breakdown Handling	Bi-Objective	Robust Rescheduling	Adaptive Local Search	Real-World Instances
Lu et al. [1]	✓		✓			✓
Wang et al. [2]				✓		✓
Liu et al. [3]	✓					✓
Zhao et al. [4]		✓				✓
Vieira et al. [16]						✓
Zhang et al. [12]	✓		✓		✓	✓
Calamoneri et al. [19]						✓
Wei et al. [21]						✓
Hu et al. [5]		✓				✓
Mancini [10]	✓				✓	✓
This work	✓	✓	✓	✓	✓	✓

This paper addresses these gaps. We first formally define the DHEVRP-VB and develop a two-stage MIP model that jointly minimizes transportation time and cost. We then present FR-AMOH, a tailored algorithm that integrates three complementary mechanisms, described in detail in the next section.

3. Proposed FR-AMOH

This section presents the overall framework of FR-AMOH and its three core mechanisms, followed by an analysis of time complexity.

3.1. Overall Framework of FR-AMOH

FR-AMOH integrates bi-objective optimization and resilient rescheduling into a four-phase pipeline that progresses from initial population generation to final solution output.

Phase 1. Initial population generation. Emergency demand nodes vary widely in urgency, and fleet vehicles differ across multiple attributes. Random or naive greedy initialization tends to produce poor-quality starting populations that slow convergence and trap the search in local optima. To overcome this, a heterogeneous vehicle-demand node pre-matching mechanism (HVDNPM) employs fuzzy comprehensive evaluation with entropy-

weighted objective scoring to quantify how well each vehicle-task pairing fits, yielding an initial population that is both task-adaptive and diverse.

Phase 2. Bi-objective evolutionary optimization. Building on the hybrid genetic search (HGS) framework [39,40], this phase performs selection, crossover, and mutation, augmented by an adaptive multi-neighborhood local search (ALM-PDR). The search direction is steered by evaluating each neighborhood operator's historical contribution to the two objectives, and the Pareto dominance criterion balances global exploration with local exploitation to efficiently approximate the Pareto front.

Phase 3. Dynamic breakdown rescheduling. The algorithm continuously monitors vehicle status. When a breakdown is detected, a robustness-oriented rescheduling mechanism (ROM-MDR) activates. The current system state is frozen, the remaining scheduling problem is reconstructed, and candidate recovery plans are scored by a multi-dimensional robustness metric. Plans with the highest robustness are retained, ensuring stable post-disaster distribution even under cascading failures.

Phase 4. Termination and output. Once the maximum iteration count or runtime budget is reached, the algorithm terminates and outputs the final Pareto-optimal solution set.

The complete pseudocode of the algorithm is given in Algorithm 1, and the overall workflow is illustrated in Figure 2.

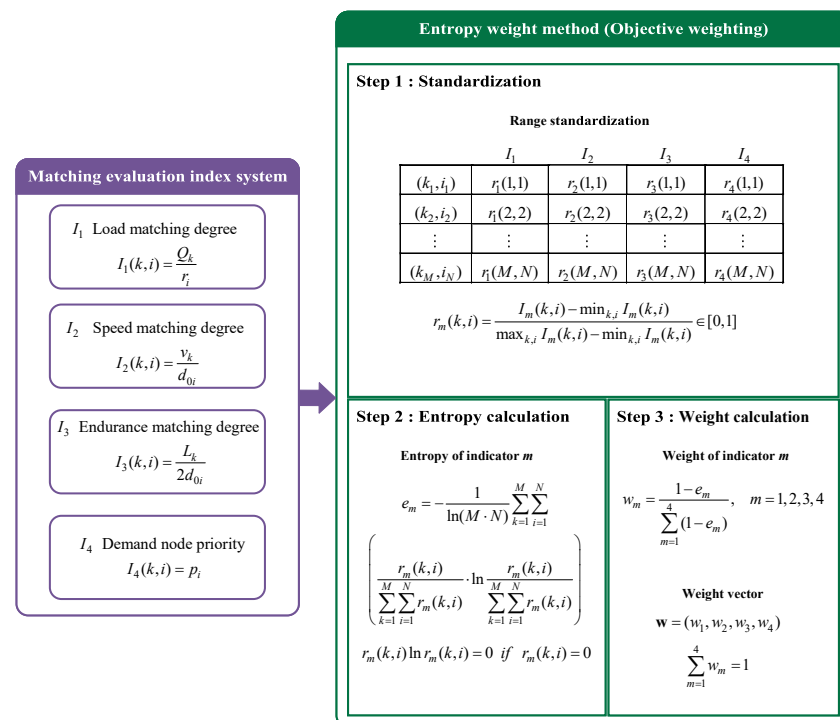


Figure 2. Weight calculation steps.

3.2. HVDNPM

In a typical post-disaster setting, some demand nodes are far more urgent than others, while the available fleet includes vehicles with widely differing payloads, speeds, ranges, and costs. Randomly generated or simply greedy initial solutions rarely capture these differences, often leading to slow convergence and premature stagnation. HVDNPM addresses this by combining fuzzy comprehensive evaluation with entropy-weighted scoring to pre-match vehicles to demand nodes in a principled way, producing a high-quality yet diverse initial population.

Algorithm 1: Pseudocode of the FR-AMOA

Input: Parameters of the DHEVRP-VB include the set of demand nodes C , vehicle type set K , demand matrix, time windows, vehicle breakdown monitoring threshold, maximum iterations G , population size P , and initial temperature T_0 .

Output: Final Pareto optimal solution set Ω_{final} .

```

/* Phase 1 initial population generation */;
1 Construct a 4-dimensional evaluation index system for vehicle-demand node matching pairs;
2 Calculate the objective weight vector  $\omega$  of each index using the entropy weight method;
3 Compute the comprehensive matching degree score  $S(k, i)$  for all matching pairs based on fuzzy comprehensive evaluation;
4 Generate the initial population  $P_0$  via greedy pre-matching with random perturbation, and initialize the iteration counter  $g = 0$ ;
5 Initialize the neighborhood objective contribution degree  $\Delta_m(\mathcal{N}_i) = 0$ , and set the simulated annealing temperature  $\mathcal{T} = T_0$ ;
/* Phase 2 main iterative optimization loop */;
6 while  $g < G$  do
    /* Genetic operations */;
    7 Select parent individuals from the current population  $P_g$  using tournament selection;
    8 Generate the offspring population  $O_1$  via the order crossover operator;
    9 Perform inversion mutation on  $O_1$  to generate the offspring population  $O_2$ ;
    /* Adaptive multi-neighborhood local search */;
    10 for each individual  $X \in O_2$  do
        11 Calculate the normalized objective values and objective deviation degree  $D_m(X)$  of the current solution  $X$ ;
        12 Select the neighborhood operator  $\mathcal{N}_i$  according to the adaptive selection probability  $p(\mathcal{N}_i)$ ;
        13 Execute the neighborhood operator  $\mathcal{N}_i$  to generate a new solution  $X'$ ;
        14 Judge whether to accept  $X'$  based on the Pareto dominance acceptance criterion;
        15 Update the neighborhood objective contribution degree  $\Delta_m(\mathcal{N}_i)$ ;
    16 end
    /* Population update */;
    17 Merge the parent population  $P_g$  and the offspring population  $O_2$  to form the combined population  $U_g$ ;
    18 Perform non-dominated sorting and crowding distance calculation on  $U_g$ ;
    19 Select the top  $P$  individuals to generate the new generation population  $P_{g+1}$ ;
    /* Temperature decay */;
    20  $\mathcal{T} = \mathcal{T} \times \gamma$ ;
    /* Phase 3 Dynamic breakdown monitoring and rescheduling */;
    21 if a vehicle breakdown event is detected then
        22 Freeze the current system state, and extract the set of available vehicles, set of uncompleted tasks, and real-time vehicle status;
        23 Reconstruct the rescheduling optimization problem, and perform iterative optimization with the current population as the initial solution;
        24 Generate the rescheduling Pareto solution set  $\Omega_{\text{reschedule}}$ ;
        25 Calculate the comprehensive robustness metric  $R_{\text{total}}(X)$  for all schemes in  $\Omega_{\text{reschedule}}$ ;
        26 Select the top 30% high-robustness schemes to update the current execution plan and population  $P_{g+1}$ ;
    27 end
    28  $g = g + 1$ ;
29 end
/* Phase 4 Termination and output */;
30 Perform non-dominated sorting on the final population  $P_G$ , and extract the Pareto optimal solution set  $\Omega_{\text{final}}$ ;
31 return  $\Omega_{\text{final}}$ 

```

For any vehicle type $k \in K$ and demand node $i \in C$, we define four benefit-type evaluation criteria covering capacity adaptability and emergency priority. The load matching degree $I_1(k, i) = Q_k/q_i$ measures how well the vehicle's rated load covers the node's demand. The speed matching degree $I_2(k, i) = s_k/d_{0i}$ reflects the vehicle's time-efficiency potential, where d_{0i} is the depot-to-node distance. The endurance matching degree $I_3(k, i) = R_k/(2d_{0i})$ checks whether the vehicle's maximum range can sustain a round

trip. The demand node priority $I_4(k, i) = p_i$ encodes the urgency of node i based on factors such as casualty severity and affected population, independent of vehicle attributes, where p_i denotes the emergency priority weight of node i .

To avoid the bias inherent in subjective weighting, the entropy weight method is used to derive objective weights for the four criteria in three steps (Figure 2). First, range standardization eliminates dimensional effects across all matching pairs.

$$r_m(k, i) = \frac{I_m(k, i) - \min_{k,i} I_m(k, i)}{\max_{k,i} I_m(k, i) - \min_{k,i} I_m(k, i)}, \quad m = 1, 2, 3, 4 \quad (9)$$

where $r_m(k, i) \in [0, 1]$ is the standardized value of criterion m for matching pair (k, i) . Second, the information entropy of each criterion is computed. Lower entropy means higher discriminative power and therefore a larger weight.

$$e_m = -\frac{1}{\ln(n_K \cdot n_C)} \sum_{k=1}^{n_K} \sum_{i=1}^{n_C} \left(\frac{r_m(k, i)}{\sum_{k=1}^{n_K} \sum_{i=1}^{n_C} r_m(k, i)} \cdot \ln \frac{r_m(k, i)}{\sum_{k=1}^{n_K} \sum_{i=1}^{n_C} r_m(k, i)} \right) \quad (10)$$

By convention, $r_m(k, i) \ln r_m(k, i) = 0$ when $r_m(k, i) = 0$. Third, the objective weight of each criterion is derived directly from its entropy.

$$\omega_m = \frac{1 - e_m}{\sum_{m=1}^4 (1 - e_m)}, \quad m = 1, 2, 3, 4 \quad (11)$$

where ω_m is the weight of criterion m , and $1 - e_m$ represents its utility, where higher values indicate greater contribution to the matching decision.

With the weights in hand, fuzzy comprehensive evaluation quantifies the overall matching quality of each vehicle–node pair. Standardized criterion values are mapped to four fuzzy levels, namely, “Excellent”, “Good”, “Medium”, and “Poor”, using trapezoidal membership functions. For the “Excellent” level, the membership function is given below.

$$\mu_{\text{Excellent}}(r) = \begin{cases} 0, & r < 0.6 \\ \frac{r-0.6}{0.2}, & 0.6 \leq r < 0.8 \\ 1, & r \geq 0.8 \end{cases} \quad (12)$$

The remaining three membership functions (μ_{Good} , μ_{Medium} , μ_{Poor}) follow analogous definitions with thresholds adjustable via sensitivity analysis. Each matching pair (k, i) then yields a 4×4 fuzzy evaluation matrix $\mathbf{E}(k, i)$, whose (m, l) entry is the membership value of criterion m at fuzzy level l . Combining $\mathbf{E}(k, i)$ with the weight vector $\omega = (\omega_1, \omega_2, \omega_3, \omega_4)$ through the weighted average fuzzy synthesis operator produces the comprehensive evaluation vector as shown below.

$$\begin{aligned} \mathbf{B}(k, i) &= \omega \circ \mathbf{E}(k, i) \\ &= \left(\sum_{m=1}^4 \omega_m \cdot \mu_{\text{Excellent}}(r_m(k, i)), \sum_{m=1}^4 \omega_m \cdot \mu_{\text{Good}}(r_m(k, i)), \right. \\ &\quad \left. \sum_{m=1}^4 \omega_m \cdot \mu_{\text{Medium}}(r_m(k, i)), \sum_{m=1}^4 \omega_m \cdot \mu_{\text{Poor}}(r_m(k, i)) \right) \end{aligned} \quad (13)$$

A scalar matching score $S(k, i)$ is obtained by weighting the four components of $\mathbf{B}(k, i)$ with level weights 0.9, 0.7, 0.5, and 0.3, as shown below.

$$S(k, i) = 0.9 \cdot B_1(k, i) + 0.7 \cdot B_2(k, i) + 0.5 \cdot B_3(k, i) + 0.3 \cdot B_4(k, i) \quad (14)$$

where $B_1(k, i)$ through $B_4(k, i)$ are the four components of $\mathbf{B}(k, i)$, and $S(k, i) \in [0.3, 0.9]$, where higher values indicate a better vehicle–node fit.

The initial population is then assembled by greedy pre-matching with random perturbation. All matching pairs are ranked by descending $S(k, i)$ so that high-priority demand nodes are served first. For each node in turn, the highest-scoring available vehicle is assigned subject to load and range feasibility checks. Finally, 10–20% random perturbation is added to the pre-matched solutions to prevent population homogeneity and preserve global search capability.

3.3. ROM-MDR

Post-disaster environments are naturally volatile. Beyond the initial breakdown, secondary failures, road damage, and aftershocks can further destabilize an already fragile distribution plan. Conventional rescheduling methods optimize only for the immediate time and cost, yielding plans that look optimal on paper but collapse at the first subsequent disturbance. ROM-MDR tackles this by embedding a multi-dimensional robustness metric directly into the bi-objective rescheduling process, filtering out plans that lack resilience without altering the original bi-objective structure.

Three robustness metrics, each capturing a different dimension of operational resilience, are constructed to evaluate how well a routing scheme withstands subsequent disturbances (Figure 3).

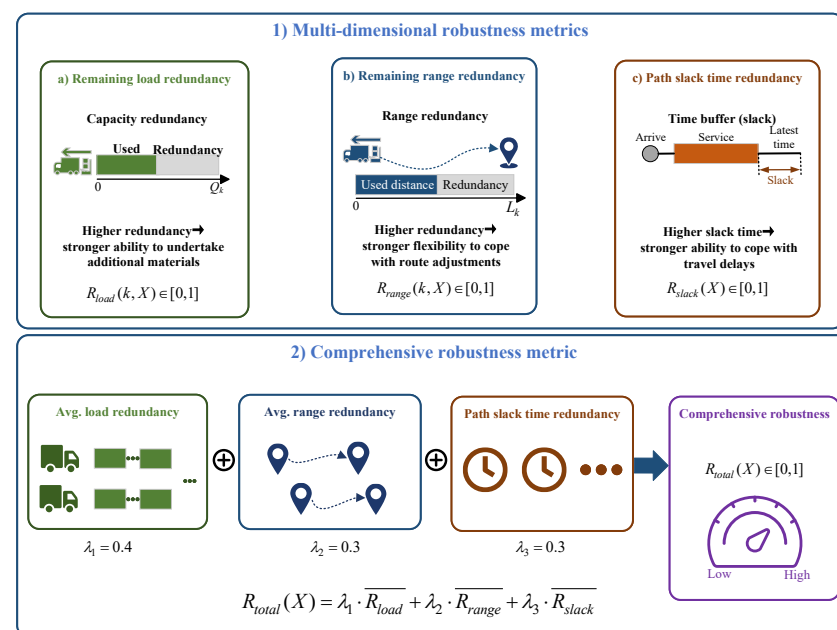


Figure 3. Three robustness metrics.

The remaining load redundancy $R_{load}(v, X)$ measures how much spare payload capacity vehicle v retains under plan X . Higher redundancy means the vehicle can absorb demand surges or take over cargo from a failed peer. For available vehicle set V_a and plan X , this metric is defined below.

$$R_{load}(v, X) = \frac{Q_v - \max_{(i,j) \in X} f_{ij}^v}{Q_v} \quad (15)$$

where $\max_{(i,j) \in X} f_{ij}^v$ is the peak on-board load (cargo flow) of vehicle v along its route in plan X , with $R_{load}(v, X) \in [0, 1]$.

The remaining range redundancy $R_{range}(v, X)$ measures the vehicle's ability to accommodate detours caused by road damage or route changes, and is computed as follows.

$$R_{range}(v, X) = \frac{R_v - \sum_{(i,j) \in X} d_{ij} x_{ij}^v}{R_v} \quad (16)$$

where $\sum_{(i,j) \in X} d_{ij} x_{ij}^v$ is the planned travel distance of vehicle v in plan X , with $R_{range}(v, X) \in [0, 1]$.

The path slack time redundancy $R_{slack}(X)$ quantifies the temporal buffer available to absorb delays from congestion or service overruns. For each demand node i , the individual slack is computed as follows.

$$Slack_i(X) = LT_i - (T_i^v + \tau_i) \quad (17)$$

where LT_i is the latest allowable service time, T_i^v is the arrival time of vehicle v at node i , and τ_i is the service duration at node i . The aggregate slack redundancy is then computed as follows.

$$R_{slack}(X) = \frac{1}{n_{Ca}} \sum_{i \in C_a} \frac{\max(Slack_i(X), 0)}{\max_{X' \in \Omega} \max(Slack_i(X'), 0)} \quad (18)$$

where $n_{Ca} = |C_a|$ is the number of unserved demand nodes, C_a is the set of unserved nodes, Ω is the rescheduling Pareto set, and $R_{slack}(X) \in [0, 1]$.

The three metrics are combined using analytic hierarchy process (AHP) weights calibrated for emergency scenarios, where $\lambda_1 = 0.4$ for load redundancy, $\lambda_2 = 0.3$ for range redundancy, and $\lambda_3 = 0.3$ for slack redundancy ($\lambda_1 + \lambda_2 + \lambda_3 = 1$). The comprehensive robustness score is computed as follows.

$$R_{total}(X) = \lambda_1 \cdot \frac{1}{|V_a|} \sum_{v \in V_a} R_{load}(v, X) + \lambda_2 \cdot \frac{1}{|V_a|} \sum_{v \in V_a} R_{range}(v, X) + \lambda_3 \cdot R_{slack}(X) \quad (19)$$

where the two summation terms are the fleet-average load and range redundancies, respectively, and $R_{total}(X) \in [0, 1]$, where higher values indicate a more resilient plan.

When a breakdown is detected, the robustness-embedded rescheduling process activates in three steps. First, the system state is frozen at the moment vehicle v_b fails, extracting key parameters including the set of operational vehicles V_a , the set of unfinished tasks $C_b^{v_b}$ of the broken-down vehicle, its remaining cargo u_{v_b, t_b} (where t_b is the breakdown time), and real-time status of all healthy vehicles (position, remaining load, distance traveled). A rescheduling problem for the outstanding deliveries is then reconstructed with the same bi-objective goals. Second, FR-AMOHAs evolutionary and local search framework solves this reduced problem to produce a Pareto set Ω of candidate plans. Third, $R_{total}(X)$ is computed for every plan in Ω , and the top 30% most robust plans are retained as the final candidate set, preserving optimization quality while hardening the schedule against further disruptions. The 30% threshold is selected based on preliminary sensitivity analysis that tested retention ratios ranging from 10% to 50%. Lower ratios (10–20%) excessively narrow the candidate pool, limiting the diversity of available solutions for decision-makers and occasionally discarding plans with marginally lower robustness but substantially better time or cost performance. Higher ratios (40–50%) include too many fragile plans that lack sufficient redundancy to withstand subsequent disturbances. The 30% level strikes an effective balance. It preserves solution diversity for flexible decision-making while ensuring that all retained plans meet a meaningful robustness standard, and this setting showed the most stable performance across all test instances in our pilot experiments.

3.4. ALM-PDR

The standard HGS local search cycles through a fixed sequence of neighborhood operators and accepts moves based on a single-objective criterion, a design ill-suited to bi-objective optimization, where different operators excel at improving different objectives. A rigid sequence wastes search effort and traps the algorithm on a local Pareto front. ALM-PDR addresses this by evaluating each operator's historical contribution to both objectives and dynamically adjusting selection probabilities, combined with a Pareto-dominance acceptance rule that preserves solution diversity.

Four complementary neighborhood operators form the search toolkit (Figure 4). The 2-opt operator reverses a sub-path within a single route to eliminate crossovers and shorten travel distance, primarily benefiting the time objective. The swap operator exchanges one demand node between two routes to rebalance load and duration, also favoring time reduction. The insert operator relocates a demand node from one route into the best position of another, improving load utilization and primarily serving the cost objective. The cross-exchange operator swaps two consecutive sub-paths between routes, providing a balanced effect on both objectives.

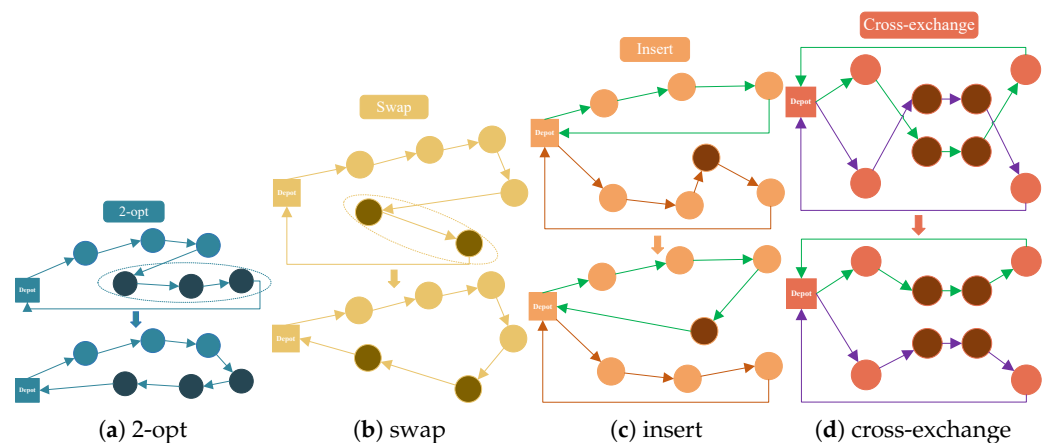


Figure 4. Four types of multi-neighborhood operators.

Each operator's historical track record is captured by the neighborhood objective contribution $\Delta_m(\mathcal{N}_t)$, the average improvement operator $\mathcal{N}_t$ has delivered on objective m , where $m = 1$ denotes time and $m = 2$ denotes cost, defined as follows:

$$\Delta_m(\mathcal{N}_t) = \frac{1}{H_t} \sum_{h=1}^{H_t} (F_m(X_h) - F_m(X'_h)) \quad (20)$$

where H_t is the cumulative call count of $\mathcal{N}_t$, X_h and X'_h are the solutions before and after the h -th application, and $F_m(X)$ is the m -th objective value. A call contributes to Δ_m only when X'_h Pareto-dominates X_h . Otherwise, the contribution is recorded as zero. Larger $\Delta_m(\mathcal{N}_t)$ means operator $\mathcal{N}_t$ has historically been more effective at improving objective m .

Selection probabilities are computed adaptively so that the operator best at improving the current solution's weaker objective is favored. Two steps are involved. First, the current solution's objective values are normalized as follows.

$$\tilde{F}_m(X) = \frac{F_m(X) - \min_{X' \in \Omega} F_m(X')}{\max_{X' \in \Omega} F_m(X') - \min_{X' \in \Omega} F_m(X')}, \quad m = 1, 2 \quad (21)$$

where Ω is the current Pareto set and $\tilde{F}_m(X) \in [0, 1]$. The objective deviation degree then measures how far the current solution lags behind the set average, as shown below.

$$D_m(X) = \tilde{F}_m(X) - \frac{1}{|\Omega|} \sum_{X' \in \Omega} \tilde{F}_m(X') \quad (22)$$

A larger $D_m(X)$ signals that objective m is the current bottleneck. Second, the softmax function converts deviation-weighted contributions into selection probabilities, as defined below.

$$p(\mathcal{N}_t) = \frac{\exp\left(\beta \cdot \sum_{m=1}^2 D_m(X) \cdot \Delta_m(\mathcal{N}_t)\right)}{\sum_{t'=1}^4 \exp\left(\beta \cdot \sum_{m=1}^2 D_m(X) \cdot \Delta_m(\mathcal{N}_{t'})\right)} \quad (23)$$

where β (default 2) controls the exploration and exploitation trade-off. Larger β concentrates selection on historically effective operators (exploitation), while smaller β encourages random exploration. This parameter can be adjusted dynamically during the run.

New solutions generated by the neighborhood operators are accepted or rejected via a Pareto-dominance rule. If X' strongly dominates X (i.e., $F_1(X') \leq F_1(X)$ and $F_2(X') \leq F_2(X)$ with at least one strict inequality), it is accepted outright. If X' and X are mutually non-dominated, acceptance follows a simulated annealing probability.

$$p_{\text{accept}} = \exp\left(-\frac{\tilde{F}_1(X') + \tilde{F}_2(X') - (\tilde{F}_1(X) + \tilde{F}_2(X))}{\mathcal{T}}\right) \quad (24)$$

where $\mathcal{T}$ is the annealing temperature, initialized at $\mathcal{T}_0 = 1$ and decayed by $\mathcal{T} = \mathcal{T} \times \gamma$ per iteration. This rule prevents premature convergence to a local Pareto front while preserving solution diversity. If X strongly dominates X' , the new solution is rejected. After each acceptance decision, the corresponding $\Delta_m(\mathcal{N}_t)$ is updated, completing one local search iteration.

3.5. Time Complexity Analysis of FR-AMOH

We now analyze the computational cost of FR-AMOH module by module. Let P be the population size, G the maximum iterations, n_C the number of demand nodes, and B the maximum number of breakdown events.

Initial population generation. Entropy-weight standardization and fuzzy evaluation each cost $O(n_K \cdot n_C)$, while greedy pre-matching for P individuals costs $O(P \cdot n_K \cdot n_C)$. The dominant term is $O(P \cdot n_K \cdot n_C)$.

Single-iteration evolutionary optimization. Genetic operations (selection, crossover, mutation) cost $O(P \cdot n_C)$. Adaptive local search applies one neighborhood operator per individual at $O(n_C^2)$ per operation, giving $O(P \cdot n_C^2)$ for the full population. Non-dominated sorting and crowding distance computation cost $O(P^2 \cdot n_C)$, which grows more slowly than $O(P \cdot n_C^2)$. The per-iteration cost is therefore $O(P \cdot n_C^2)$.

Dynamic breakdown rescheduling. Each breakdown triggers a reduced-scale re-optimization over $n'_C \leq n_C$ nodes and $n'_V \leq n_V$ vehicles at cost $O(P \cdot n_C^2)$. Over B breakdowns in the worst case, the total rescheduling cost is $O(B \cdot P \cdot n_C^2)$.

Based on the above analysis, the overall worst-case time complexity of the FR-AMOH algorithm is given by

$$O(G \cdot P \cdot n_C^2 + B \cdot P \cdot n_C^2) = O((G + B) \cdot P \cdot n_C^2) \quad (25)$$

This complexity is polynomial in n_C (quadratic in n_C), which is favorable for large-scale applications. Population size P and iteration count G are user-controlled parameters that remain roughly constant across problem scales in practice, typically set to 100 and 5000 respectively regardless of instance size. The number of breakdown events B is bounded by n_V , which grows much more slowly than n_C in realistic emergency scenarios. For com-

parison, exact mixed-integer linear programming (MILP) solvers such as Gurobi show exponential worst-case complexity for NP-hard routing problems, which explains their inability to scale beyond small instances as observed in our experiments. The polynomial-time complexity of FR-AMOHHA enables it to handle instances with up to 1000 demand nodes within practical computation times, making it suitable for real-world emergency logistics deployment.

In terms of practical running time, FR-AMOHHA completes instances with 1000 nodes in roughly 250 s on standard hardware, which is comparable to or faster than many state-of-the-art metaheuristics reported in the recent VRP literature. For instance, TAVNS requires 230 s on the same large instances, while NSGA-II and NSGA-III are faster (around 70 s) but produce clearly lower solution quality. This running time is well within the decision-making window available to emergency logistics coordinators, who typically have 30 min to a few hours to develop initial response plans after a disaster event.

4. Experimental Comparisons and Analysis

4.1. Experimental Setup and Evaluation Metrics

All algorithms are implemented in Python 3.13 and run on a machine with an Intel Core i7-12700 processor and 16 GB RAM under Windows 11. Every experiment uses the same hardware and software environment to ensure fair comparison.

The test suite comprises 40 real-world post-disaster delivery instances with 50 to 1000 demand nodes, covering small, medium, and large scales. The first 30 instances are built from field data of emergency logistics operations in Nanchang. The remaining 10 come from JD Logistics distribution records, adding diversity in network structure and demand patterns. During simulation, each vehicle may suffer a sudden breakdown with fixed probability and immediately ceases service, triggering dynamic rescheduling of the surviving fleet to complete outstanding deliveries. Cargo transshipment at the breakdown site is also supported when operationally necessary.

Two widely used and complementary metrics, inverted generational distance (IGD) and hypervolume (HV), evaluate the quality of the Pareto sets produced by each algorithm. All metric calculations use instance-specific normalization to ensure consistent evaluation. Each algorithm is run independently 20 times per instance, and reported values represent means across runs. Statistical significance is assessed using the Wilcoxon rank-sum test at the 0.05 significance level. IGD measures both convergence and spread of the approximate front relative to a reference.

$$\text{IGD}(\mathcal{R}, A) = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \min_{a \in A} \|r - a\|_2 \quad (26)$$

where $\mathcal{R}$ is a pseudo-reference Pareto front adaptively constructed from the best non-dominated solutions across all runs, and A is the solution set under evaluation. Lower IGD indicates better convergence and coverage. HV captures the dominated volume in objective space, jointly reflecting convergence and diversity.

$$\text{HV}(A, \mathbf{r}^{\text{ref}}) = \lambda \left(\bigcup_{p \in A} [F_1(p), r_1^{\text{ref}}] \times [F_2(p), r_2^{\text{ref}}] \right) \quad (27)$$

where $\lambda(\cdot)$ is the Lebesgue measure, $\mathbf{r}^{\text{ref}} = (r_1^{\text{ref}}, r_2^{\text{ref}})$ is an instance-specific reference point, and $F_1(p), F_2(p)$ are the two objective values of solution p . Higher HV indicates better overall performance. For each instance, the best non-dominated subset across all algorithms constructs the reference front.

4.2. Parameter Settings and Comparative Algorithms

All algorithm parameters are set based on preliminary sensitivity analysis, and their values are summarized in Table 2 for reproducibility.

Table 2. Parameter settings for FR-AMOH.

Parameter	Symbol	Value	Description
Population size	P	100	Number of individuals in the population
Maximum iterations	G	100	Termination criterion
Initial temperature	T_0	1.0	Starting temperature for simulated annealing
Cooling rate	γ	0.99	Temperature decay factor per iteration
Selection pressure	β	2.0	Exploration and exploitation balance in operator selection
Breakdown inspection interval	I_b	10	Iterations between breakdown checks
Robust retention ratio	ρ	0.3	Fraction of robust solutions retained
Load redundancy weight	λ_1	0.4	Weight for load redundancy in robustness
Range redundancy weight	λ_2	0.3	Weight for range redundancy in robustness
Slack redundancy weight	λ_3	0.3	Weight for slack redundancy in robustness
Perturbation rate	r_p	0.15	Random perturbation rate in initialization

Seven representative algorithms/solvers are selected for comparison.

- NSGA-II [14]. An elitist multi-objective genetic algorithm with fast non-dominated sorting and crowding distance for diverse Pareto approximations.
- MOEA/D [15]. Decomposes a multi-objective problem into scalar subproblems optimized together.
- NSGA-III [26]. The Reference-point-based non-dominated sorting genetic algorithm III extends NSGA-II with reference-point adaptation for many-objective problems.
- ALNS [10]. Adaptive large neighborhood search well-suited to combinatorial VRP structures.
- TAVNS [27]. The two-stage adaptive variable neighborhood search with dedicated pre-optimization and rescheduling modules.
- Gurobi [41]. A leading commercial exact solver with powerful constraint handling. We include Gurobi as a baseline despite its status as an exact solver because it provides a well-established optimization benchmark that represents the current state of the art in exact MILP solving. Comparing against Gurobi on small instances where it can terminate within time limits allows us to quantify how far the heuristic solutions are from proven optima, and showing that FR-AMOH matches or exceeds Gurobi's solution quality on those instances, while scaling to much larger instances where Gurobi cannot run within practical time limits, validates the effectiveness of our approach. We acknowledge that heuristic approaches are expected to outperform exact solvers on large NP-hard problems, and our inclusion of Gurobi serves as a sanity check rather than a primary claim of superiority.
- PyVRP [42]. An open-source HGS-based VRP solver with a mixed Python/C++ architecture.

4.3. Performance Comparison with Seven Involved Algorithms

Before presenting the results, we note the following conventions used consistently across all result tables. Instances prefixed "A" are Nanchang emergency logistics cases, while those prefixed "B" come from JD Logistics. Each algorithm runs independently 20 times per instance, and reported values represent means over these runs. The best value per instance is **bolded**. "N/A" indicates that the corresponding algorithm (Gurobi) cannot produce a solution of comparative value within acceptable computation time on

large instances due to its exponential-time nature. Statistical significance is assessed via the Wilcoxon rank-sum test at the 0.05 significance level, where “+” indicates that the competitor significantly outperforms FR-AMOH, “=” indicates no statistically significant difference, and “−” indicates that FR-AMOH significantly outperforms the competitor. Since the true Pareto front is unknown, the best observed front across all algorithms serves as the reference, yielding an optimal IGD of 0.000.

Table 3. IGD values of the eight compared algorithms over 40 real-world instances.

Instance	NSGA-II	MOEA/D	NSGA-III	ALNS	TAVNS	Gurobi	PyVRP	FR-AMOH
A_01_n50	0.220−	0.328−	0.273−	0.509−	0.300−	1.190−	0.157−	0.000
A_02_n83	0.072=	0.090=	0.239−	0.522−	0.000+	1.414−	0.711−	0.074
A_03_n116	0.061−	0.165−	0.006=	0.677−	0.111−	1.414−	0.921−	0.000
A_04_n148	0.081−	0.275−	0.153−	0.927−	0.216−	1.414−	1.284−	0.000
A_05_n181	0.000=	0.233−	0.122−	0.831−	0.181−	1.414−	1.080−	0.007
A_06_n214	0.183=	0.229=	0.000+	0.838−	0.249=	1.414−	1.136−	0.213
A_07_n247	0.011+	0.123=	0.000+	0.633−	0.217−	1.414−	0.901−	0.115
A_08_n279	0.076=	0.252−	0.082=	0.841−	0.000+	1.414−	1.200−	0.079
A_09_n312	0.026+	0.265=	0.000+	0.475−	0.238=	1.414−	1.025−	0.241
A_10_n345	0.000+	0.248−	0.054+	0.455−	0.106=	1.414−	1.076−	0.120
A_11_n378	0.040+	0.263+	0.071+	0.546−	0.000+	1.414−	1.104−	0.332
A_12_n410	0.000+	0.234=	0.176+	0.663−	0.179+	2.358−	1.414−	0.230
A_13_n443	0.011=	0.183−	0.163−	0.609−	0.022=	N/A−	1.414−	0.000
A_14_n476	0.214−	0.546−	0.000+	0.515−	0.075=	N/A−	1.414−	0.087
A_15_n509	0.035=	0.176−	0.000=	0.650−	0.018=	N/A−	1.414−	0.003
A_16_n541	0.125−	0.458−	0.196−	0.641−	0.000=	N/A−	1.414−	0.045
A_17_n574	0.179−	0.472−	0.254−	0.523−	0.093−	N/A−	1.414−	0.000
A_18_n607	0.145+	0.303−	0.155+	0.724−	0.000+	N/A−	1.414−	0.233
A_19_n640	0.012=	0.274−	0.011=	0.378−	0.000=	N/A−	1.414−	0.004
A_20_n672	0.446−	0.496−	0.326−	0.541−	0.342−	N/A−	1.414−	0.000
A_21_n705	0.134−	0.431−	0.364−	0.595−	0.118−	N/A−	1.414−	0.000
A_22_n738	0.072−	0.494−	0.142−	0.489−	0.005=	N/A−	1.414−	0.000
A_23_n771	0.161−	0.416−	0.125−	0.345−	0.083−	N/A−	1.414−	0.000
A_24_n803	0.205−	0.316−	0.498−	0.300−	0.033=	N/A−	1.414−	0.000
A_25_n836	0.189−	0.286−	0.134−	0.387−	0.120−	N/A−	1.414−	0.000
A_26_n869	0.188−	0.416−	0.521−	0.485−	0.257−	N/A−	1.414−	0.000
A_27_n902	0.380−	0.483−	0.570−	0.682−	0.323−	N/A−	1.414−	0.000
A_28_n934	0.131−	0.565−	0.333−	0.294−	0.000=	N/A−	1.414−	0.046
A_29_n967	0.130−	0.373−	0.108=	0.511−	0.000+	N/A−	1.414−	0.064
A_30_n1000	0.260−	0.457−	0.305−	0.354−	0.139−	N/A−	1.414−	0.000
B_01_n138	0.000+	0.221+	0.134+	0.935−	0.105+	N/A−	1.414−	0.362
B_02_n207	0.097+	0.473−	0.000+	0.941−	0.044+	N/A−	1.414−	0.260
B_03_n277	0.000+	0.201=	0.200=	0.725−	0.031+	N/A−	1.414−	0.187
B_04_n346	0.093+	0.237=	0.242=	0.928−	0.000+	N/A−	1.414−	0.242
B_05_n416	0.000=	0.272−	0.140−	0.508−	0.134−	N/A−	1.414−	0.043
B_06_n485	0.077−	0.299−	0.172−	0.538−	0.176−	N/A−	1.414−	0.000
B_07_n555	0.204+	0.419−	0.203+	0.688−	0.000+	N/A−	1.414−	0.258
B_08_n624	0.079−	0.385−	0.176−	0.626−	0.074−	N/A−	1.414−	0.000
B_09_n694	0.000+	0.408−	0.208−	0.606−	0.043=	N/A−	1.414−	0.076
B_10_n763	0.056−	0.484−	0.350−	0.742−	0.297−	N/A−	1.414−	0.000
+ / = / −	12/8/20	2/7/31	11/7/22	0/0/40	11/12/17	0/0/40	0/0/40	−

Table 3 reports IGD values for all eight algorithms. Gurobi and PyVRP perform the worst, with IGD reaching 1.414 on most instances. This pattern reflects persistent entrapment in local optima. ALNS fares somewhat better (all IGD < 1) but remains worse than FR-AMOH on every instance. Among the metaheuristics, NSGA-II is the strongest competitor, beating FR-AMOH on 12 small and medium instances yet losing on 20, failing to scale to the A_13_n443–A_30_n1000 range. NSGA-III mirrors this pattern with slightly weaker overall results. TAVNS ranks second only to FR-AMOH, losing on 17 instances

but winning on 11; however, FR-AMOHAs holds the edge across all scale ranges and claims optimal solutions on 17 instances versus TAVNS's 10. These results confirm FR-AMOHAs clear advantage in IGD performance.

HV results (Table 4) broadly mirror the IGD findings. TAVNS beats FR-AMOHAs on 16 instances but loses on 20, with a wider performance gap than on IGD. The remaining competitors show even larger differences from FR-AMOHAs on HV, with fewer ties and more decisive wins or losses.

Table 4. HV values of the eight compared algorithms over 40 real-world instances.

Instance	NSGA-II	MOEA/D	NSGA-III	ALNS	TAVNS	Gurobi	PyVRP	FR-AMOHAs
A_01_n50	1.010–	0.839–	1.318+	0.584–	0.978–	0.036–	1.077–	1.191
A_02_n83	1.389–	1.360–	1.124–	0.740–	1.386–	0.053–	0.528–	1.511
A_03_n116	1.564+	1.385–	1.663+	0.664–	1.674+	0.086–	0.413–	1.477
A_04_n148	1.550–	1.448–	1.662–	0.551–	1.952+	0.158–	0.240–	1.796
A_05_n181	1.908=	1.492–	1.689–	0.638–	1.583–	0.149–	0.387–	1.921
A_06_n214	1.576–	1.497–	1.919+	0.628–	1.524–	0.148–	0.339–	1.823
A_07_n247	1.930+	1.716=	1.952+	0.901–	1.919+	0.158–	0.578–	1.732
A_08_n279	2.159=	1.814–	2.152=	0.866–	2.326+	0.276–	0.457–	2.164
A_09_n312	1.968=	1.522–	2.020+	1.178–	1.565–	0.178–	0.485–	1.945
A_10_n345	2.299+	1.798–	2.185+	1.426–	2.286+	0.267–	0.571–	2.050
A_11_n378	2.102+	1.670+	2.039+	1.192–	2.185+	0.229–	0.486–	1.546
A_12_n410	2.779+	2.254=	2.379+	1.436–	2.374+	N/A–	0.445–	2.263
A_13_n443	2.768=	2.376–	2.422–	1.538–	2.742–	N/A–	0.450–	2.793
A_14_n476	1.928–	1.330–	2.370+	1.382–	2.210=	N/A–	0.291–	2.184
A_15_n509	2.511–	2.204–	2.589=	1.322–	2.548=	N/A–	0.371–	2.581
A_16_n541	2.093–	1.466–	1.950–	1.169–	2.356+	N/A–	0.286–	2.260
A_17_n574	2.276–	1.693–	2.120–	1.601–	2.464–	N/A–	0.403–	2.674
A_18_n607	2.397+	2.064–	2.375+	1.297–	2.726+	N/A–	0.424–	2.209
A_19_n640	3.019=	2.408–	3.019=	2.186–	3.038=	N/A–	0.556–	3.047
A_20_n672	1.480–	1.396–	1.693–	1.322–	1.665–	N/A–	0.283–	2.347
A_21_n705	2.306–	1.713–	1.839–	1.423–	2.341–	N/A–	0.376–	2.603
A_22_n738	2.546–	1.683–	2.391–	1.693–	2.700=	N/A–	0.418–	2.711
A_23_n771	2.570–	2.024–	2.653–	2.170–	2.749–	N/A–	0.514–	2.947
A_24_n803	2.347–	2.111–	1.754–	2.144–	2.734–	N/A–	0.458–	2.811
A_25_n836	2.363–	2.155–	2.482–	1.951–	2.514–	N/A–	0.450–	2.791
A_26_n869	2.279–	1.817–	1.622–	1.688–	2.133–	N/A–	0.412–	2.697
A_27_n902	1.929–	1.731–	1.573–	1.381–	2.043–	N/A–	0.432–	2.747
A_28_n934	2.651–	1.746–	2.207–	2.290–	2.961+	N/A–	0.520–	2.851
A_29_n967	3.204–	2.619–	3.260–	2.312–	3.541+	N/A–	0.778–	3.372
A_30_n1000	2.309–	1.905–	2.214–	2.112–	2.576–	N/A–	0.495–	2.902
B_01_n138	1.461–	1.711–	1.877–	0.646–	1.934–	N/A–	0.216–	2.145
B_02_n207	1.954+	1.281–	2.150+	0.642–	2.061+	N/A–	0.218–	1.646
B_03_n277	1.900–	1.875–	1.875–	0.997–	2.216–	N/A–	0.261–	2.283
B_04_n346	2.542+	2.226=	2.217=	1.007–	2.755+	N/A–	0.435–	2.215
B_05_n416	2.447–	1.965–	2.237–	1.526–	2.250–	N/A–	0.353–	2.543
B_06_n485	2.459+	1.990–	2.251=	1.542–	2.632+	N/A–	0.387–	2.245
B_07_n555	2.101+	1.684–	2.104+	1.226–	2.541+	N/A–	0.353–	1.993
B_08_n624	2.592–	1.943–	2.375–	1.496–	2.604–	N/A–	0.443–	2.775
B_09_n694	2.720+	1.852–	2.257–	1.490–	2.620+	N/A–	0.422–	2.546
B_10_n763	2.775–	1.859–	2.125–	1.395–	2.236–	N/A–	0.498–	2.909
+ / = / –	11 / 5 / 24	1 / 3 / 36	12 / 5 / 23	0 / 0 / 40	16 / 4 / 20	0 / 0 / 40	0 / 0 / 40	–

4.4. Convergence Curves Comparison with Six Metaheuristic Algorithms

Figure 5 plots convergence curves for six metaheuristic algorithms (Gurobi is not included because as an exact solver it does not have an iterative convergence curve in the heuristic sense; it either finds the optimum or times out, and on larger instances it exceeds the time limit before producing meaningful intermediate results). PyVRP converges the slowest across all four instances. Its curve nearly plateaus around generation 30 on

A_03_n116 and A_17_n574. FR-AMOH, by contrast, leads from generation 20 onward on three instances and pulls ahead clearly by generation 40 on A_17_n574. NSGA-II also shows fast convergence on three of the four instances. TAVNS's advantage is less visible here because the figure tracks a single objective; it may perform better on the other. Overall, FR-AMOH exhibits the fastest convergence.

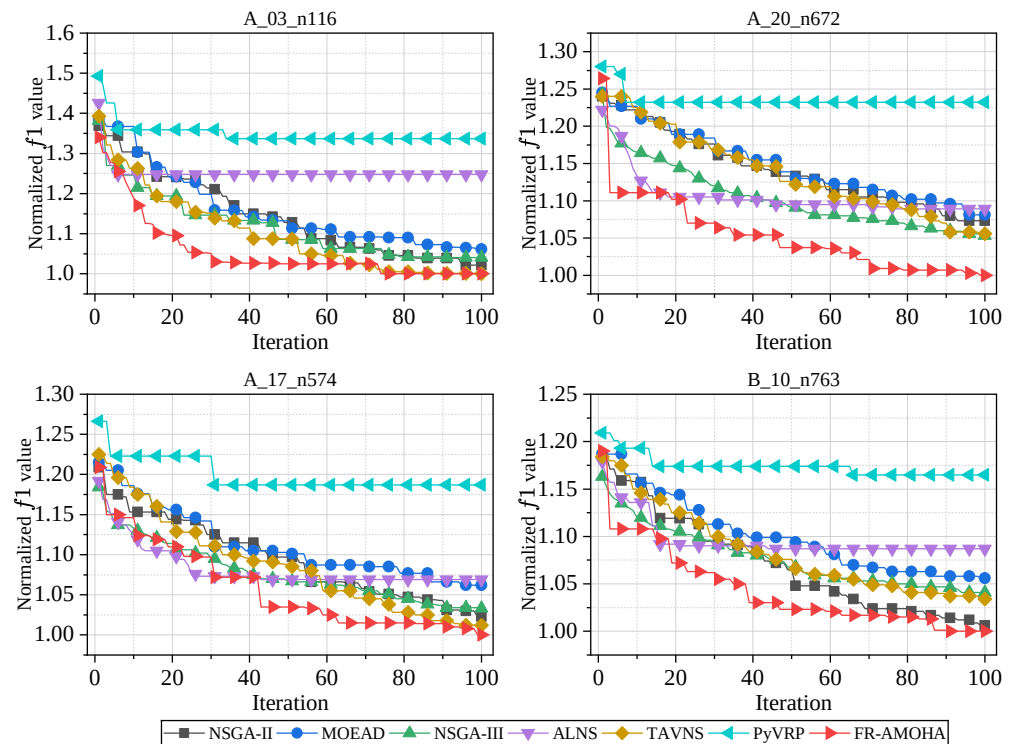


Figure 5. Convergence curves of the six compared metaheuristic algorithms.

4.5. Running Time Comparison with Seven Involved Algorithms

Table 5 summarizes the wall-clock times. Gurobi is by far the slowest, often exceeding thousands of seconds on large instances. This behavior stems from its exact algorithm approach. PyVRP stays in a narrow 400–700 s band because its local search budget is capped by function evaluations. NSGA-II, MOEA/D, and NSGA-III are the fastest (<100 s per instance) but produce low-quality Pareto sets. FR-AMOH requires 40–250 s, well within a practical range, while delivering the best solution quality. The algorithm strikes a favorable balance between optimization performance and computational cost.

Table 5. Running time (s) of the eight compared algorithms over 40 real-world instances.

Instance	NSGA-II	MOEA/D	NSGA-III	ALNS	TAVNS	Gurobi	PyVRP	FR-AMOH
A_01_n50	3	2	2	5	37	3119	454	42
A_02_n83	4	3	2	19	56	3326	503	62
A_03_n116	4	3	3	46	53	3585	505	64
A_04_n148	5	3	4	92	54	3945	508	62
A_05_n181	4	4	4	160	57	4700	512	64
A_06_n214	5	4	5	240	69	5816	515	67
A_07_n247	6	6	7	294	65	7853	519	72

Table 5. Cont.

Instance	NSGA-II	MOEA/D	NSGA-III	ALNS	TAVNS	Gurobi	PyVRP	FR-AMOHA
A_08_n279	7	7	9	416	65	9804	524	76
A_09_n312	9	8	9	480	73	13,228	530	79
A_10_n345	11	10	11	422	77	17,020	535	84
A_11_n378	14	11	13	521	80	21,497	542	88
A_12_n410	15	13	16	651	84	N/A	547	93
A_13_n443	17	15	17	646	91	N/A	554	97
A_14_n476	20	17	20	774	97	N/A	563	102
A_15_n509	21	19	22	849	111	N/A	568	108
A_16_n541	23	21	24	801	117	N/A	580	114
A_17_n574	25	24	26	1064	113	N/A	609	119
A_18_n607	28	27	30	929	121	N/A	626	128
A_19_n640	30	28	31	1236	125	N/A	640	133
A_20_n672	33	31	34	1294	143	N/A	656	142
A_21_n705	36	35	38	1347	143	N/A	637	150
A_22_n738	39	37	41	1459	152	N/A	650	159
A_23_n771	44	43	46	1437	161	N/A	657	171
A_24_n803	46	44	48	1396	180	N/A	677	179
A_25_n836	50	48	53	1437	182	N/A	706	189
A_26_n869	53	53	56	1764	191	N/A	727	199
A_27_n902	57	59	61	1643	212	N/A	729	211
A_28_n934	59	61	63	1941	209	N/A	732	218
A_29_n967	64	65	67	1833	228	N/A	750	233
A_30_n1000	69	70	73	1932	230	N/A	771	247
B_01_n138	4	3	3	81	59	N/A	508	58
B_02_n207	6	5	5	240	60	N/A	515	68
B_03_n277	8	7	8	343	78	N/A	525	75
B_04_n346	11	11	11	432	85	N/A	538	84
B_05_n416	15	15	15	706	89	N/A	552	94
B_06_n485	19	19	20	871	108	N/A	569	106
B_07_n555	23	24	25	846	118	N/A	587	117
B_08_n624	28	29	30	971	134	N/A	608	131
B_09_n694	34	37	37	1381	152	N/A	634	150
B_10_n763	40	43	44	1309	171	N/A	660	169
Sum	989	964	1033	34,308	4630	NA	23,722	4804

4.6. Ablation Study

To isolate the contribution of each mechanism, three ablated variants are tested.

- FR-AMOHA-N1. HVDNPM removed (random initialization instead).
- FR-AMOHA-N2. ROM-MDR removed (no robustness-guided rescheduling).
- FR-AMOHA-N3. ALM-PDR removed (fixed neighborhood sequence).

Table 6 compares the three variants against the full FR-AMOHA. All three suffer performance degradation to varying degrees. N1 is the weakest, outperforming FR-AMOHA on only four instances and losing on thirty-four, confirming that the HVDNPM's high-quality initialization is the single most impactful component. N2 shows no statistically significant difference from FR-AMOHA on 31 instances, yet its practical importance should not be dismissed. When breakdowns actually occur, robustness-guided rescheduling is what prevents cascading service failures. N3 loses on 24 instances despite winning on 13, indicating that the adaptive neighborhood selection of ALM-PDR plays a substantial role in overall optimization quality. Together, these results validate that all three mechanisms contribute meaningfully to FR-AMOHA's performance.

Table 6. IGD values of the three modified algorithms over 40 real-world instances.

Instance	FR-AMOHA-N1	FR-AMOHA-N2	FR-AMOHA-N3	FR-AMOHA
A_01_n50	0.000=	0.058=	0.062=	0.046
A_02_n83	0.000=	0.021=	1.414–	0.021
A_03_n116	1.414–	0.000=	0.851–	0.000
A_04_n148	1.291+	1.414=	0.000+	1.414
A_05_n181	1.414–	0.001=	0.238–	0.000
A_06_n214	1.414–	0.000=	1.105–	0.000
A_07_n247	1.414–	0.001=	1.281–	0.000
A_08_n279	1.414–	0.000=	0.240–	0.000
A_09_n312	1.144+	1.414=	0.000+	1.414
A_10_n345	1.414–	0.000=	0.799–	0.000
A_11_n378	0.000+	1.414=	0.835+	1.414
A_12_n410	0.000+	1.414=	0.595+	1.414
A_13_n443	1.414–	0.105–	0.887–	0.000
A_14_n476	1.414–	0.241–	1.262–	0.000
A_15_n509	1.414–	0.000=	0.249–	0.000
A_16_n541	1.414–	0.236–	0.818–	0.000
A_17_n574	1.414–	0.000=	0.237–	0.000
A_18_n607	1.414–	0.679=	0.000+	0.679
A_19_n640	1.017–	0.000=	1.414–	0.000
A_20_n672	1.414–	0.424–	0.250–	0.000
A_21_n705	1.414–	0.436=	0.000+	0.436
A_22_n738	1.414–	0.000=	0.242–	0.000
A_23_n771	1.414–	0.311–	0.485–	0.000
A_24_n803	1.414–	0.176–	0.226–	0.000
A_25_n836	1.414–	0.022=	0.000=	0.022
A_26_n869	1.414–	0.000=	0.417–	0.000
A_27_n902	1.414–	0.064–	0.752–	0.000
A_28_n934	1.414–	0.017=	1.158–	0.000
A_29_n967	1.414–	0.943=	0.000+	0.943
A_30_n1000	1.414–	0.500=	0.000+	0.500
B_01_n138	1.414–	0.000=	0.135–	0.000
B_02_n207	1.414–	0.522=	0.000+	0.522
B_03_n277	1.414–	0.044=	0.000=	0.044
B_04_n346	1.414–	0.574=	0.000+	0.574
B_05_n416	1.414–	0.123–	0.738–	0.000
B_06_n485	1.414–	0.269=	0.000+	0.269
B_07_n555	1.414–	0.554=	0.000+	0.554
B_08_n624	1.414–	0.339–	0.708–	0.000
B_09_n694	1.414–	0.361=	0.000+	0.361
B_10_n763	1.414–	0.000=	0.301–	0.000
+ / = / –	4 / 2 / 34	0 / 31 / 9	13 / 3 / 24	–

5. Conclusions and Future Work

Efficient fleet scheduling is the foundation of post-disaster relief operations. This paper has tackled the dynamic scheduling of heterogeneous emergency vehicles subject to real-time breakdowns (DHEVRP-VB), formulated as a two-stage MIP model that jointly minimizes transportation time and cost. To solve this NP-hard problem, we developed FR-AMOHA, which integrates three complementary mechanisms. HVDNPM uses entropy-weighted fuzzy evaluation to seed the population with task-adaptive, high-quality initial solutions. ROM-MDR activates upon breakdown detection to reconstruct and robustify the schedule using multi-dimensional resilience metrics. ALM-PDR dynamically selects among four neighborhood operators based on their historical contribution to each objective, balancing exploration and exploitation under Pareto dominance. Experiments on 40 real-world instances (50–1000 demand nodes) demonstrate that FR-AMOHA consistently

outperforms Gurobi, NSGA-II, and five other leading algorithms in both IGD and HV metrics, while keeping computation time within a practical range.

From a practical standpoint, the proposed framework offers value for emergency logistics operations. The bi-objective formulation provides decision-makers with a Pareto front of solutions rather than a single prescribed plan, enabling them to balance response speed against operational cost based on disaster phase and resource availability. The robustness-aware rescheduling mechanism helps operators move beyond purely short-sighted optimization by explicitly accounting for subsequent failure risks, reducing the likelihood of cascading failures. The polynomial-time complexity of the algorithm means these decisions can be computed within minutes even for large-scale scenarios, supporting real-time decision-making in the field. Several limitations of the present study should be acknowledged. The model does not yet incorporate time-dependent travel speeds reflecting road recovery dynamics, multiple supply categories are handled implicitly rather than explicitly, robustness metrics do not incorporate probabilistic breakdown prediction, and algorithm parameters are set through pilot experiments rather than learned automatically from data.

These limitations point to several promising directions for future research. The model could be extended to explicitly capture multiple supply categories with vehicle compatibility constraints, time-dependent travel times reflecting road recovery dynamics, and coordinated multi-depot operations, thereby broadening its operational relevance. On the algorithm side, reinforcement learning or other data-driven methods could replace manual parameter tuning, while integrating machine learning models to predict breakdown probabilities would allow the robustness module to move toward proactive risk-aware planning. Field deployment and validation with actual emergency management agencies would also provide valuable feedback to refine the model assumptions and algorithm behavior for real-world operational use.

Author Contributions: Conceptualization, Z.C.; methodology, Z.C. and C.X.; software, Z.C. and C.X.; data curation, Z.C.; writing—original draft preparation, Z.C.; writing—review and editing, Z.C. and C.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Project of Education Department of Jiangxi Province (No. GJJ2200381), Jiangxi Province Technological Innovation Base Program (No. 20242BCC32021).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in the experiment is available online: <https://github.com/zhuangcai97/FR-AMOH> (accessed on 25 June 2026).

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

References

1. Lu, Y.; Yang, J.; Yang, C. A humanitarian vehicle routing problem synchronized with drones in time-varying weather conditions. *Comput. Ind. Eng.* **2023**, *184*, 109563.
2. Wang, W.; Yang, K.; Yang, L.; Gao, Z. Distributionally robust chance-constrained programming for multi-period emergency resource allocation and vehicle routing in disaster response operations. *Omega* **2023**, *120*, 102915.
3. Liu, H.; Sun, Y.; Pan, N.; Li, Y.; An, Y.; Pan, D. Study on the optimization of urban emergency supplies distribution paths for epidemic outbreaks. *Comput. Oper. Res.* **2022**, *146*, 105912.
4. Zhao, X.; Bai, Q.; Wei, J.; Yang, B. A routing problem with dynamic damage and multi-road selection in post-earthquake humanitarian logistics. *Expert Syst. Appl.* **2026**, *314*, 131644.
5. Hu, Y.; Zhang, P.; Zhao, K.; Zhang, S.; Fan, B. Disruption recovery for the pickup and delivery problem with time windows: A scenario-based approach for online food delivery. *Comput. Oper. Res.* **2023**, *159*, 106337.

6. Ahmed, F.; Huynh, N.; Ferrell, W.; Badyal, V.; Padmanabhan, B. Vehicle re-routing under disruption in cross-dock network with time constraints. *Expert Syst. Appl.* **2024**, *237*, 121517.
7. Doostmohammadian, M.; Pequito, S. Distributed allocation and resource scheduling algorithms resilient to link failure. *Eur. J. Control* **2025**, *86*, 101405.
8. Pasha, J.; Nwodu, A.L.; Fathollahi-Fard, A.M.; Tian, G.; Li, Z.; Wang, H.; Dulebenets, M.A. Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings. *Adv. Eng. Inform.* **2022**, *52*, 101623.
9. Amiri, A.; Amin, S.H.; Zolfaghariania, H. A bi-objective green vehicle routing problem with a mixed fleet of conventional and electric trucks: Considering charging power and density of stations. *Expert Syst. Appl.* **2023**, *213*, 119228.
10. Mancini, S. A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: Formulation and adaptive large neighborhood search based matheuristic. *Transp. Res. Part C Emerg. Technol.* **2016**, *70*, 100–112.
11. Shi, W.; Wang, N.; Zhou, L.; He, Z. The bi-objective mixed-fleet vehicle routing problem under decentralized collaboration and time-of-use prices. *Expert Syst. Appl.* **2025**, *273*, 126875.
12. Zhang, H.; Pan, C.; Gen, M. A robust bi-objective optimization model and heuristic solution for truck-drone collaboration in humanitarian logistics under truck travel time uncertainty. *Swarm Evol. Comput.* **2026**, *101*, 102309.
13. Liao, S.; Xu, Y.; Niu, Y.; Cao, Z. Learning-guided bi-objective evolutionary optimization for green municipal waste collection vehicle routing. *J. Clean. Prod.* **2025**, *501*, 145316.
14. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197.
15. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731.
16. Vieira, Y.E.M.; de Mello Bandeira, R.A.; da Silva Júnior, O.S. Multi-depot vehicle routing problem for large scale disaster relief in drought scenarios: The case of the Brazilian northeast region. *Int. J. Disaster Risk Reduct.* **2021**, *58*, 102193.
17. Wang, Y.; Wang, X.; Fan, J.; Wang, Z.; Zhen, L. Emergency logistics network optimization with time window assignment. *Expert Syst. Appl.* **2023**, *214*, 119145.
18. Lin, P.; Chen, Z.; Pei, M.; Ding, Y.; Qu, X.; Zhong, L. Multiple emergency vehicle priority in a connected vehicle environment: A cooperative method. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 173–188.
19. Calamoneri, T.; Corò, F.; Mancini, S. Management of a post-disaster emergency scenario through unmanned aerial vehicles: Multi-depot multi-trip vehicle routing with total completion time minimization. *Expert Syst. Appl.* **2024**, *251*, 123766.
20. Faiz, T.I.; Vogiatzis, C.; Noor-E-Alam, M. Computational approaches for solving two-echelon vehicle and UAV routing problems for post-disaster humanitarian operations. *Expert Syst. Appl.* **2024**, *237*, 121473.
21. Wei, X.; Qiu, H.; Wang, D.; Duan, J.; Wang, Y.; Cheng, T. An integrated location-routing problem with post-disaster relief distribution. *Comput. Ind. Eng.* **2020**, *147*, 106632.
22. Chen, L.; Chen, E.; Li, R.; Zhou, Z.; Sun, W.; Shi, J. The scheduling problem with delivery and observation in collaboration after the disaster. *Swarm Evol. Comput.* **2025**, *97*, 102047.
23. Sadeghi, A.; Mosadegh, H.; Younessinaki, R.; Aros-Vera, F. A scenario-based robust optimisation approach for social cost vehicle routing problem in disaster response. *Int. J. Prod. Res.* **2026**, *1–23*. doi:10.1080/00207543.2026.2651396.
24. Liu, Q.; Xu, P.; Wu, Y.; Shen, T. Efficient Post-Disaster Emergency Energy Delivery Under Dynamic Constraints: An Electric Vehicle Routing Approach. *IEEE Trans. Intell. Transp. Syst.* **2026**, *27*, 3986–4003.
25. Ardeh, M.A.; Mei, Y.; Zhang, M.; Yao, X. Knowledge transfer genetic programming with auxiliary population for solving uncertain capacitated arc routing problem. *IEEE Trans. Evol. Comput.* **2023**, *27*, 311–325.
26. Jain, H.; Deb, K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Trans. Evol. Comput.* **2014**, *18*, 602–622.
27. Han, F.; Zhu, Y. A two-stage adaptive variable neighborhood search approach for electric vehicle routing problem with fuzzy demand. *Swarm Evol. Comput.* **2026**, *102*, 102332.
28. Melián-Batista, B.; De Santiago, A.; AngelBello, F.; Alvarez, A. A bi-objective vehicle routing problem with time windows: A real case in Tenerife. *Appl. Soft Comput.* **2014**, *17*, 140–152.
29. Liang, X.; Wang, N.; Zhang, M.; Jiang, B. Bi-objective multi-period vehicle routing for perishable goods delivery considering customer satisfaction. *Expert Syst. Appl.* **2023**, *220*, 119712.
30. Zhao, W.; Yuan, Y.; Cheng, C.; Liu, W. Bi-objective sustainable urban logistics vehicle routing problem with workload balance. *J. Ind. Inf. Integr.* **2025**, *48*, 100985.
31. Wang, Y.; Liu, J.; Zhao, X.; Wei, Y.; Zhen, L.; Zhou, J. A clustering-based improved simulated annealing algorithm for collaborative logistics vehicle routing problem with time windows and demand priority rules. *Swarm Evol. Comput.* **2026**, *104*, 102388.
32. Leelertkij, T.; Buddhakulsomsiri, J.; Huynh, V.N. A multi-thread simulated annealing for multi-objective vehicle routing problem with time windows and demand priority. *Comput. Ind. Eng.* **2025**, *207*, 111253.

33. Nourmohammadi, Z.; Hu, B.; Rey, D.; Saberi, M. A data-driven preference learning approach for multi-objective vehicle routing problems in last-mile delivery. *Transp. Res. Part C Emerg. Technol.* **2025**, *174*, 105101.
34. Nyako, S.I.; Tayachi, D.; Abdelaziz, F.B. Machine learning multi-objective optimization for time-dependent green vehicle routing problem. *Energy Econ.* **2025**, *148*, 108628.
35. Hou, Y.; Shen, Y.; Han, H.; Wu, Y.; Huang, Y. Adaptive constrained multi-objective differential evolution algorithm for vehicle routing problem considering crowdsourcing delivery. *Appl. Soft Comput.* **2025**, *169*, 112517.
36. Cai, Y.; Zheng, H.; Liang, B.; Zhou, Y.; Tian, H. Constrained multi-objective evolutionary algorithm with an infeasibility-aided strategy for the vehicle routing problem with time windows. *Appl. Soft Comput.* **2026**, *190*, 114597.
37. Duan, H.; Li, X.; Feng, Y.; Zhang, G.; Lu, Q.; Meng, X. Multi-objective hyper heuristic memetic algorithm for time-dependent vehicle routing problem with drone considering simultaneous pickup and delivery. *Expert Syst. Appl.* **2026**, *297*, 129196.
38. Mahmoodi, A.; Sajadi, S.M.; Laliberte, J.; Easa, S.M. Environmental sustainability and risk-aware optimization in hybrid truck-drone logistics: A holistic multi-objective framework. *J. Air Transp. Manag.* **2026**, *134*, 102978.
39. Vidal, T.; Crainic, T.G.; Gendreau, M.; Prins, C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Comput. Oper. Res.* **2013**, *40*, 475–489.
40. Vidal, T. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Comput. Oper. Res.* **2022**, *140*, 105643.
41. Gurobi Optimization, L. Gurobi Optimizer Reference Manual, 2026.
42. Wouda, N.A.; Lan, L.; Kool, W. PyVRP: A high-performance VRP solver package. *INFORMS J. Comput.* **2024**, *36*, 943–955.

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.