Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Multi-Objective Path Optimization for Truck–Drone Collaborative Delivery Based on Subspace Contribution-Driven Adaptive Resource Allocation

yuehua liao
Hebei University of Technology
jia zhao
zhaojia@hebut.edu.cn

Hebei University of Technology

Research Article

Keywords: truck-drone collaborative delivery, multi-objective path optimization, adaptive resource allocation, subspace contribution, RC-NSGA-II, flexible time window, preference decision

Posted Date: September 7th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10746044/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Multi-Objective Path Optimization for Truck–Drone Collaborative Delivery Based on Subspace Contribution-Driven Adaptive Resource Allocation

Yuehua Liao Jia Zhao

School of Sciences, Hebei University of Technology, Tianjin 300401, China

Corresponding author: Jia Zhao (zhaojia@hebut.edu.cn)

Abstract: Truck-drone collaborative delivery can improve last-mile operations, but its optimization is complicated by tightly coupled vehicle schedules, a high-dimensional combinatorial search space, and conflicting objectives. We formulate a three-objective model for a single truck and multiple drones under flexible time windows, minimizing delivery cost, makespan, and customer dissatisfaction. To allocate search effort more effectively in this discrete space, we propose RC-NSGA-II. The method partitions the normalized objective space into K subspaces by K-means clustering, updates each subspace's recent contribution to the Pareto front through exponential decay, and assigns local-search effort probabilistically in favor of high-contribution regions while retaining uniform exploration. Random-key encoding and greedy decoding connect continuous evolutionary operators with feasible truck-drone plans, and a decaying-perturbation neighborhood search refines selected solutions. An AEW-AHP post-optimization module then recommends solutions for different operating priorities. Across N20W80–N80W80 benchmark instances with 20 independent runs, RC-NSGA-II obtains the highest mean hypervolume (HV) on N20W80, N40W80, and N80W80 and is effectively tied with NSGA-II on N60W80. It also achieves a higher mean HV and a lower mean runtime than ODEA-ARA-adapted on all four instances. Set-coverage and ablation results further support the directional dominance of the resulting solution sets and the separate contributions of local search and RC-guided allocation.

Keywords: truck-drone collaborative delivery; multi-objective path optimization; adaptive resource allocation; subspace contribution; RC-NSGA-II; flexible time window; preference decision

1 Introduction

The rapid growth of e-commerce and rising expectations for timely delivery have made urban last-mile logistics a major source of cost and operational complexity. Last-mile activities are estimated to account for approximately 28%–53% of total logistics costs. As delivery density and service requirements increase, truck-only distribution systems face growing difficulty in balancing timeliness, cost, and service quality [1–4].

Truck-drone collaborative delivery offers a promising alternative. In the Truck-Drone Routing Problem (TDRP), a truck provides high payload capacity, long range, and mobile launch and recovery support, while drones serve selected customers through rapid point-to-point flights. The truck visits a sequence of ground nodes and may launch or recover drones at designated stops; each drone serves a customer and subsequently rejoins the truck. This operating mode can reduce last-mile operating costs by approximately 20%–40% in suitable settings [1, 3, 5, 6]. Its feasibility, however, depends on tight temporal synchronization between truck movements and drone sorties.

The growing low-altitude economy and the commercial deployment of delivery drones further motivate research on coordinated truck-drone routing. Pilot operations by major logistics and on-demand delivery providers in China indicate that drone-assisted delivery is moving from small demonstrations toward routine service. This transition increases the practical need for optimization methods that can represent realistic operating constraints while remaining computationally tractable.

Three features make the TDRP particularly challenging. First, heterogeneous-vehicle coupling links the truck route and drone sorties through launch, recovery, and synchronization constraints; modifying either component can invalidate the other. Second, conflicting objectives require decision makers to balance operating cost, completion time, and service quality rather than optimize a single measure. Third, the combined customer-assignment and sequencing decisions produce a high-dimensional discrete search space that grows rapidly with instance size, making exact optimization increasingly difficult for medium- and large-scale problems.

Existing solution methods fall broadly into three categories [7]. Exact methods, including mixed-integer linear programming and branch-and-bound or branch-price-and-cut frameworks, can establish optimality for small instances but become computationally expensive as the problem grows [8, 9]. Heuristics and metaheuristics provide a more practical balance between solution quality and runtime for larger instances, although many studies optimize only cost or time and therefore do not capture the full trade-off structure [5, 10]. Multi-objective evolutionary algorithms (MOEAs) instead generate a set of Pareto-optimal solutions in a single run and have consequently attracted increasing attention for multi-objective TDRP variants [11–14].

Among MOEAs, NSGA-II is widely used because of its fast non-dominated sorting and crowding-distance selection [15]. Nevertheless, standard NSGA-II allocates search ef-

fort relatively uniformly across the objective space. In high-dimensional discrete problems, this strategy may spend substantial effort in low-potential regions while population diversity declines and the search converges prematurely. Previous work has improved NSGA-II through diversity preservation and local search [16, 17], but adaptive allocation of search resources has received less attention in truck-drone routing with tightly coupled temporal constraints.

ODEA-ARA addresses adaptive search by allocating resources according to improvement and contribution indicators [13]. However, the original method was designed for a simultaneous delivery-and-pickup problem and uses a specialized representation together with seven problem-specific neighborhoods. Its published numerical results are therefore not directly comparable with those obtained under the model considered here.

To address this gap, we propose RC-NSGA-II, a lightweight adaptive evolutionary algorithm driven by subspace contribution. K-means clustering partitions the normalized objective space, while an exponentially decayed contribution measure records the recent search value of each subspace. Local-search resources are then assigned probabilistically, favoring high-contribution regions while retaining a uniform exploration component. Unlike the RI–RC dual-indicator strategy, RC-NSGA-II uses contribution with decayed memory as its sole allocation signal, avoiding the need to maintain a separate push-distance improvement measure.

The main contributions of this study are as follows:

1. We formulate a three-objective mixed-integer model for single-truck, multi-drone delivery with flexible time windows. The model jointly minimizes delivery cost, makespan, and customer dissatisfaction while explicitly representing launch-recovery synchronization and heterogeneous service capabilities.
2. We develop RC-NSGA-II, which combines random-key encoding, greedy decoding, contribution-driven adaptive resource allocation, and decaying-perturbation local search to improve the allocation of search effort in a high-dimensional discrete space.
3. We integrate an AEW-AHP post-optimization decision module that recommends interpretable solutions for cost-, time-, and satisfaction-oriented operating scenarios.
4. We evaluate the proposed method over four instance scales using 20 independent runs and compare it with five established MOEAs and ODEA-ARA-adapted in terms of hypervolume, set coverage, and runtime.

2 Related Work

2.1 Truck-Drone Collaborative Delivery

Murray and Chu [1] introduced the Flying Sidekick Traveling Salesman Problem (FSTSP), in which a truck and a drone coordinate to minimize delivery completion time. Agatz et al. [2] subsequently formulated the Traveling Salesman Problem with Drone (TSP-D) and developed exact and heuristic solution approaches that account for operating cost and route synchronization. Ha et al. [5] incorporated drone-endurance constraints into a cost-minimization model, bringing the formulation closer to practical operating conditions.

Later studies extended truck-drone routing in several directions. Poikonen et al. [10] considered operations with multiple drones, whereas Wang et al. [6] examined synchronization under time-window constraints. Yin et al. [8] developed a branch-price-and-cut algorithm for truck-drone routing with time windows; their general MILP solved instances with up to 10 customers, while the decomposition framework supported larger cases. Xing et al. [12] addressed reliable routing under dynamic synchronization through a high-dimensional network-programming formulation. Ge et al. [11] proposed a multi-objective model with customer classification and fuzzy time windows, optimizing both distribution cost and customer satisfaction. These studies provide the methodological basis for the flexible-time-window and satisfaction components considered in this paper.

Recent research has also begun to address uncertainty and larger-scale operations. Representative topics include stochastic demand [18], stochastic and robust routing with deadline constraints [19], and robust truck-drone routing under uncertain demand [20]. Collectively, these developments reflect a shift from stylized deterministic formulations toward more operationally realistic settings.

2.2 Multi-Objective Evolutionary Algorithms

MOEAs are well suited to discrete routing problems because they do not require convex objective functions and can approximate an entire Pareto front in a single run. NSGA-II combines fast non-dominated sorting with crowding-distance selection and remains a standard baseline for multi-objective combinatorial optimization [15]. SPEA2 uses dominance strength and density information to preserve diversity [21]. MOEA/D decomposes a multi-objective problem into a set of scalar subproblems that are optimized cooperatively [22], although its performance may depend on how well predefined weight vectors represent an irregular Pareto front.

Research on improving NSGA-II has primarily focused on diversity preservation, hybrid local search, and adaptive allocation of search effort. Tian et al. [16] reviewed diversity-maintenance strategies for large-scale multi-objective optimization, while Liu et al. [17] summarized approaches that combine learnable evolutionary operators with local search. These lines of work show that both population structure and targeted refinement can materially affect convergence in large discrete spaces.

Adaptive resource allocation provides a complementary mechanism. ODEA-ARA concentrates search effort in promising subregions using relative improvement (RI) and relative contribution (RC), together with seven problem-specific neighborhoods [13]. Because its original delivery-and-pickup formulation and solution representation differ from those used here, we construct ODEA-ARA-adapted for a mechanism-level comparison. The adapted method retains direction-vector subspaces, minimum-angle association, RI/RC sliding memories, and probabilistic resource allocation, while using the same encoding, decoder, local-search interface, population size, and computational budget as the other algorithms. Adaptive operator selection based on deep reinforcement learning

has also advanced this area [23]; nevertheless, lightweight mechanisms that require limited hyperparameter tuning remain attractive for practical routing applications.

2.3 Preference-Based Decision Making

Multi-objective optimization produces a set of non-dominated solutions rather than a single operational decision. A decision-support procedure is therefore required to select a solution consistent with the decision maker's priorities. The entropy weight method (AEW) estimates objective weights from the information carried by the observed data, whereas the analytic hierarchy process (AHP) represents subjective preferences through Saaty's pairwise-comparison matrix [24]. Combining AEW and AHP balances data-driven discrimination with scenario-specific judgment, providing an interpretable basis for selecting representative solutions from the Pareto set.

3 Problem Formulation

3.1 Problem Description

We consider a single-truck, multi-drone last-mile delivery problem with flexible time windows (TDRP-FTW). A truck carrying multiple drones departs from a distribution center, serves a set of customers with known locations, demands, and time windows, and returns to the distribution center after all deliveries have been completed.

Figure 1 illustrates the coordination mechanism. The truck follows a ground route and may launch a drone at a visited node. The drone serves one customer by direct flight and is recovered at a subsequent truck stop. Launch and recovery decisions therefore couple the truck route, drone sorties, and service times. The optimization problem jointly determines (i) the truck's customer sequence, (ii) the customers assigned to drones together with their launch and recovery nodes, and (iii) the temporal synchronization of truck and drone operations. These decisions are optimized with respect to delivery cost, makespan, and customer dissatisfaction.

3.2 Model Assumptions

The model is based on the following assumptions. Their implications for external validity are discussed in Section 7.

a) *Known customers and unique service:* Customer locations, demands, and time windows are known, and each customer is served exactly once by either the truck or a drone.

b) *Single-customer drone sorties:* Each sortie carries one parcel and serves one customer. The assigned demand must not exceed the drone payload Q_U, and the launch-customer-recovery distance must satisfy the endurance limit $E_U = 30\,\text{km}$.

c) *Temporal coordination:* A drone is launched only when the truck is stationary at a visited node and is recovered at a subsequent truck stop. Launch and recovery while the truck is moving are not permitted.

d) *Reusable drone resources:* The truck carries R physical drones. Sorties assigned to the same drone cannot overlap, and a drone becomes available again only after recovery.

e) *Single depot:* The distribution center is the unique origin and destination for the truck-drone system.

f) *Constant travel speeds:* The truck and drones travel at constant speeds V_T and V_D, respectively. Drone distance is Euclidean, whereas truck distance is approximated by Manhattan distance.

g) *Negligible auxiliary time:* Launch, recovery, and battery-replacement times are included in the truck service time and are not modeled separately.

h) *Safe waiting:* If a drone reaches the recovery node before the truck, it may wait in a designated safe area until the truck arrives.

The Manhattan-distance approximation may not fully reproduce travel on an actual urban road network. In addition, omitting explicit launch, recovery, and battery-handling times may overstate the operational speed advantage of drones, particularly in node-dense settings. These limitations could be reduced by using a road-network distance matrix and empirically calibrated auxiliary-time parameters.

3.3 Notation

The main sets, parameters and decision variables are shown in Table 1.

3.4 Objective Functions

Goal 1 - Minimize total distribution cost:

$$\begin{aligned}\min f_1 = {} & c_T \sum_{i\in V}\sum_{j\in V} d^T_{ij} y_{ij} \\ & + c_U \sum_{s\in S}\sum_{i\in C}\sum_{j\in C^U}\sum_{k\in C} \left(d^U_{ij} + d^U_{jk}\right) x_{ijks} \\ & + C_F \sum_{s\in S} v_s\end{aligned} \tag{1}$$

The three terms represent truck travel cost, drone flight cost, and the fixed launch-and-recovery cost of activated sorties, respectively.

Goal 2 - Minimize total completion time (makespan):

$$\min f_2 = \max\left(T_n,\ \max_{s\in S} H_s\right) \tag{2}$$

Goal 3 - Minimize customer dissatisfaction:

$$\begin{aligned}\min f_3 = {} & \sum_{j\in C} z^T_j\, u_j(T_j) \\ & + \sum_{s\in S}\sum_{i\in C}\sum_{j\in C^U}\sum_{k\in C} x_{ijks}\, u_j\left(L_s + 60\frac{d^U_{ij}}{V_D}\right)\end{aligned} \tag{3}$$

Because distances are measured in kilometers, speeds in kilometers per hour, and all scheduling variables in minutes, each travel-time term is multiplied by 60 to maintain dimensional

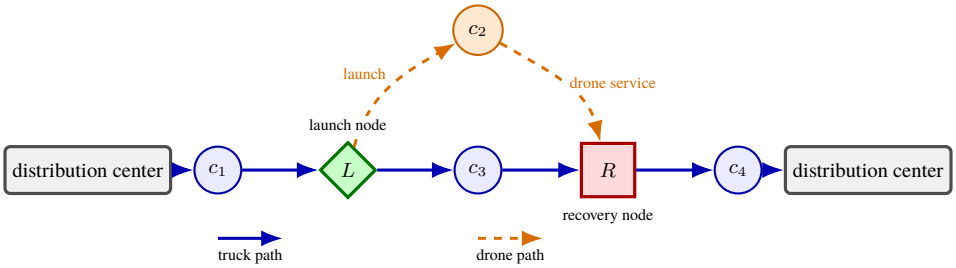


Figure 1: Schematic diagram of truck-drone collaborative delivery path. The truck visits customers along the solid line and passes through the launch node L and recovery node R; the drone takes off from L, serves customer c_2, and recovers at R.

Table 1: Symbol explanation

Symbol	Definition
Sets and Indices	
$V = \{0, 1, \ldots, n\}$	node set, 0 is the starting point of the warehouse, n is the virtual end warehouse
$C = V \setminus \{0, n\}$	set of customer nodes
$C^T \subseteq C$	customers restricted to truck service because demand exceeds drone capacity
$C^U = C \setminus C^T$	customers eligible for truck or drone service
$D = \{1, \ldots, R\}$	set of physical drones carried by the truck
$S = \{1, \ldots, S_{\max}\}$	set of potential drone sorties
Parameters	
$V_T,\ V_D$	truck/drone travel speed (km/h)
$Q_T,\ Q_U$	truck/drone payload capacity (kg)
q_i	demand of customer i (kg)
$[e_i, l_i]$	time window of customer i (min)
$s_i^T,\ s_i^U$	truck/drone service time at customer i (min)
$c_T,\ c_U$	truck/drone travel cost per unit distance (yuan/km)
C_F	fixed launch-and-recovery cost per drone sortie (yuan)
$\alpha,\ \beta$	early/late service penalty coefficients ($\alpha = 0.2,\ \beta = 1.0$)
R	number of physical drones carried by the truck
E_U	maximum drone-sortie range (30 km)
M	sufficiently large positive constant
Decision Variables	
$y_{ij} \in \{0, 1\}$	1 if the truck travels directly from i to j
$z_i^T \in \{0, 1\}$	1 if customer i is served by the truck
$x_{ijks} \in \{0, 1\}$	1 if sortie s is launched at i, serves j, and is recovered at k
$a_{sd} \in \{0, 1\}$	1 if sortie s is assigned to physical drone d
$v_s \in \{0, 1\}$	1 if sortie s is activated
$g_{ss'd} \in \{0, 1\}$	1 if drone d executes sortie s before s'
$T_i,\ L_s,\ H_s$	truck arrival time at i and launch/recovery times of sortie s (min)
$w_i^T \geq 0$	truck waiting time at node i (min)
u_i	truck visit-order variable for node i

consistency. The dissatisfaction incurred when customer j is served at time t is defined as

$$u_j(t) = \begin{cases} \alpha(e_j - t), & t < e_j \\ 0, & e_j \le t \le l_j \\ \beta(t - l_j), & t > l_j \end{cases} \tag{4}$$

$\alpha = 0.2$ assigns a relatively small penalty to early service, whereas $\beta = 1.0$ imposes a larger penalty on late service.

3.5 Constraints

Unique service constraints

$$z_j^T + \sum_{s\in S}\sum_{i\in C}\sum_{k\in C} x_{ijks} = 1, \qquad j \in C^U \tag{5}$$

$$z_j^T = 1, \qquad j \in C^T \tag{6}$$

Constraints (5)–(6) ensure that every customer is served exactly once. Drone-eligible customers may be assigned to either vehicle type, whereas customers in C^T must be served by the truck.

Truck routing and subtour-elimination constraints

$$\sum_{j\in V\setminus\{0\}} y_{0j} = 1 \tag{7}$$

$$\sum_{i\in V\setminus\{n\}} y_{in} = 1 \tag{8}$$

$$\sum_{i\in V\setminus\{j\}} y_{ij} = \sum_{k\in V\setminus\{j\}} y_{jk} = z_j^T, \quad j \in C \tag{9}$$

$$1 \le u_i \le n, \quad i \in C \tag{10}$$

$$u_i - u_j + n y_{ij} \le n - 1, \quad i \ne j,\ i,j \in C \tag{11}$$

Because both launch and recovery locations must be visited by the truck, constraint (9) uses z_j^T to indicate whether customer node j is included in the truck route. Constraints (10)–(11) are Miller-Tucker-Zemlin conditions that eliminate subtours disconnected from the depot.

Flight activation and physical drone allocation constraints

$$v_s = \sum_{i\in C}\sum_{j\in C^U}\sum_{k\in C} x_{ijks} \le 1, \quad s \in S \tag{12}$$

$$\sum_{d\in D} a_{sd} = v_s, \quad s \in S \tag{13}$$

Each potential sortie serves at most one customer. Every activated sortie is assigned to exactly one physical drone, whereas inactive sorties consume no drone resources.

Launch-recovery and visit-order constraints Let $r_i = \sum_{j\in V} y_{ij}$ indicate whether the truck visits node i. Then

$$x_{ijks} \le r_i, \quad x_{ijks} \le r_k, \quad i,k \in C,\ j \in C^U,\ s \in S \tag{14}$$

$$u_i + 1 \le u_k + M(1 - x_{ijks}), \quad i,k \in C,\ j \in C^U,\ s \in S \tag{15}$$

Constraint (14) requires both endpoints of a sortie to lie on the truck route, while constraint (15) requires the launch node to precede the recovery node.

Load and endurance constraints

$$\sum_{j\in C} q_j z_j^T \le Q_T \tag{16}$$

$$q_j x_{ijks} \le Q_U, \quad i,k \in C,\ j \in C^U,\ s \in S \tag{17}$$

$$\left(d_{ij}^U + d_{jk}^U\right) x_{ijks} \le E_U, \quad i,k \in C,\ j \in C^U,\ s \in S \tag{18}$$

If $q_j > Q_U$, all corresponding x_{ijks} variables are fixed to zero. Constraint (18) applies the endurance limit to the complete launch-customer-recovery sortie.

Truck time recursion and launch synchronization constraints

$$T_j \ge T_i + s_i^T z_i^T + 60\frac{d_{ij}^T}{V_T} + w_i^T - M(1 - y_{ij}), \quad i,j \in V \tag{19}$$

$$L_s \ge T_i - M(1 - x_{ijks}), \quad i,k \in C,\ j \in C^U,\ s \in S \tag{20}$$

$$L_s \le T_i + M(1 - x_{ijks}), \quad i,k \in C,\ j \in C^U,\ s \in S \tag{21}$$

Because launch preparation time is included in the service time, constraints (20)–(21) enforce $L_s = T_i$ for every activated sortie.

Drone recovery with truck waiting constraints

$$H_s \ge L_s + 60\frac{d_{ij}^U}{V_D} + s_j^U + 60\frac{d_{jk}^U}{V_D} - M(1 - x_{ijks}), \quad i,k \in C,\ j \in C^U,\ s \in S \tag{22}$$

$$T_k + w_k^T \ge H_s - M(1 - x_{ijks}), \quad i,k \in C,\ j \in C^U,\ s \in S \tag{23}$$

$$w_k^T \ge H_s - T_k - M(1 - x_{ijks}), \quad w_k^T \ge 0 \tag{24}$$

If a drone reaches the recovery node after the truck, constraints (23)–(24) add the delay to the truck waiting time. The truck may depart only after recovery is complete.

Non-overlap constraints for sorties assigned to the same drone For two distinct sorties $s, s' \in S$ assigned to drone $d \in D$, let $g_{ss'd}$ denote their execution order:

$$L_{s'} \ge H_s - M(1 - g_{ss'd}) - M(2 - a_{sd} - a_{s'd}) \tag{25}$$

$$L_s \ge H_{s'} - M g_{ss'd} - M(2 - a_{sd} - a_{s'd}), \quad s < s',\ d \in D \tag{26}$$

When both sorties are assigned to drone d, constraints (25)–(26) impose a valid execution order and prevent temporal overlap. Together with constraint (13), they ensure that the number

of concurrent sorties does not exceed $|D| = R$ and that each physical drone is reused only after recovery.

4 The RC-NSGA-II Algorithm

4.1 Overall Framework

RC-NSGA-II extends NSGA-II with three components. First, random-key encoding and greedy decoding connect continuous evolutionary operators with the discrete truck-drone routing structure. Second, the RC mechanism allocates local-search effort according to the recent contribution of objective-space subregions. Third, a decaying-perturbation neighborhood search refines selected solutions while gradually shifting from exploration to exploitation. Algorithm 1 summarizes the complete procedure.

Algorithm 1 The RC-NSGA-II procedure

Input: Customer set C; population size N; generations G; subspaces K; decay factor λ; allocation weight γ
Output: Final non-dominated solution set $\mathcal{F}_1$

```
Generate an initial random-key population P_0 of size N
Decode and evaluate F(x) = (f_1, f_2, f_3) for every x ∈ P_0
Initialize RC_k^(0) = 0 for k = 1, ..., K
for t = 1, 2, ..., G do
    Generate offspring Q_t by tournament selection, SBX crossover, and polynomial mutation
    Set R_t = P_t ∪ Q_t and perform fast non-dominated sorting
    Select P_{t+1} using Pareto rank and crowding distance
    Partition P_{t+1} by K-means and match subspaces across generations
    Update RC_{π(k)}^(t) = λ RC_k^(t-1) + cnt_{π(k)}^(t)
    Sample L_s = 5 subspaces according to p_k and apply decaying-perturbation local search
    Replace a selected solution if the perturbed solution strictly dominates it
end for
return F_1
```

4.2 Random-Key Encoding and Greedy Decoding

Random-key encoding maps routing priorities to a continuous search space:

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n), \quad x_i \in [0, 1] \tag{27}$$

Here, x_i is the priority key associated with customer i. Sorting the keys in ascending order determines the initial truck sequence. For example, $\boldsymbol{x} = (0.32, 0.85, 0.41, 0.12)$ produces the route $0 \to 4 \to 1 \to 3 \to 2 \to 0$.

The resulting sequence is converted into a feasible truck-drone plan by the following deterministic greedy decoder:

1. Identify drone-eligible customers using payload and endurance limits.
2. Traverse eligible customers in truck-route order and enumerate launch-recovery pairs that respect the visit sequence.
3. Assign the earliest available physical drone to each candidate sortie.
4. Retain truck service when all R drones are occupied.
5. Recompute truck arrival times and drone launch and recovery times after inserting a sortie.
6. Add truck waiting time whenever a drone arrives later than the truck at the recovery node.
7. Accept only solutions satisfying unique service, capacity, endurance, visit-order, and non-overlap constraints.

The chromosome determines customer priority, whereas the decoder determines truck-drone assignment and launch-recovery structure. Consequently, the algorithm searches the subset of feasible plans reachable through the decoder rather than explicitly enumerating the complete feasible region of the mathematical model. The continuous representation allows standard simulated binary crossover (SBX) and polynomial mutation to operate directly on the chromosome, avoiding additional route-specific crossover operators.

4.3 Contribution-Driven Adaptive Resource Allocation

4.3.1 Objective-Space Partitioning

At each generation, the three objective values of population P_{t+1} are first normalized by min-max scaling:

$$F_i^{\text{norm}} = \frac{F_i - F^{\min}}{F^{\max} - F^{\min}} \tag{28}$$

K-means then partitions the normalized objective vectors into K subspaces $\mathcal{S}_1, \ldots, \mathcal{S}_K$. We use $K = 7$; Section 6.5 examines sensitivity to this choice. Because cluster labels are arbitrary and may change between generations, the previous centers $\boldsymbol{\mu}_k^{(t-1)}$ and current centers $\boldsymbol{\mu}_l^{(t)}$ are matched by minimizing their total Euclidean distance:

$$\pi^{(t)} = \arg\min_{\pi} \sum_{k=1}^{K} \left\| \boldsymbol{\mu}_k^{(t-1)} - \boldsymbol{\mu}_{\pi(k)}^{(t)} \right\|_2 . \tag{29}$$

This matching preserves the identity of comparable objective-space regions and allows their contribution histories to be updated consistently.

4.3.2 Recursive Contribution Update

Let $RC_k^{(t-1)}$ denote the previous contribution of subspace k, and let $\text{cnt}_{\pi(k)}^{(t)}$ be the number of newly generated non-dominated solutions in its matched current subspace. The contribution is updated as

$$RC_{\pi(k)}^{(t)} = \lambda RC_k^{(t-1)} + \text{cnt}_{\pi(k)}^{(t)} \tag{30}$$

The decay factor $\lambda \in (0, 1)$ controls how quickly past evidence is forgotten; we set $\lambda = 0.7$. A large $RC_k^{(t)}$ indicates that the corresponding region has recently produced new non-dominated solutions and may therefore benefit from additional local search. The recursive memory smooths generation-to-generation noise, while exponential decay prevents early improvements from dominating the allocation indefinitely.

Unlike the RI–RC allocation mechanism in ODEA-ARA [13], the proposed method uses only the contribution of newly generated non-dominated solutions. It therefore does not compute push distance during resource allocation. The recursive update costs $O(K)$; the complete RC module additionally requires objective normalization, K-means clustering, and inter-generational cluster matching. Their combined overhead is reflected in the reported runtimes.

4.3.3 Adaptive Search Probabilities

The probability of selecting subspace k for local search is

$$p_k = \gamma \cdot \frac{RC_k^{(t)}}{\sum_{k=1}^{K} RC_k^{(t)}} + (1-\gamma) \cdot \frac{1}{K} \tag{31}$$

The parameter $\gamma \in (0,1)$ balances exploitation and exploration; we use $\gamma = 0.7$. The first term allocates probability in proportion to recent contribution, whereas the second assigns every subspace a uniform baseline probability. Thus, high-potential regions receive more search effort without permanently excluding low-contribution regions.

Design motivation. The allocation rule is analogous to a multi-armed bandit: the K subspaces act as arms, and $RC_k^{(t)}$ serves as a recency-weighted reward signal [25, 26]. Exponential decay adapts the signal as the search landscape changes, while the uniform term $(1-\gamma)/K$ maintains exploration [27]. This analogy motivates the design but is not presented as a formal bandit derivation. Section 6.4 evaluates the empirical contribution of the RC mechanism.

4.4 Decaying-Perturbation Local Search

We sample $L_s = 5$ subspaces by roulette-wheel selection using p_k; repeated selections are allowed. One solution $\boldsymbol{x} = (x_1, \ldots, x_D)$ is selected from each sampled subspace and perturbed componentwise:

$$x_i' = x_i + r_t \cdot \delta_i, \quad \delta_i \sim \mathcal{U}(-1,1) \tag{32}$$

The perturbation radius decreases exponentially with generation t:

$$r_t = r_0 \cdot 0.95^t, \quad r_0 = 0.15 \tag{33}$$

This schedule supports broader exploration early in the run and finer local refinement later. The decoder checks the feasibility of each perturbed solution $\boldsymbol{x}'$. If $\boldsymbol{x}'$ is feasible and strictly dominates the original solution, i.e., $F(\boldsymbol{x}') \prec F(\boldsymbol{x})$, it replaces the original; otherwise, the original solution is retained.

5 AEW-AHP Preference-Based Decision Making

The optimizer returns a Pareto set $\mathcal{F}_1$ rather than a single implementable plan. We therefore combine the entropy weight method (AEW) with the analytic hierarchy process (AHP) to rank non-dominated solutions under different operating priorities. AEW captures the discriminatory information in the observed objective values, while AHP represents scenario-specific managerial preferences.

5.1 AEW Objective Weights

Suppose $\mathcal{F}_1$ contains m candidate solutions evaluated on the three minimization objectives f_1, f_2, f_3. The objective values are first transformed into benefit-type normalized scores:

$$Z_{ij} = \frac{f_j^{\max} - f_{ij}}{f_j^{\max} - f_j^{\min}} \tag{34}$$

Column normalization yields p_{ij}. The entropy E_j and information-utility coefficient D_j of objective j are then

$$E_j = -\frac{1}{\ln m} \sum_{i=1}^{m} p_{ij} \ln p_{ij}, \qquad D_j = 1 - E_j \tag{35}$$

The AEW weight is

$$W_j^{\mathrm{AEW}} = \frac{D_j}{\sum_{j=1}^{3} D_j} \tag{36}$$

5.2 AHP Subjective Weights

For each preference scenario, a judgment matrix $A = (a_{ij})_{3\times 3}$ is constructed using Saaty's 1–9 scale [24]. The principal eigenvector gives W^{AHP}, and judgment consistency is evaluated by

$$CR = \frac{\lambda_{\max} - 3}{2 \times RI} \tag{37}$$

For a 3×3 matrix, $RI = 0.58$; the judgment matrix is accepted when $CR < 0.1$.

5.3 Combined Weights and Solution Recommendation

The subjective and objective weights are combined as

$$W_j = \theta \cdot W_j^{\mathrm{AHP}} + (1-\theta) \cdot W_j^{\mathrm{AEW}}, \quad \theta = 0.7 \tag{38}$$

Each candidate solution receives the score

$$S_i = \sum_{j=1}^{3} W_j Z_{ij} \tag{39}$$

The recommended solution is $\boldsymbol{x}^* = \arg\max_i S_i$. We consider three preference scenarios—cost priority, timeliness priority, and satisfaction priority—each represented by a different AHP judgment matrix. The procedure therefore produces scenario-specific recommendations from the same Pareto set.

6 Experimental Design and Results

6.1 Experimental Setup

Test instances. Following the benchmark design of Luo et al. [13], we generate four instance groups from TSPTW data: N20W80, N40W80, N60W80, and N80W80. The first number denotes the number of customers, and W80 denotes the time-window width. Seventy percent of customers are designated as drone eligible to represent parcels that differ in payload or service requirements. Each instance contains customer coordinates, demand, and a time window $[e_i, l_i]$. Euclidean distance is used for drone flights, and Manhattan distance approximates truck travel on the road network.

Computational environment and parameters. All experiments were conducted on a computer with an Intel Core i7-14700HX processor, 16 GB of memory, and Windows 11, using Python 3.12 with pymoo, NumPy, and Matplotlib. Each algorithm was run independently 20 times under the same environment and computational budget. Table 2 lists the principal parameter settings.

Performance metrics. Hypervolume (HV) measures both convergence and diversity relative to a fixed reference point [28]; larger values indicate better solution-set quality. Before computing HV, the three objectives are normalized using common lower and upper bounds obtained from all comparison runs for the same instance. Every algorithm is then evaluated in the same normalized objective space with the same reference point.

Set coverage $C(A, B)$ measures directional dominance and is defined as the proportion of solutions in B dominated by at least one solution in A. The spacing metric [29] assesses the uniformity of a non-dominated set through the standard deviation of nearest-neighbor distances:

$$S = \sqrt{\frac{1}{|P|-1}\sum_{i=1}^{|P|}\left(\bar{d}-d_i\right)^2}, \quad d_i = \min_{j\neq i}\left\|\mathbf{f}^{(i)}-\mathbf{f}^{(j)}\right\|_2, \tag{40}$$

where $\bar{d}$ is the mean of d_i and $\mathbf{f}^{(i)}$ is the normalized objective vector of solution i. A smaller S indicates a more uniform distribution.

Comparison algorithms. We compare RC-NSGA-II with six multi-objective baselines: NSGA-II [15], SPEA2 [21], MOEA/D [22], MOPSO [30], IBEA [31], and ODEA-ARA-adapted [13]. The first five represent dominance-, decomposition-, swarm-, and indicator-based paradigms. ODEA-ARA-adapted provides the closest comparison at the resource-allocation level.

ODEA-ARA-adapted is a mechanism-level adaptation rather than a line-by-line reproduction of the original implementation. It retains approximately uniform direction-vector subspaces, minimum-angle association, push-distance-based relative improvement, archive-based relative contribution, length-20 RI/RC memories, equal weighting of RI and RC, and probabilistic resource allocation. To ensure comparability under the present model, it uses the same random-key representation, greedy decoder, perturbation-based local search, population size, number of generations, and local-search budget as RC-NSGA-II. Its external archive is deduplicated and, when larger than $\max(300, 2N)$, truncated by crowding distance. The comparison therefore evaluates resource-allocation mechanisms under a unified implementation; it is not an evaluation of the original ODEA-ARA on its delivery-and-pickup problem.

6.2 Comparison of Optimization Results

6.2.1 HV Convergence

Figure 2 presents the mean HV convergence curves over 20 independent runs. RC-NSGA-II remains among the leading algorithms across all four scales:

- *N20W80 and N40W80:* RC-NSGA-II maintains the highest mean HV in the middle and late stages, indicating that contribution guidance continuously exploits high-potential subspaces as the search progresses.
- *N60W80:* The convergence curves of RC-NSGA-II and NSGA-II are close, with final mean HV values of 0.735 and 0.736, respectively.
- *N80W80:* RC-NSGA-II regains the highest mean HV and maintains stronger convergence quality in the larger search space.

MOEA/D and MOPSO exhibit substantially lower mean HV on the medium- and large-scale instances under the present parameterization. This result is specific to the model and experimental settings used here and should not be interpreted as a general assessment of either framework.

6.2.2 HV and Runtime Comparison

Table 3 reports mean HV, its standard deviation, and mean runtime over 20 independent runs.

RC-NSGA-II achieves the highest mean HV on N20W80, N40W80, and N80W80. On N60W80, its mean HV is 0.735, only 0.001 below NSGA-II. Relative to ODEA-ARA-adapted, the mean-HV gains are 0.005, 0.012, 0.026, and 0.024 for N20W80 through N80W80, respectively. Mean runtime is lower by approximately 20.9%, 16.6%, 3.2%, and 47.7%. Across the four scales, the results indicate that RC-NSGA-II maintains strong solution quality without incurring the runtime of the adapted resource-allocation baseline.

6.2.3 Set-Coverage Analysis

Table 4 shows a consistently asymmetric coverage relationship. For most baselines, $C(\text{RC-NSGA-II}, B)$ exceeds the reverse measure $C(B, \text{RC-NSGA-II})$, indicating that the advantage is not confined to a single comparator. Against ODEA-ARA-adapted, the forward coverage values are 0.733, 0.826, 0.835, and 0.802, compared with reverse values of 0.219, 0.113, 0.152, and 0.178. Together with the HV results, these values show that the RC-guided method produces a more dominant solution set under the unified implementation and experimental conditions.

6.3 Collaborative Route Visualization

Figure 3 visualizes the truck-drone routes associated with the three recommended N20W80 solutions. Section 6.6 compares their objective values and operating trade-offs.

6.4 Ablation Study

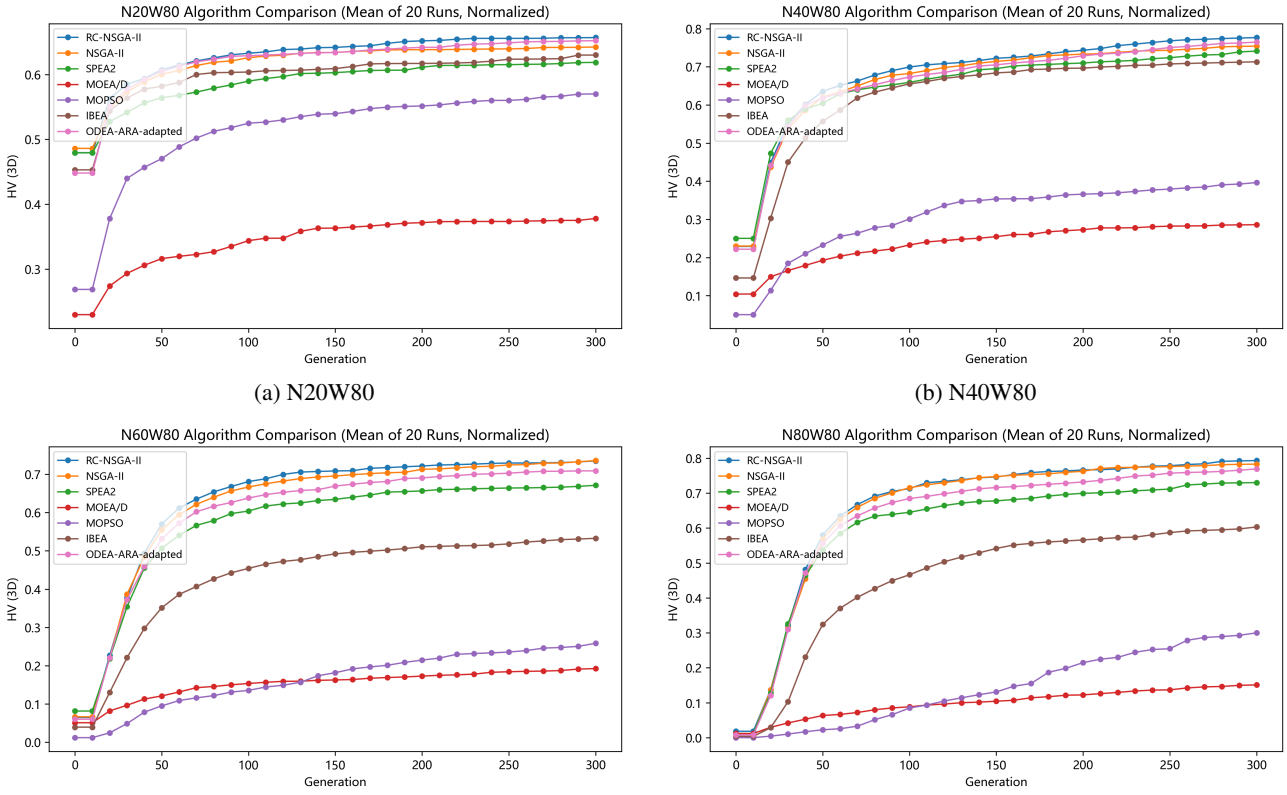
To isolate the contribution of local search and RC-guided allocation, we compare three variants:

- *NSGA-II*: the standard algorithm without local search or RC guidance;
- *NSGA-II-RandLS*: NSGA-II with local search and uniform random subspace selection;
- *RC-NSGA-II*: the complete method with local search and contribution-guided subspace selection.

Table 2: Experimental parameter settings

Category	Parameter	Value
Truck-drone	Truck payload Q_T	300 kg
	Drone payload Q_U	5 kg
	Truck speed V_T	50 km/h
	Drone speed V_D	70 km/h
	Truck travel cost c_T	3 yuan/km
	Drone travel cost c_U	8 yuan/km*
	Fixed sortie cost C_F	10 yuan/sortie
	Truck service time s^T	5 min
	Drone service time s^U	2 min
	Number of drones R	3
Algorithm	Population size N	150
	Maximum generations G	300
	Number of subspaces K	7
	Decay factor λ	0.7
	Allocation weight γ	0.7
	Local-search samples L_s	5
	Initial perturbation radius r_0	0.15
	Crossover probability/distribution index	0.9/20

* The drone unit cost is consistent with the ranges reported by Chiang et al. [3] and Luo et al. [13], approximately 7–10 yuan/km after conversion.



(a) N20W80

(b) N40W80

(c) N60W80

(d) N80W80

Figure 2: HV convergence curves of seven algorithms under different scale instances (average of 20 independent runs)

Table 3: HV and average running time of seven algorithms at different scales (20 independent runs)

	N20W80		N40W80		N60W80		N80W80	
Algorithm	HV (mean ± SD)	time/s	HV (mean ± SD)	time/s	HV (mean ± SD)	time/s	HV (mean ± SD)	time/s
RC-NSGA-II	$\mathbf{0.657 \pm 0.058}$	9.553	$\mathbf{0.777 \pm 0.050}$	12.913	0.735 ± 0.121	17.536	$\mathbf{0.793 \pm 0.104}$	41.992
NSGA-II	0.642 ± 0.050	8.241	0.754 ± 0.078	11.801	$\mathbf{0.736 \pm 0.114}$	17.239	0.783 ± 0.127	41.146
SPEA2	0.619 ± 0.068	7.913	0.741 ± 0.072	11.832	0.671 ± 0.167	17.420	0.730 ± 0.154	40.679
MOEA/D	0.378 ± 0.107	21.667	0.286 ± 0.076	27.802	0.192 ± 0.075	132.480	0.151 ± 0.085	68.028
MOPSO	0.570 ± 0.042	18.677	0.397 ± 0.071	10.635	0.259 ± 0.091	13.830	0.300 ± 0.113	37.950
IBEA	0.630 ± 0.069	6.555	0.713 ± 0.078	13.404	0.533 ± 0.145	18.036	0.603 ± 0.151	52.407
ODEA-ARA-adapted	0.652 ± 0.045	12.074	0.765 ± 0.052	15.477	0.709 ± 0.136	18.108	0.769 ± 0.133	80.260

Table 4: C indicator comparison

	NSGA-II	SPEA2	MOEA/D	MOPSO	IBEA	ODEA-ARA-adapted
C(RC-NSGA-II, $-$)						
N20W80	0.681	0.726	0.894	0.832	0.770	0.733
N40W80	0.694	0.753	0.987	0.806	0.806	0.826
N60W80	0.747	0.674	0.965	0.856	0.865	0.835
N80W80	0.762	0.676	0.954	0.982	0.892	0.802
C($-$, RC-NSGA-II)						
N20W80	0.231	0.401	0.095	0.098	0.173	0.219
N40W80	0.214	0.410	0.084	0.087	0.152	0.113
N60W80	0.187	0.430	0.054	0.072	0.019	0.152
N80W80	0.163	0.407	0.067	0.034	0.072	0.178

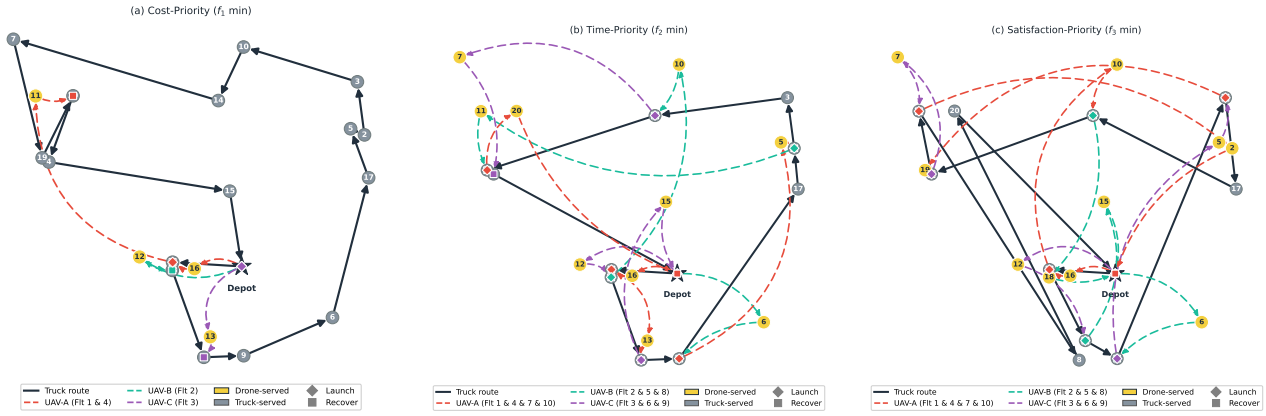


Figure 3: Truck-drone routes for three N20W80 preference scenarios: (a) cost priority (4 drone sorties), (b) timeliness priority (10 sorties), and (c) satisfaction priority (10 sorties). Solid lines denote the truck route, dashed arcs denote drone flights, and L and R denote launch and recovery nodes. Yellow nodes are drone served and gray nodes are truck served. Coordinates are obtained by multidimensional scaling of the distance matrix and are shown for visualization only.

Each variant was run independently 20 times on N40W80. HV was computed using normalization bounds shared by all three variants within each run.

Table 5: N40W80 ablation experiment results (20 independent runs)

Algorithm	HV (mean ± SD)	Mean NNS	Mean runtime (s)
NSGA-II	0.7384 ± 0.0512	99.35	10.55
NSGA-II-RandLS	0.7461 ± 0.0409	102.45	11.43
RC-NSGA-II	$\mathbf{0.7619 \pm 0.0479}$	98.90	12.24

Table 5 shows that local search increases the mean HV from 0.7384 to 0.7461. Adding RC-guided resource allocation further raises the mean HV to 0.7619, corresponding to improvements of approximately 2.12% over NSGA-II-RandLS and 3.18% over NSGA-II. The complete method therefore combines local refinement with more effective allocation of search effort. Its mean runtime of 12.24 s reflects the additional cost of subspace matching and contribution updates.

Figure 4 shows that NSGA-II-RandLS is competitive early in the search, whereas RC-NSGA-II overtakes both ablation variants in the middle generations and retains the highest mean HV at termination. This trajectory is consistent with Table 5: local refinement provides an initial gain, and contribution-guided allocation improves how that effort is directed over time.

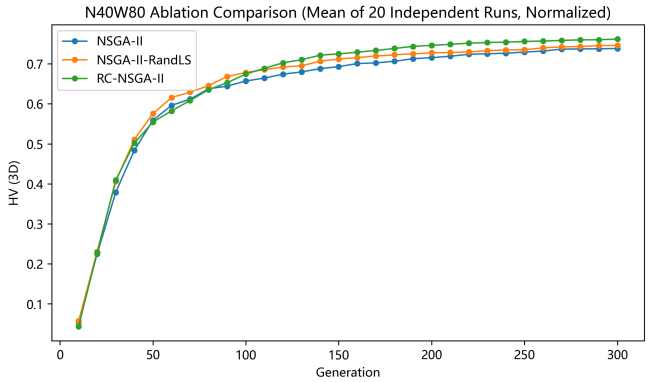


Figure 4: Comparison of HV convergence curves in ablation experiments (N40W80, average of 20 independent runs)

6.5 Parameter Sensitivity

The number of subspaces K controls the granularity of RC guidance. A small K produces coarse regions, whereas a large K leaves fewer individuals in each region and may reduce the reliability of contribution estimates. We evaluate $K \in \{3, 5, 7, 9, 11\}$ on N40W80 using 10 independent runs per setting. Because HV is normalized within this sensitivity experiment, its absolute values are not directly comparable with those in the main algorithm comparison.

Table 6: Normalized HV mean and standard deviation under different K values

K	HV mean	HV standard deviation	ranking
3	0.3015	0.0367	3
5	0.2955	0.0494	5
7	0.3175	0.0322	1
9	0.3053	0.0365	2
11	0.3053	0.0316	4

$K = 7$ yields the highest mean HV and a relatively small standard deviation. Performance varies only moderately across the tested values, suggesting that the method is not strongly dependent on a narrowly tuned K within this range.

6.6 AEW-AHP Preference Analysis

Table 7 compares the recommended solutions under the three preference scenarios, and Figure 3 shows their corresponding route structures.

Table 7: Three target values of recommended solutions under different preference scenarios

Scenario	Cost f_1 (yuan)	Completion time f_2 (min)	Dissatisfaction f_3	Drone flights
Cost priority	**1282**	327	361.6	4
Timeliness priority	3247	**234**	149.9	10
Satisfaction priority	3764	376	**20.0**	10

The three recommendations differ substantially in route structure and drone utilization. The cost-priority solution activates only four drone sorties and assigns most deliveries to the truck, producing the lowest cost ($f_1 = 1282$) but a longer completion time. The time- and satisfaction-priority solutions each use 10 sorties and fully exploit the three available drones. The former achieves the shortest makespan ($f_2 = 234$ min), whereas the latter reduces dissatisfaction to $f_3 = 20.0$. Because all three solutions lie on the same Pareto front, the differences reflect the explicit preference weights rather than separate optimization runs.

7 Conclusion and Outlook

This study addressed multi-objective routing for a single truck and multiple drones under flexible time windows. It combined RC-NSGA-II with an AEW-AHP preference module to connect Pareto-set generation with scenario-specific decision support. The principal findings are as follows:

1. *Algorithm effectiveness:* Across the seven-algorithm comparison, RC-NSGA-II ranks first in mean HV on N20W80, N40W80, and N80W80 and is essentially tied with NSGA-II on N60W80. The set-coverage results also show stronger directional dominance over the comparison algorithms.
2. *Mechanism competitiveness:* Compared with ODEA-ARA-adapted, RC-NSGA-II achieves a higher mean HV and a lower mean runtime on all four scales. The bidirectional coverage results likewise favor RC-NSGA-II under the unified experimental model.

3. *Component contributions:* Local search improves the mean HV of NSGA-II, and RC-guided allocation further increases it to 0.7619. The 20-run mean convergence curve confirms that the complete method maintains its advantage in the middle and late stages of the search.
4. *Preference decision:* AEW-AHP produces differentiated recommendations from the same Pareto solution set for different operating priorities. The decision process is transparent and interpretable.

Several limitations define directions for future work. First, Manhattan distance is only an approximation to travel on an urban road network; using OpenStreetMap or operational travel-time data would improve spatial realism. Second, the current fleet is homogeneous, with fixed drone payload and endurance. Extending the decoder to heterogeneous drones would broaden the range of practical applications. Third, demand is static. A rolling-horizon framework could accommodate cancellations, newly arriving orders, and other real-time changes.

Fourth, ODEA-ARA-adapted preserves the resource-allocation mechanism of the original method but not its delivery-and-pickup representation or seven specialized neighborhoods. The findings therefore support a mechanism-level comparison under a unified model and should not be interpreted as a reproduction study of the original algorithm. Fifth, the experiments do not report optimality gaps relative to an exact solver. Even N20W80 produces a large MILP with tightly coupled synchronization constraints, and related exact approaches require specialized pricing and acceleration strategies [8]. Establishing exact bounds for smaller compatible instances is an important next step.

Finally, the current decoder assumes a single truck. The RC module itself operates in objective space: K-means partitioning and the recursive update in Eq. (30) do not depend on the number of trucks. A multi-truck extension would therefore mainly require a new chromosome, decoder, and neighborhood structure. Its effectiveness, however, must be established through dedicated algorithm design and experiments.

Funding

This work was supported by the Tianjin Science and Technology Program (Grant No. 25YDPJC00550) and the Scientific Research Project of Higher Education Institutions in Hebei Province (Grant No. CXZX2026001).

References

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

[2] N. Agatz, P. Bouman, and M. Schmidt. Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981, 2018.

[3] W. C. Chiang, Y. Li, J. Shang, and T. L. Urban. Impact of drone delivery on sustainability and cost: Realizing the UPS vision. *Omega*, 84:58–73, 2019.

[4] N. Boysen, S. Schwerdfeger, and F. Weidinger. Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, 271(3):1085–1099, 2018.

[5] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà. On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, 86:597–621, 2018.

[6] Z. Wang and J.-B. Sheu. Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, 122:350–364, 2019.

[7] Qun Chen and Chao Li. A review of truck–drone collaborative transportation for urban last-mile logistics. *Journal of Changsha University of Science and Technology (Natural Science)*, 2025. Online First, 10 July 2025.

[8] Yunqiang Yin, Dongwei Li, Dujuan Wang, Joshua Ignatius, T. C. E. Cheng, and Sutong Wang. A branch-and-price-and-cut algorithm for the truck-based drone delivery routing problem with time windows. *European Journal of Operational Research*, 309(3):1125–1144, 2023.

[9] A. Ponza. Optimization of drone-assisted parcel delivery. *Master's Thesis, University of Padova*, 2016.

[10] S. Poikonen, X. Wang, and B. Golden. The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1):34–43, 2017.

[11] Xianlong Ge, Qiushuang Yin, Md. Abdul Moktadir, and Jingzheng Ren. Multi-objective optimization of truck-drone cooperative routing problem based on customer classification and fuzzy time windows. *Transportation Research Part E: Logistics and Transportation Review*, 203:104375, 2025.

[12] J. Xing, T. Guo, and L. C. Tong. Reliable truck-drone routing with dynamic synchronization: A high-dimensional network programming approach. *Transportation Research Part C: Emerging Technologies*, 165:104698, 2024.

[13] Qizhang Luo, Guohua Wu, Anupam Trivedi, Fangyu Hong, Ling Wang, and Dipti Srinivasan. Multi-objective optimization algorithm with adaptive resource allocation for truck-drone collaborative delivery and pick-up services. *IEEE Transactions on Intelligent Transportation Systems*, 24(9):9642–9657, 2023.

[14] Linjing Zhang and Yi Ding. Bi-objective emergency material delivery optimization based on truck–drone collaborative distribution. *Journal of Systems and Management*, 34(1):81–95, 2025.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[16] Y. Tian, L. Si, X. Zhang, R. Cheng, C. He, K. C. Tan, and Y. Jin. Evolutionary large-scale multi-objective optimization: A survey. *ACM Computing Surveys*, 55(4):1–34, 2023.

[17] S. Liu, Q. Lin, J. Li, and K. C. Tan. A survey on learnable evolutionary algorithms for scalable multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 27(6):1941–1961, 2023.

[18] F. Wang, H. Li, and H. Xiong. Truck–drone routing problem with stochastic demand. *European Journal of Operational Research*, 322(3):854–869, 2025.

[19] M. Deng, Y. Li, J. Ding, Y. Zhou, and L. Zhang. Stochastic and robust truck-and-drone routing problems with deadlines: A Benders decomposition approach. *Transportation Research Part E: Logistics and Transportation Review*, 190:103690, 2024.

[20] Y. Yin, Y. Yang, Y. Yu, D. Wang, and T. C. E. Cheng. Robust vehicle routing with drones under uncertain demands and truck travel times in humanitarian logistics. *Transportation Research Part B: Methodological*, 174:102781, 2023.

[21] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report TIK Report 103, ETH Zürich, 2001.

[22] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[23] Y. Tian, X. Li, H. Ma, X. Zhang, K. C. Tan, and Y. Jin. Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(4):1051–1064, 2023.

[24] T. L. Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(3):234–281, 1977.

[25] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002.

[26] Álvaro Fialho, Luis Da Costa, Marc Schoenauer, and Michèle Sebag. Analyzing bandit-based adaptive operator selection mechanisms. *Annals of Mathematics and Artificial Intelligence*, 60(1–4):25–64, 2010.

[27] Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys*, 45(3):35:1–35:33, 2013.

[28] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Shaker, Ithaca, NY, 1999.

[29] J. R. Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Massachusetts Institute of Technology, 1995.

[30] C. A. C. Coello and M. S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 1051–1056, 2002.

[31] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature*, pages 832–842, 2004.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- MultiObjectivePathOptimizationforTruckDroneCollaborativeDeliveryBasedonSubspaceContributionDrivenAdaptiveResourceAllocation.zip