WEP Warwick Evans Publishing

International Journal of Global Economics and Management

ISSN: 3005-9690 (Print), ISSN: 3005-8090 (Online) | Volume 7, Number 2, Year 2025
DOI: https://doi.org/10.62051/ijgem.v7n2.15
Journal homepage: https://ijgem.org

# Optimization of Semi-Open Multi-Depot Vehicle Routing and Delivery Scheduling

Wanqi Luo

College of Management, Shanghai University, Shanghai, China

## ABSTRACT

With the ongoing advancement of the "Internet Plus" initiative and the sharing economy, crowd-sourced delivery—a critical component of urban instant logistics—faces challenges such as high order heterogeneity and complex resource allocation. This study constructs a semi-open multi-depot scheduling model with dual objectives of minimizing delivery costs and maximizing customer satisfaction. The model systematically incorporates an order classification mechanism and a rider competency hierarchy, while integrating multiple real-world constraints including time windows, task load balancing, and route feasibility. To efficiently solve this NP-hard problem, we propose an improved genetic algorithm (IGA) featuring a dual-layer encoding structure. By introducing a coordinated optimization strategy for task assignment and route sequencing, the algorithm enhances solution stability and quality in complex scenarios. Numerical experiments based on real-world platform data demonstrate the superiority of the model and algorithm in scheduling efficiency, route optimization, and satisfaction improvement. This research provides both theoretical foundations and empirical evidence for intelligent scheduling in instant delivery platforms.

## KEYWORDS

Crowd-sourced Delivery; Route Optimization; IGA; Semi-open Scheduling

## 1. INTRODUCTION

Against the backdrop of rapid urban logistics system development, crowd-sourced delivery has emerged as a critical service modality to meet users' instant and diversified demands. Particularly in the post-pandemic era, the rise of "contactless economy" has further accelerated the proliferation and specialization of instant delivery services. As a specialized segment, crowd-sourced delivery—centered on "purchasing, delivering, fetching, and task completion" services—exhibits distinctive characteristics including small-batch tasks, high-frequency operations, and elastic resource allocation. However, traditional single-depot closed routing strategies have become increasingly inadequate in addressing the complex heterogeneous task structures and multi-resource coordination challenges in urban logistics scenarios.This study investigates a semi-open multi-depot vehicle routing problem with platform

## 2. LITERATURE REVIEW

The rapid development of "Internet + Logistics" and the sharing economy has drawn increasing academic attention to instant delivery services as a critical component of urban logistics systems. Scholars have conducted systematic research on multi-depot scheduling models, real-time order allocation mechanisms, and pickup-delivery routing optimization algorithms, establishing a substantial theoretical and methodological framework in this field.

## 2.1. Research on Multi-Depot Coordination and Semi-Open Route Optimization

Early research on multi-depot delivery problems primarily focused on closed routing constraints and independent depot scheduling. Wen et al. [1] and Zhang et al. [2] advanced this field by incorporating nonlinear energy consumption, load capacity, and road resistance factors, thereby establishing more realistic energy consumption models for electric vehicle routing. Regarding route structure optimization, Chen et al. [3] introduced a multi-depot resource sharing mechanism into routing optimization, achieving multi-objective optimization through integrated vehicle scheduling and energy consumption control.

Despite these significant advancements in multi-depot architecture and routing modeling, most existing models still rely on homogeneous order assumptions, failing to systematically incorporate critical real-world elements such as order prioritization mechanisms and courier capability heterogeneity in instant delivery scenarios.

## 2.2. Dynamic Scheduling and Order Matching Mechanisms in Instant Delivery Scenarios

In the field of instant delivery problems, the research focus has gradually shifted from static route optimization to dynamic order dispatching and platform capacity allocation strategies. Zhang et al. [4] designed a dynamic rider collaboration scheduling model in the context of crowdsourcing platforms, introducing order rolling and transfer mechanisms to enhance system responsiveness. Huang et al. [5] developed a spatiotemporal network optimization model to address split-order consolidation issues. Wang et al. [6] proposed a hierarchical recommendation strategy integrating prediction and rider preferences to achieve personalized order assignment. Zhen et al. [7] combine rolling scheduling with fuzzy time windows to construct an immediate demand-oriented dispatch model. Gu et al. [8] approached intelligent instant order allocation from the perspectives of column generation and ant colony optimization, respectively, emphasizing the importance of dynamic route adaptability and customer timeliness responsiveness.

## 2.3. Research Progress on PDPTW Routing Optimization and Algorithms

Although the above studies have effectively expanded the scheduling algorithms and business modeling for instant delivery scenarios, there are still obvious gaps in path network structure modeling, order level differentiation and hierarchical utilization of rider resources. Pickup and Delivery Problem with Time Window (PDPTW) is an important theoretical foundation for modeling the errand delivery problem. liu et al. [9] use a two-stage robust optimization model to deal with demand uncertainty and path volatility, and Xu et al. [10] combine Q-learning and transit point constraints to improve the path scheduling robustness. xue et al. [11] propose a similarity insertion algorithm for order trajectories.

In terms of algorithm design, Wu et al. [12] enhance the path perturbation reconstruction ability by improving ant colony optimization, and Pilati et al. [13] introduce Relocate with 2-opt* local search operator to improve the adaptability of PDPTW model to order changes. Muraretu et al. [14] simulate the negotiation behavior of delivery workers through multi-intelligence body model to strengthen the platform autonomous scheduling mechanism. Fan et al. [15] introduced a dynamic energy consumption strategy in a time-varying traffic network scenario, combining K-means clustering with an improved ant colony algorithm for multi-stage optimization.

Although PDPTW related research has been continuously developed at the model and algorithm level, most of them still focus on the traditional one-to-one task structure of “pick-up-and-deliver”, which fails to adapt to the complex characteristics of multi-centers, multiple order levels, and unpaired service structure in the instant delivery.

In summary, the current literature has formed a variety of theoretical results in the integration of multi-center scheduling mechanism, real-time order allocation strategy and path algorithm, but there are still three deficiencies: (1) most of the models lack the systematic modeling of heterogeneous order levels and delivery personnel ability grading; (2) the structure of the intelligent optimization algorithm has limited expressive power, and it lacks the synergistic mechanism of path and task allocation; (3) the multi-center semi-open path structure has been studied, but it is not consistent with the actual structure of the multi-center and semi-open path. has been studied, but it is insufficiently docked with actual instant delivery platforms and lacks a comprehensive integrated perspective

### 2.4. Contributions of This Paper

This paper focuses on the problems of strong order heterogeneity and complex resource scheduling in errand delivery, and proposes a unified modeling and solution framework that integrates multicenter synergy and path optimization. Firstly, a multicenter semi-open scheduling model is constructed, which integrally introduces constraints such as order level, rider capability, time window and path feasibility, breaking through the traditional closed path and homogeneous task assumptions. Secondly, the IGA with two-layer structure is designed to enhance the feasibility and global search capability of the algorithm through the co-optimization of task assignment and path ordering. Finally, multiple simulation experiments and sensitivity analysis are carried out based on actual data to verify the effectiveness of the model and algorithm in path optimization, satisfaction improvement and scheduling robustness.

## 3. PROBLEM DEFINITION AND DECOMPOSITION

### 3.1. Problem Definition

The study constructs a scheduling optimization model with high practical adaptability based on actual platform operation scenarios. The model is based on the multi-center semi-open path structure, and introduces key constraints such as rider level, order level, service time window, path cost and satisfaction penalty mechanism.

In order to ensure the solvability of the model, this paper makes the following assumptions in the context of actual orders: (1) the number of delivery personnel remains stable during the system operation, without considering personnel mobility; (2) the platform has multiple distribution centers, and the delivery personnel can pick up and return the equipment in the vicinity of the platform, and the paths do not need to be closed; (3) the same level of delivery personnel use standardized means of transportation, and the driving speed and energy consumption per unit are constant; (4) transportation interruption is not considered. situations such as traffic control or vehicle breakdown; (5) customer satisfaction is only affected by delivery time, ignoring other factors.

### 3.2. Mathematical Models

This section formulates an MILP model for an integrated optimization problem on assigning multiple types of orders to multiple grades of couriers and sequencing serving activities of nodes dedicated to these orders for each courier.

The relevant variables, sets and parameters are categorized and organized in this paper and are listed in detail in Table1 and Table 2.

**Table 1.** Notation for Decision Variables

| notation | instruction |
|---|---|
| $a_{wk}$ | 1 when order w is assigned to delivery agent k, 0 otherwise |
| $b_{kg}$ | 1 when distributor k's rank is g, 0 otherwise |
| $\varepsilon_{ik}$ | 1 when distributor k visits the node $i$, 0 otherwise |
| $\beta_{ii'k}$ | 1 when distributor k visits the node immediately after visiting the node, 0 otherwise |

**Table 2.** Notation for Model Sets and Parameters

| notation | instruction |
|---|---|
| W | The set of all orders indexed by w |
| K | The set of all distributors, indexed by k |
| G | The set of all distributor ranks, indexed by g |
| E | The set of all distribution center points, indexed by e |
| I | The set of all demand nodes in the distribution network, which are indexed by $i$ |
| $I_{\mathrm{w}}$ | The set of nodes contained in order W, indexed by $i_w$ |
| $I^0$ | The set of all pickup nodes |
| $\bar{I}$ | A collection of all delivery nodes |
| $I_w^0$ | The set of pickup nodes contained in order W |
| $\bar{I}_w$ | Order W contains a collection of delivery nodes |
| $n_g$ | Maximum number of orders that can be assigned to a delivery person of rank g |
| $D_{\mathrm{i}}$ | The set of loadings for all nodes |
| $J_i$ | The set of unloadings for all nodes |
| $h_i$ | The set of waiting times at node $i$ |
| $d_{ii'k}$ | Distributor k travel cost from node $i$ to node $i'$ |
| Q | Maximum load capacity of vehicles for distributors |
| $c_{g_i}$ | Penalty cost coefficient for node $i$ of rank g |
| $\lambda_{g_i}$ | Satisfaction coefficient of node $i$ with rank g |
| M | A sufficiently large positive number |
| $t_{ii'k}$ | Distributor k travel time from node i to node $i'$ |
| $u_{ik}$ | Load when distributor k leaves the node $i$ |
| $[e_i, l_i]$ | Optimal service time window for node $i$ |
| $[E_i, L_i]$ | Maximum time window that node $i$ is allowed to be served |

### 3.3. Mixed Integer Linear Programming Model

The study in this paper aims to minimize the total distribution cost, where the total cost consists of the vehicle's travel cost and the penalty cost for delayed delivery.

$$C_1 = \sum_{k\in K}\sum_{i\in I}\sum_{i'\in I} d_{ii'k}\beta_{ii'k} \tag{3-1}$$

$$C_2 = \sum_{i\in I}\sum_{k\in K} \varepsilon_{ik} c_{g_i}\left(l_i - \delta_{ik}\right)^+ \quad g \in G \tag{3-2}$$

$$S = \sum_{i \in I} \sum_{k \in K} \varepsilon_{ik} \lambda_{g_i} \cdot \min(1, P_i) \quad g \in G \tag{3-3}$$

Where,

$$P_i = \begin{cases} 1 \quad , \ \delta_{ik} \le E_i \\ \dfrac{\delta_{ik} - E_i}{e_i - E_i}, \ E_i \le \delta_{ik} < e_i \\ 0 \quad , e_i \le \delta_{ik} < l_i \\ \dfrac{L_i - \delta_{ik}}{L_i - l_i}, l_i \le \delta_{ik} < L_i \\ 1 \quad , \quad \delta_{ik} \ge L_i \end{cases} \tag{3-4}$$

$$\begin{aligned} Z = Minimize[ & \sum_{k \in K} \sum_{i \in I} \sum_{i' \in I} d_{ii'k} \beta_{ii'k} + \sum_{i \in I} \sum_{k \in K} \varepsilon_{ik} c_{g_i} \left( l_i - \delta_{ik} \right)^+ ] \cdot \phi\alpha \\ & + [\sum_{i \in I} \sum_{k \in K} \varepsilon_{ik} \left( 1 - \lambda_{g_i} \cdot \min(1, P_i) \right)] \cdot \gamma \quad g \in G \end{aligned} \tag{3-5}$$

Subject to

$$\sum_{k \in K} \alpha_{wk} = 1 \quad w \in W \tag{3-6}$$

$$\sum_{w \in W} b_{kg} \alpha_{wk} \le n_g \quad k \in K, g \in G \tag{3-7}$$

$$\sum_{k \in K} \alpha_{wk} \cdot k_g \ge w_g \quad w \in W, g \in G \tag{3-8}$$

$$u_{ik} \le Q \quad k \in K, i \in I \tag{3-9}$$

$$u_{ik} + D_{i'} - J_{i'} \le Q + M(1 - \beta_{ii'k}) \quad i \in I, i' \in I, k \in K \tag{3-10}$$

$$\sum_{i \in I^0} \beta_{ii'k} = \sum_{i \in \bar{I}} \beta_{i'ik} = \varepsilon_{i'k} \quad i' \in I, k \in K \tag{3-11}$$

$$\sum_{i \in I} \sum_{e \in E} \beta_{eik} = \sum_{i' \in I} \sum_{e' \in E} \beta_{i'e'k} \quad k \in K \tag{3-12}$$

$$\alpha_{wk} = \varepsilon_{ik} \quad i \in I_w, w \in W, k \in K \tag{3-13}$$

$$\delta_{i'k} \ge \delta_{ik} - M(2 - \varepsilon_{ik} - \varepsilon_{i'k}) \quad i \in I_w^0, i' \in \bar{I}_w, w \in W, k \in K \tag{3-14}$$

$$\delta_{ik} \le M \varepsilon_{ik} \quad i \in I, k \in K \tag{3-15}$$

$$\delta_{i'k} \ge \delta_{ik} + h_i + t_{ii'k} - M(1 - \beta_{ii'k}) \quad i \in I, i' \in I, k \in K \tag{3-16}$$

$$\alpha_{wk}, \beta_{ii'k}, \varepsilon_{ik} \in \{0,1\} \quad i \in I, i' \in I, i \ne i', w \in W, k \in K \tag{3-17}$$

$$\delta_{ik} \ge 0 \quad i \in I, k \in K \tag{3-18}$$

The objective function (3-1) represents the cost of vehicle travel; the objective function (3-2) represents the penalty cost of delayed delivery; and the objective function (3-3) represents the satisfaction level. Explanatory notes on the above constraints: Constraints (3-6) ensures that each order can be assigned; Constraints (3-7) indicates that the number of orders carried by each different rank of distributor cannot exceed the maximum number of orders of its corresponding rank; Constraints (3-8) requires that the rank of the distributor has to be higher than or equal to the rank of the order; Constraints (3-9) and (3-10) together constrain that the vehicle load of the distributor at any one node cannot be exceed the vehicle load of the distributor of that rank; Constraints (3-11) ensures that the routes of all distributors are consecutive; Constraints (3-12) means that a distributor does not have to return to the original distribution center after completing the assigned task; Constraints (3-13) implies that only a single distributor can access the nodes dedicated to the same order; and constraints (3-14) ensures that the nodes included in each order conform to the priority relationship; Constraints (3-15) indicates that the time variable is equal to 0 if courier k does not visit node i; Constraints (3-16) represents the time constraint for the successive visits of the distributor to these two nodes; Constraints (3-17) (3-18) define the domain of the decision variables.

# 4. SOLUTION STRATEGY

Aiming at the high-dimensional combinatorial complexity and strong constraints of the multi-center semi-open errand delivery scheduling problem, this paper proposes an IGA for solving the problem. Compared with the traditional genetic algorithm (GA), the algorithm makes key improvements in the coding structure, constraint handling mechanism and genetic operation to improve its adaptability and convergence performance in complex distribution scenarios.

## 4.1. Double-layer Encoding Strategy

In order to realize the decoupling optimization of task allocation and path ordering, this paper designs a double-layer chromosome structure of “task allocation layer + task ordering layer”: the task allocation layer indicates which rider executes each delivery task, encoded as a sequence of integers with the length of the number of orders, with each gene position representing an order, and the value field is the feasible rider set number. Each gene position represents an order, and the value field is the set number of feasible riders; the task sequencing layer encodes the specific order in which tasks are served, and is used to determine the execution path of the orders received by each rider. The ordering layer adopts a chain structure to maintain the order of task services, which facilitates the subsequent insertion and swapping operations.

## 4.2. Decoding Strategy

The decoding process consists of five stages: (1)Assign each distributor a set of orders based on the first-level chromosome's task allocation code; (2)Decompose each distributor's orders into task points and sequence them according to the second-level chromosome's task ordering code to form the initial task sequence; (3)Use a bidirectional linked list to store each distributor’s task sequence, facilitating order adjustment, legality repair, and local search; (4)For each order, ensure the pickup node precedes all delivery nodes in the task chain by repositioning the pickup node as needed; (5)Finally, assign the nearest distribution center as the return point based on each distributor’s last delivery location to complete the route.

## 4.3. Adaptive Selection Operation

In the GA selection operation, this paper adopts the adaptive bidding race selection mechanism and elite reverse learning strategy approach. Individuals are ranked by fitness value, and the top 10% elite

individual set is identified and EOBL is used, while the remaining non-elite individuals are generated by adaptive bidding tournament method.

In this research, we introduce the inverse coefficient $\Psi$ and the dynamic boundary mechanism to improve the classical inverse solution computation, set the value of elite individuals in the $i$ dimension in the t generation population as $x_i^{elite}$, and design the dynamic boundary of its corresponding dimension based on the overall boundary and the dynamic contraction window as $[a_i^t, b_i^t]$, which is computed by the following formula:

$$a_i^t = a_i + r(t)\cdot(b_i - a_i)\ ,\ b_i^t = b_i - r(t)\cdot(b_i - a_i) \tag{4-1}$$

$$r(t) = \eta\cdot(1-\frac{t}{T_{\max}}) \tag{4-2}$$

Where $[a_i, b_i]$ is the global initial fixed boundary of the $i$ dimension, $T_{\max}$ is the maximum number of iterations of GA, t represents the number of current iteration rounds r(t) is the boundary scaling factor under the current iteration, which will be gradually reduced as the iteration progresses. $\eta \in [0,0.5]$ is used to control the boundary compression, and $\eta = 0.5$ indicates that the initial shrinkage ratio is the largest, and the initial search width is half of the global boundary.

### 4.4. Overall Improved Genetic Algorithm

Based on the above components, the framework of our proposed IGA is described as follows.

Step1: Set the core parameters of the GA, including population size, crossover probability, mutation probability and maximum number of generations.

Step2: Randomly generate a set of feasible individuals (chromosomes), each composed of two encoding layers: a task allocation layer and a task sequence layer. The initialization process ensures that all individuals satisfy courier-level compatibility constraints, maximum task capacity limitations, and basic feasibility rules such as pickup-before-delivery ordering.

Step3: Each individual is decoded by constructing delivery routes using a doubly linked list structure, enforcing pickup-before-delivery constraints and calculating route feasibility. The overall fitness is computed based on two objectives: minimizing total delivery cost and maximizing customer satisfaction.

Step4: A combined mechanism of elite reverse isochronous learning selection and competitive race selection is used to select parent individuals for the next generation.

Step5: For the task assignment layer, the partial mapping crossover operator is applied to Partially-matched crossover (PMX); For the task sequence layer apply a Order Crossover(OX) operator to preserve task order feasibility and enhance route quality.

Step6: On the task allocation layer, introduce small-scale perturbations by reassigning orders among eligible couriers; On the task sequence layer, apply a Feasible Insertion Mutation operator that relocates task points while maintaining the pickup-before-delivery constraint.

Step7: For any individual violating the constraint that a courier's level must be greater than or equal to the order's level, perform repair operations. If a courier exceeds the allowed number of orders, redistribute the excess tasks to other eligible couriers.

Step8: Record the best individual with the highest fitness across generations. Output includes the optimal delivery route, order–courier assignment scheme, and objective performance indicators such as total cost and customer satisfaction.

Its simulation framework diagram is shown in Figure 1.

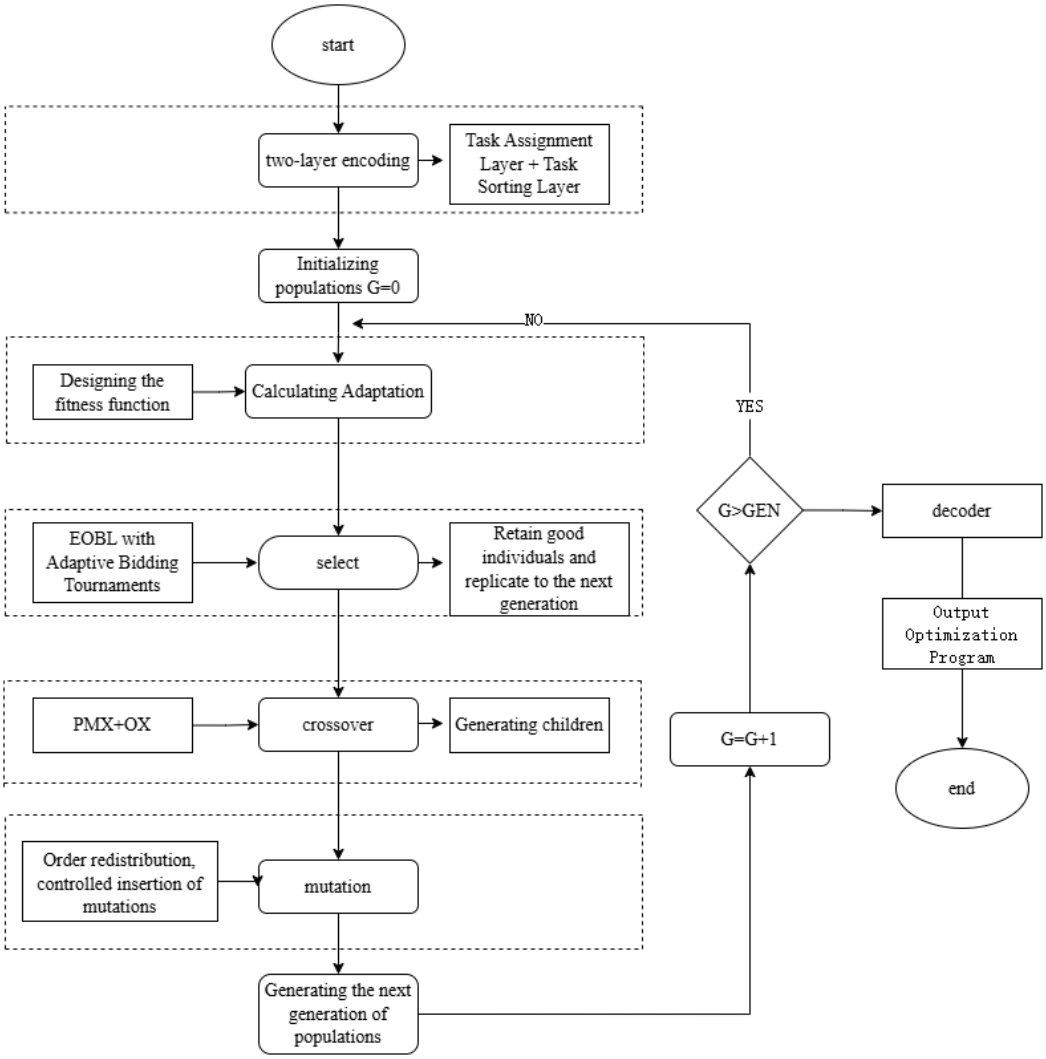



**Figure 1.** IGA Framework

## 5. NUMERICAL EXPERIMENTS

Several numerical experiments were conducted to evaluate the efficiency of the proposed model and the efficiency of the proposed algorithm. The experimental platform is a laptop equipped with Intel i5-5600H processor and 16GB RAM, and the operating system is Windows 10 64-bit Professional, the algorithm implementation and solution are done in MATLAB R2023a environment, and the time limit for all test instances is 3,600s.

### 5.1. Experimental Settings

In this research, the typical business scenario of "M company" in a city center area is selected as a case study, in which three distribution centers and 50 potential customer nodes (i.e., office buildings, residential areas, schools, etc.) have been identified in the network, whose locations are shown in Fig. 2, and the delivery workers are divided into three levels according to their service experience and ability. Based on the distribution network of these nodes, order information values are randomly

generated during the same period of time, and Table 3 shows the parameter settings of the experiment. We generated five groups of test algorithms with different scales, and each group contains five randomly generated algorithms. The key parameters involved in the algorithm are shown in Table 4.

**Table 3.** Benchmark Instance Configuration

| Example ID | Order Number | Number of couriers | Number of nodes |
|---|---|---|---|
| ISG1 | 5 | 3 | [10, 25] |
| ISG2 | 7 | 4 | [14, 35] |
| ISG3 | 20 | 12 | [40, 100] |
| ISG4 | 30 | 20 | [60, 120] |
| ISG5 | 50 | 30 | [100, 200] |

**Table 4.** System Parameters for Routing Optimization Model

| notation | define | unit | value |
|---|---|---|---|
| $n_g$ | Maximum number of orders for different levels of delivery personnel | order | 4, 5, 6 |
| v | travel speed | km/h | 30 |
| $c_1$ | Vehicle transportation rates | Yuan/kg·km | 0.002 |
| $c_{g_i}$ | Penalty coefficients for late arrival of different grades of delivery personnel | — | 1, 1.2, 1.5 |
| $\lambda_{g_i}$ | Satisfaction coefficients for different levels of orders | — | 1, 1.2, 1.5 |

## 5.2. Small-Scale Test Cases Verify the Accuracy of the Algorithm in Solving the Model

We first conducted experiments on small-scale instances to compare the efficiency of the exact solver and particle swarm optimization (PSO) algorithms, and the difference evolution algorithm (DE) with IGA. All are used to solve mixed integer programming models. The results in Table 5 show that the IGA and the gurobi solver obtain the same optimal solution for the small-scale instances, which indicates that our algorithm can solve the model to optimality. On the other hand, the proposed exact algorithm is very efficient in solving this model, with a running time about three times faster than the exact solution.

## 5.3. Medium-Scale Test Cases Verify the Superiority of the Algorithm in Solving the Model

In this study, 10 sets of medium-scale test cases were randomly generated as experimental subjects. For each case, comparative experiments were conducted using the exact solution method, DE, PSO, and IGA. Each algorithm was independently executed 10 times for the same case, with the optimal total cost obtained in each run systematically recorded. The detailed computational results of each algorithm across different test cases are presented in Tables 6.

For instances ISG3 3-1 and ISG3 3-2, the IGA successfully obtained optimal solutions consistent with the exact method, achieving a 0% GAP, while requiring significantly shorter computation time. In contrast, for larger-scale instances (3-3 to 5-5), the increased problem complexity made it impossible to derive optimal solutions using exact methods. However, by comparing the results of the three metaheuristic algorithms, it was observed that the IGA consistently demonstrated superior solution quality, with an average single-run time ranging between 43 and 77 seconds.

**Table 5.** Computational Results of 10 Small-scale Test Instances

| Instance | | | Exact Solution | | IGA | | | |
|---|---|---|---|---|---|---|---|---|
| Group | ID | Nodes | Optimum | Time (s) | Optimal Solution | Time (s) | GAP | Time ratio |
| ISG1 | 1-1 | 13 | 67.631 | 35.55 | 67.631 | 11.51 | 0% | 32.38% |
| | 1-2 | 13 | 62.616 | 63.63 | 62.616 | 11.10 | 0% | 17.44% |
| | 1-3 | 15 | 76.367 | 40.17 | 76.367 | 12.32 | 0% | 30.67% |
| | 1-4 | 15 | 84.122 | 39.74 | 84.122 | 10.36 | 0% | 26.07% |
| | 1-5 | 16 | 88.934 | 68.37 | 88.934 | 15.13 | 0% | 22.13% |
| ISG2 | 2-1 | 18 | 92.099 | 85.65 | 92.099 | 24.31 | 0% | 28.38% |
| | 2-2 | 20 | 103.438 | 159.93 | 103.438 | 13.15 | 0% | 8.22% |
| | 2-3 | 20 | 95.980 | 212.24 | 95.980 | 22.57 | 0% | 10.63% |
| | 2-4 | 21 | 107.832 | 179.08 | 107.832 | 29.64 | 0% | 16.55% |
| | 2-5 | 21 | 101.760 | 238.72 | 101.76 | 30.30 | 0% | 12.69% |

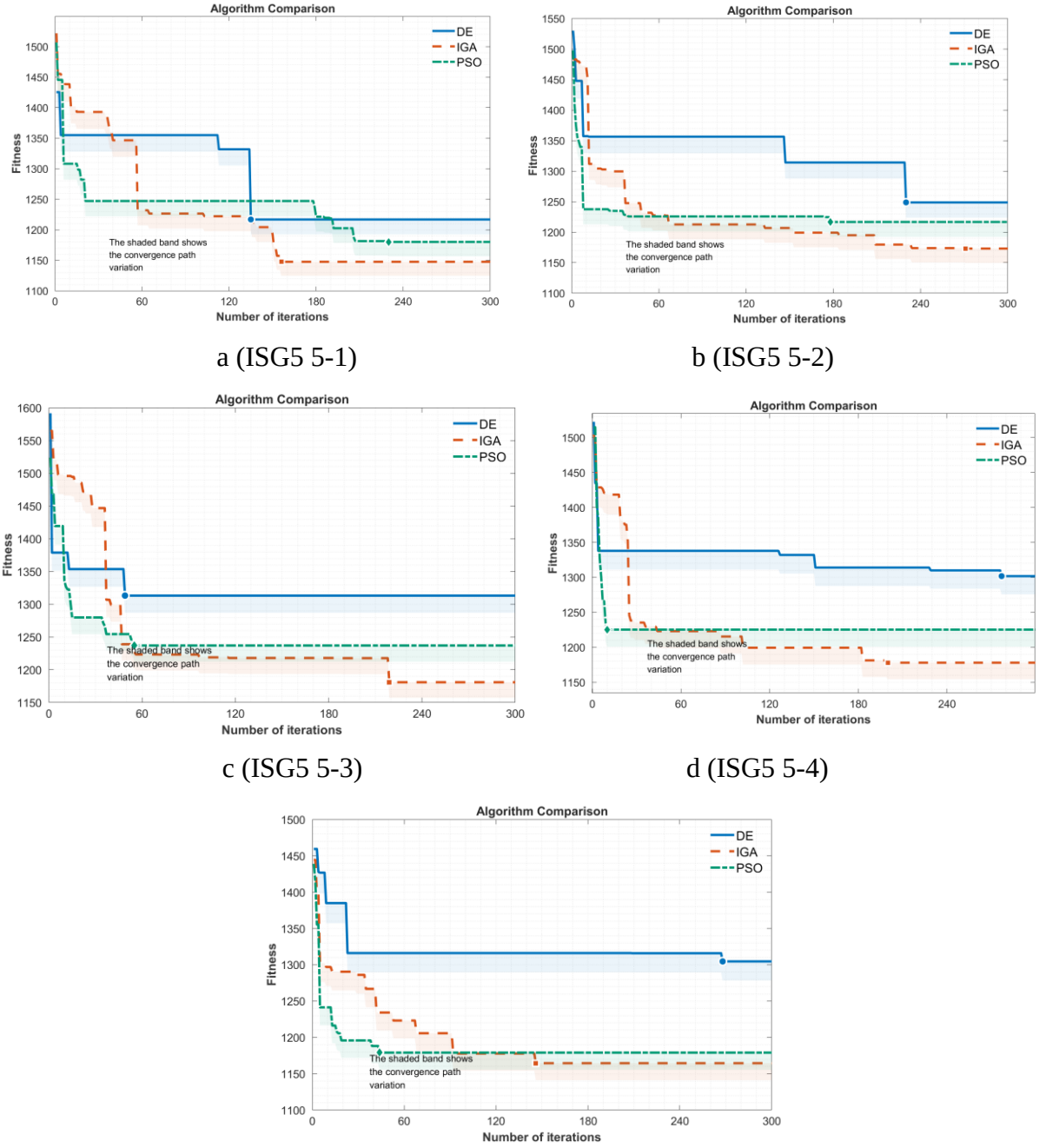
**Table 6.** Comparative Algorithm Performance on Medium-scale Instances

| Instance | | | Exact Solution | | DE | | | PSO | | | IGA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Group | ID | Nodes | $F_1$ | $t_1$ | $F_2$ | $t_2$ | GAP | $F_3$ | $t_3$ | GAP | $F_4$ | $t_4$ | GAP |
| ISG3 | 3-1 | 47 | 254.8 | 2451 | 259.9 | 56 | 2% | 257.2 | 65 | 1% | 254.8 | 55 | 0% |
| | 3-2 | 54 | 275.2 | 3346 | 278.7 | 80 | 1% | 288.5 | 68 | 5% | 275.2 | 43 | 0% |
| | 3-3 | 59 | - | - | 347.2 | 88 | - | 343.7 | 113 | - | 337.6 | 72 | - |
| | 3-4 | 59 | - | - | 338.4 | 76 | - | 348.8 | 71 | - | 328.8 | 66 | - |
| | 3-5 | 62 | - | - | 364.7 | 71 | - | 367.5 | 92 | - | 356.2 | 75 | - |
| ISG4 | 4-1 | 72 | - | - | 399.9 | 90 | - | 388.6 | 86 | - | 383.3 | 73 | - |
| | 4-2 | 74 | - | - | 426.5 | 74 | - | 441.8 | 78 | - | 419.3 | 61 | - |
| | 4-3 | 82 | - | - | 408.5 | 79 | - | 397.9 | 77 | - | 386.6 | 69 | - |
| | 4-4 | 84 | - | - | 451.9 | 96 | - | 441.8 | 102 | - | 441.4 | 77 | - |
| | 5-5 | 90 | - | - | 450.5 | 88 | - | 454.0 | 118 | - | 448.9 | 76 | - |

## 5.4. Large-Scale Test Cases Are Used to Verify The Convergence of the Algorithm in Solving the Model

To further evaluate the solution quality and convergence capability of the proposed IGA for large-scale path optimization problems, this study selects multiple large-scale test cases and conducts a comparative analysis of the convergence processes of DE, PSO, and IGA under identical parameter configurations. Figure 2 illustrates the fitness value evolution curves during the iterative process across five large-scale test cases.

As evident from Figure 2, the IGA demonstrates superior global optimization capability, achieving the optimal final fitness value. In all test cases, the IGA maintains continuous improvement in solution quality during middle and late iterations, ultimately obtaining the lowest fitness values. Compared with DE, the IGA not only escapes local optima traps but also exhibits stronger sustained optimization capability. When contrasted with PSO, the IGA shows significant advantages in both convergence speed and solution stability.




a (ISG5 5-1) b (ISG5 5-2)

c (ISG5 5-3) d (ISG5 5-4)

e (ISG5 5-5)

**Figure 2.** Fitness Evolution Curves for Large-scale Instance

## 5.5. Sensitivity Analysis of Order Tier Structure

The specific experimental settings are as follows: The total number of orders is fixed at 50, while the total number and hierarchical structure of delivery personnel remain unchanged—specifically, 10 Level I couriers, 10 Level II couriers, and 10 Level III couriers. The order acceptance rule for couriers is that they can only accept orders with a level less than or equal to their own.

To comprehensively reflect different business scenarios, this study selects five representative order-level distribution combinations. Each experimental group is independently executed five times, with the scheduling results corresponding to the optimal total cost selected for analysis. The 3D surface plot illustrating the impact of order-level structures on total costs is shown in Figure 3.

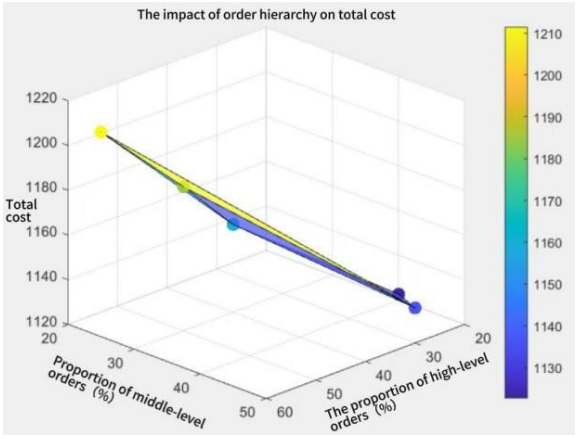



**Figure 3.** Impact of Order Priority Structure on Total Cost

The surface plot exhibits an upward trend from the lower-left corner toward the back-right direction, indicating that the system experiences the heaviest operational burden when high-tier orders increase while mid-tier orders decrease proportionally. Conversely, when the order distribution is more balanced—with high-, mid-, and low-tier orders proportionally closer—the total cost remains at a relatively lower level.

## 5.6. Sensitivity Analysis of Courier Tier Structure

In this experiment, the order structure was fixed at 50 orders with a proportional distribution of 40% high-tier, 30% mid-tier, and 30% low-tier orders. By systematically adjusting the courier tier composition, we investigated the impact of different team structures on the dispatching system's responsiveness. The experimental results are visualized in a 3D surface plot (Figure 4), which intuitively demonstrates the comprehensive effects of varying tier combinations on system performance

The constructed 3D surface in the figure exhibits a downward-sloping plane structure, indicating that the total cost is highly sensitive to the courier tier configuration. Particularly during the initial increase of Level III couriers, the cost decreases at a faster rate. Once the number of Level III couriers reaches a certain threshold, the system operation stabilizes, and the rate of cost reduction diminishes, suggesting that the system scheduling has approached an optimal state.

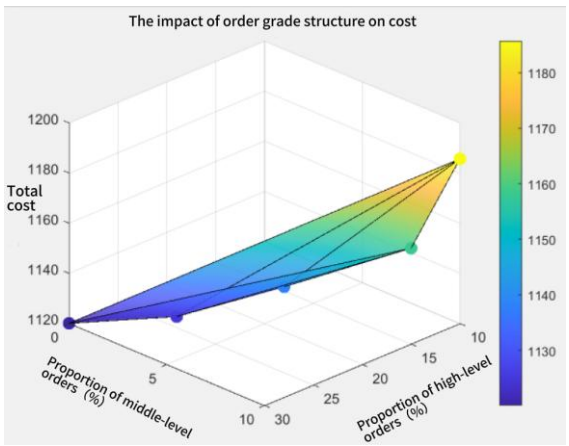



**Figure 4.** Impact of Courier Tier Structure on Total Operational Cost

## 6. CONCLUSIONS

We have developed a Mixed-Integer Programming (MIP) model tailored for on-demand delivery scenarios. The model classifies five types of orders into three tiers and assigns orders of different tiers to couriers of corresponding tiers. The objective is to minimize the total cost of servicing all orders within the same period while maximizing customer satisfaction.

This study contributes to the literature in several aspects:

From a modeling perspective, the proposed MIP model is comprehensive and incorporates diverse decision-making factors. It introduces an integrated modeling approach that combines an "order-courier" tiering mechanism with route optimization to better capture the heterogeneity of real-world delivery scenarios. Specifically, customers and orders are categorized into high-, medium-, and low-tier groups based on their time sensitivity, payment willingness, and product characteristics. Differentiated time-window constraints and penalty weights are then applied to construct a customer satisfaction evaluation function, significantly enhancing the model's adaptability to complex delivery scenarios.

From an algorithmic perspective, to address the challenges of "complex solution space structure and stringent feasibility constraints" in on-demand delivery route optimization, we designed a dual-layer chromosome structure consisting of a "task assignment layer" and a "task-point sequencing layer." The algorithm incorporates a tier-compliance repair strategy and a courier workload control mechanism to improve its ability to handle complex scheduling constraints. Furthermore, by integrating sequential crossover, controlled insertion mutation, bidirectional linked-list path adjustment, and elite opposition-based learning strategies, the algorithm enhances global search efficiency and convergence stability while maintaining population diversity.

## CONFLICTS OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## ACKNOWLEDGEMENTS

The authors thank the editor and reviewers for their constructive suggestions to improve this paper.

## REFERENCES

[1] Wen M, Laporte G. A dynamic programming approach for multi-depot electric vehicle routing problem with nonlinear energy consumption [J]. Computers & Operations Research, 2018, 93: 63–77.

[2] Zhang Y, Wang Q. Urban delivery route optimization for hybrid-fuel fleets under time windows [J]. Transportation Research Part D: Transport and Environment, 2020, 82: 102312.

[3] Chen J, Wang Y, Sun L. Joint optimization of vehicle routing and charging in collaborative urban logistics networks [J]. Applied Soft Computing, 2021, 108: 107443.

[4] Zhang R, Zhao L, Zhang X. Dynamic collaboration and reallocation in crowdshipping platforms [J]. Transportation Research Part E, 2021, 150: 102329.

[5] Huang L, Liu J. Spatio-temporal network model for consolidation of split e-grocery orders [J]. Computers & Industrial Engineering, 2020, 139: 106188.

[6] Wang X, Zhang M, Liu P. A crowdsourcing-based intelligent order recommendation system for on-demand delivery platforms [J]. Expert Systems with Applications, 2022, 194: 116541.

[7] Zhen L, Zhang X, Wang Y. Optimization of heterogeneous courier scheduling with order types in instant delivery [J]. Transportation Research Part C, 2020, 115: 102617.

[8] Gu M, Zhang C. Order clustering and assignment for online-to-offline logistics under real-time demand [J]. Computers & Operations Research, 2019, 107: 40–53.

[9] Liu J, Zhu W, Wang H. Robust vehicle routing with uncertain demand and time windows using two-stage optimization [J]. Transportation Research Part B, 2021, 147: 1–17.

[10] Xu Y, Li B, Song J. Dynamic pickup and delivery problem with transfer and reinforcement learning-based operator adjustment [J]. Expert Systems with Applications, 2022, 196: 116582.

[11] Xue M, Zhou Y, Zhang H. Online order insertion strategy based on trajectory similarity for real-time delivery routing [J]. Knowledge-Based Systems, 2023, 263: 110186.

[12] Wu T, Lin C, Tseng Y. Enhanced ant colony system for vehicle routing with pickup and delivery and time windows [J]. Soft Computing, 2020, 24(7): 5069–5081.

[13] Pilati F, Ambrosino D, Fikar C. Multi-objective local search for pickup and delivery with time windows [J]. Computers & Industrial Engineering, 2021, 160: 107602.

[14] Muraretu A, Crainic T G. Multi-agent cooperative routing with negotiation for urban distribution [J]. European Journal of Operational Research, 2020, 285(3): 1008–1023.

[15] Fan F, Zhang D. Time-dependent vehicle routing with battery constraints in urban networks [J]. Applied Energy, 2022, 314: 118981.