AIMS Mathematics, 11(1): 2613–2644.
DOI: 10.3934/math.2026106
Received: 02 November 2025
Revised: 07 January 2026
Accepted: 16 January 2026
Published: 27 January 2026

https://www.aimspress.com/journal/Math

Research article

Optimization of takeaway tableware recycling route considering order insertion

Chenghan He[1,*], Dexin Huang[2,*] and Chengyin Wang[2]

[1] Reading Academy, Nanjing University of Information Science and Technology, Nanjing, Jiangsu 210044, China
[2] School of Economics and Management, Southeast University, Nanjing, Jiangsu 211189, China

* **Correspondence:** Email: 202383100008@nuist.edu.cn, 230250057@seu.edu.cn; Tel: +8615139915710.

Abstract: China's booming food-delivery industry produces massive disposable-tableware waste, demanding efficient and low-carbon reverse logistics. Here, we studied a dynamic tableware collection routing problem with real-time order insertion: a recycling center serves preset recycling points, while new door-to-door requests appear during route execution. The objective is to minimize total cost, including vehicle dispatch fixed cost, distance-based depreciation, cleaning cost (including incremental cleaning for inserted orders), waiting and lateness penalties under soft time windows, and fuel plus carbon-emission costs. Routes must satisfy depot start/end, single-service requirements, vehicle capacity limits, feasible service-time propagation, and a minimum satisfaction threshold derived from the soft time-window function. To solve this NP-hard problem, we designed an improved genetic algorithm with time-window-based grouped initialization, natural-number encoding with depot separators, OX crossover, two-point mutation, and a destruction-repair local search using farthest insertion for reinsertion. Experiments indicated faster and more stable convergence than a basic GA. In an order-insertion case, inserting new orders into en-route tours significantly outperforms dispatching an additional vehicle (total cost about 75.7% higher). The proposed method offers implementable decision support for platforms and municipalities to run time-sensitive, low-carbon tableware recovery.

Keywords: route optimization; advanced genetic algorithm; dual carbon goal; environmental sustainability; take-out tableware recycling
Mathematics Subject Classification: 90B06, 90C27

The main parameters in the model are shown below:

Notation	Hidden meaning
V	Set of all nodes, including depot and pickup nodes: $V = \{0\} \cup F \cup D$
0	Depot (recycling center) node index
F	Set of fixed-point pickup nodes (planned tasks in the baseline tour)
D	Set of inserted/dynamic pickup nodes (newly arrived orders to be inserted)
A	Set of feasible directed arcs: $\mathrm{A} = \{(i,j) \mid i,j \in V, i \neq j\}$
K	Set of available vehicles, $K = \{0,\ 1,\ 2,\ \ldots,\ K\}$
E	Recover the set of travel paths of vehicles between two points, $E = \{(i,j) \mid i,j \in V, i \neq j\}$
F_k	Fixed cost of use of vehicle k
$C_{\mathrm{ij}}^{\mathrm{k}}$	Cost per unit distance traveled by the vehicle from the starting point i to the target point j
$\mathrm{r_i}$	Number of containers to be recycled at recycling point i
p_{c1}	Cleaning cost per unit number of containers collected from fixed collection points
p_{c2}	Cleaning cost per unit number of containers recovered from insertion order collection points
β_1	Base cost of cleaning meal boxes recovered from insertion order collection points
β_2	Variable cost of cleaning meal boxes recovered from insertion order collection points
tc_i	Recycling operation time at recycling point i
$tc_i^{'}$	Recycling point i Time from being recycled to being cleaned for dishes that need to be recycled
Q	Maximum load capacity of the vehicle
d_{ij}	Euclidean distance between origin i and destination j
t_i	Point in time to reach recovery point i
α_1	Cost of waiting for vehicles to arrive early at recovery point i
α_2	Penalty cost of delayed vehicle arrival at recovery point i

Continued on next page

Notation	Hidden meaning
f_t	Total fuel consumed by vehicles
f_p	Fuel consumption per unit distance travelled by vehicles
f_s	Fuel consumed per unit of time during vehicle recovery activities
p_f	Price per unit of fuel volume
c_0	Price per unit of carbon emissions
ω	Carbon emission factor
$[e_i, l_i]$	The time window in which the recycling point expects recycling to take place
$[E_i, L_i]$	The time window in which the recycling point is acceptable for recycling
x_{ij}^k	Whether vehicle k is traveling on the path (i, j), if yes, then 1, otherwise 0
y_i^k	Whether the recycling activity is performed by vehicle k at recycling point i, if yes, then 1, otherwise 0

1. Introduction

Over the past decade, the popularization of online food-delivery platforms and evolving consumption patterns have driven rapid sectoral growth [1]. Industry reports indicate that the "post-90s" cohort constitutes the largest consumer segment, exceeding 50% of demand, while the purchasing power of "post-70s" and "post-80s" consumers remain strong; orders priced above 30 yuan occur at markedly higher rates among these latter groups than among post-90s. The consumer base has broadened from youth to middle-aged adults. Meituan data show that, in 2019, first-tier, new first-tier, and second-tier cities formed the core markets, accounting for 64.7% of national orders. However, order volumes and sales growth in lower-tier cities have outpaced those in higher-tier cities, and for many white-collar workers, delivery has become a necessity. During the three-year pandemic, restaurants that once treated delivery as a niche strategy shifted decisively to online channels, further accelerating market expansion [2]. Against this backdrop, China's food-delivery market reached 934 billion yuan in 2021 and was expected to surpass one trillion yuan in 2022.

However, the rapid expansion of the takeaway market has brought about serious environmental issues. In 2020, China's takeaway industry generated 17 billion takeaway orders, each containing an average of 3.44 lunch boxes, 70% of which were various plastic lunch boxes. The report "Mapping the Environmental Impacts and Policy Effectiveness of China's Takeaway Food Industry" analyzes more than 35 million takeaway orders, maps the characteristics of China's takeaway industry, the use of packaging, and its environmental impacts, and evaluates the policy pressures faced by the industry as well as the effectiveness of plastic packaging control programs. The report shows that takeaways in the economically developed southeastern coastal provinces, the Yellow River Basin and the

Yangtze River Basin, generate more plastic packaging waste. Because of the proximity of these areas to bodies of water, the likelihood of water contamination from leaking plastic is also higher. In terms of the timing of orders, takeaway waste is produced at a much faster rate than its disposal. The report shows that takeaway orders are generated around the clock. The amount of takeaway waste being recycled is much lower than the amount of takeaway waste being produced. Most of the takeaway packaging waste is generated during peak order times at noon and in the evening, accounting for 60% of all orders. Due to the rapid generation of takeaway packaging waste during these peak times, a large amount of waste accumulates quickly and is not removed promptly. As a result, a significant portion of recyclable waste that is not disposed of in time ends up being treated as non-recyclable waste and is either sent to landfills or incinerated, which has a serious impact on the environment.

Currently, many foreign catering enterprises or catering platforms are exploring reusable tableware to pack takeaway food and ensuring the recycling of tableware and lunch boxes through specific ways and mechanisms [3]. For example, the foreign DabbaDrop platform requires customers to purchase stainless steel containers, which will be recycled by takeaway riders when they order takeaway food next time. The Sharepack platform is also committed to promoting reusable tableware so that customers can return the tableware from their last order when they order takeaway food the next time. Alternatively, they can return the tableware to the nearby cooperative catering outlets. In China, Haidilao hot pot's takeaways are delivered by a delivery person who agrees with the customer on a pick-up time. On the other hand, Yikou Liangshi uses recyclable and reusable ceramic bowls for takeaway delivery, which the store staff recycles.

Regarding recycling methods, domestic takeaway platforms have carried out both fixed-point and door-to-door recycling in some areas. In some places, the recycling bodies of the two recycling modes are the same, but in the actual recycling process, there are problems of higher cost and lower efficiency. Therefore, this paper proposes a takeaway recycling path optimization model considering dynamic order insertion to maximize the efficiency of recycling vehicles and solve the insertion of orders while meeting fixed-point recycling to optimize cost savings.

This paper takes the takeaway tableware recycling vehicle scheduling problem as an entry point. It proposes to achieve cost reduction, carbon emission reduction, and satisfaction enhancement through the rational arrangement of vehicle paths. The following is a brief overview of the results related to this paper from the perspectives of reverse logistics, vehicle path problems, and dynamic path planning.

1.1. Reverse logistics-related research

Research on reverse logistics for municipal recyclable waste, including takeaway tableware, mainly focuses on strategic studies, such as recycling models. Linderhof et al. [4] simulated the impact of the deposit refund model on the recycling rate and pointed out that the deposit refund model is particularly effective when the recycling rate is relatively low. Xu et al. [5] proposed a courier siting model, a door-to-door delivery model, and a courier locker model to recycle courier packages. Currently, most companies tend to outsource all or part of their recycling processes [6] and increase the flexibility of logistics reorganization by cooperating with third-party reverse logistics providers [7]. Therefore, how to select a third-party reverse logistics provider is a crucial issue. Zarbakhshnia et al. [8] introduced a hybrid multi-attribute decision-making approach that integrates the advantages of fuzzy analytic hierarchical processes and grey multi-objective optimization to help

enterprises select a third-party reverse logistics provider. Similarly, Chen et al. [9] proposed a multi-perspective, multi-attribute decision-making framework based on semantic analysis, which provides an effective and efficient way for enterprises to select the best third-party reverse logistics provider for systematic decision support. In addition, the government and individuals should be the leading players in packaging recycling activities in cooperation with enterprises. Agovino et al. [10] showed that the joint efforts of the government and the public can improve waste treatment efficiency. Therefore, it is increasingly important to explore the operational management issues related to packaging recycling from both government and consumer perspectives. As a regulator and policymaker, the government is essential to encouraging recycling activities [11]. Since government management has been ineffective in environmental protection in recent years, it was suggested that the government should play an active role in recycling management while working closely with other stakeholders [12]. Heydari et al. [13] suggested that the government can improve and coordinate reverse logistics by providing tax exemptions and subsidies. Yang et al. [14] compared the market-driven, government-driven, and cooperative-driven models and found that choosing a cooperative-driven recycling model can get the highest benefit, and the subsidies are conducive to increasing recycling efforts. However, relevant studies have pointed out that government-led recycling management activities have the disadvantages of having a single participant and low efficiency [15], which is unsuitable for modern urban packaging management. Therefore, some scholars began to explore the multi-actor shared governance system, such as Berthomé and Thomes [16], who proposed that the community should participate in that system. The government is still the center of governance but only plays a resource supply and supervision role; the community should be the primary bearer.

1.2. Research related to vehicle path problems

The vehicle routing problem, referred to as the VRP problem, originated from the traveling salesman problem (TSP), in which the customer base is much larger than that of the TSP problem. As such, the customer demand cannot be satisfied by a single vehicle or a single distribution path. And multiple delivery routes and delivery vehicles must be used to solve the problem. Compared with the TSP problem, the VRP problem is more complex and more challenging to solve, and the difficulty increases dramatically with the constraints of various realistic conditions. However, because of this, the VRP problem is closer to reality. Song [17] established a model with the objective of maximizing customer time satisfaction, taking into account customer time window constraints and release time constraints. Wang et al. [18] improved the cuckoo algorithm for the vehicle path problem, using examples to verify that the improved algorithm is more efficient than the original algorithm. Fan et al. [19] set the optimization objectives as minimizing delivery cost, the number of vehicles dispatched, delivery distance, and total order delay time. The multi-objective model was established according to the theory of credibility testing. Huang et al. [20] designed an adaptive genetic grey wolf optimization algorithm with capacity constraints for the vehicle path problem, which adopts a routing and grouping strategy. Majidi et al. [21] improved the large neighborhood search algorithm, with the optimization objective of minimizing the delivery vehicles' carbon emissions and fuel consumption.

1.3. Research related to dynamic path planning

Zhang et al. [22] improved a genetic algorithm based on a local search strategy to solve a dynamic path planning problem. Zhang et al. [23] designed a two-stage solution strategy for the dynamic pickup and delivery vehicle path problem with multiple vehicles and combined 2-OPT quantum evolution to improve the solution efficiency. Shi and Zhang [24] employed an improved iterative local search algorithm to solve a dynamic pickup and delivery vehicle routing problem with randomly distributed demand points. Zhang et al. [25] improved the iterative local search algorithm to optimize the delivery cost and time satisfaction for the delivery of takeout, taking into account the customer's priority. Zhang et al. [26] investigated the mixed loading and unloading problem that occurs during transportation under stochastic and dynamic conditions, extending the objective function from single to multi-objective. The number of vehicles was extended from single to multi-vehicle, and according to the different situations, corresponding strategies were proposed for solving the problem. The advantages and disadvantages of each solution strategy were analyzed. Lei et al. [27] improved the adaptive large-neighborhood search algorithm and considered the random distribution of the customer demand points in the model to solve the dynamic customer satisfaction problem. Demand points are randomly distributed to solve the dynamic pickup and delivery vehicle path problem.

At this stage, most of the literature on takeaway recycling focuses on strategic issues, such as the choice of recycling model, and less on practical matters, such as optimizing takeaway recycling paths. Various recycling models also need to incorporate the unique characteristics of takeaway tableware recycling. On the other hand, most of the literature on recycling path optimization mainly focuses on single-model recycling path optimization and does not consider the possibility of completing the insertion order in the process of fixed-point recycling. In summary, this paper combines the existing literature and proposes a takeaway recycling path optimization model that considers dynamic order insertion to maximize the efficiency of recycling vehicle usage, solve the insertion order while satisfying fixed-point recycling, and maximize cost savings.

The rest of the paper is organized as follows: In Section 2, we describe the problems to be solved, introduce the improved methods of genetic algorithms, and build the basic algorithmic model. In Section 3, we implement an empirical case study to validate the effectiveness of the proposed improved genetic algorithm to solve the problem of takeout order insertion. In Section 4, model performance is validated by comparison of costs, and corresponding limitations of the model are proposed. Section 5 presents the main findings and key theoretical implications and management insights, as well as recommendations for future research.

2. Materials and methods

2.1. Description of the problem

Take the "Green Mountain Program" initiated by Meituan as an example; at this stage, the main body of takeaway garbage recycling is still a garbage transfer station. Two main recycling modes are fixed-point recovery and online ordering for door-to-door recycling. A comparison of the two recycling modes is shown in Table 1.

Table 1. Comparison of the two recycling methods.

	Fixed-point recovery	Order online for door-to-door recycling
Recycling locations	Higher and fixed number	Smaller and more dispersed
Recycled quantity	More	5 kg or more to order online
Recycling time	Lower urgency	Higher urgency

In this paper, several recycling vehicles at a recycling center are studied.

A recycling center is also responsible for the fixed recycling in the surrounding area. At the same time, online orders from the surrounding area will also be sent to the recycling center in real-time, and the recycling center dispatcher will change the recycling plan in real time to meet the fixed-point recycling. The recycling center scheduler will change the recycling schedule in real-time and insert the orders while satisfying the fixed points. The diagram with before and after order insertion is shown in Figure 1.

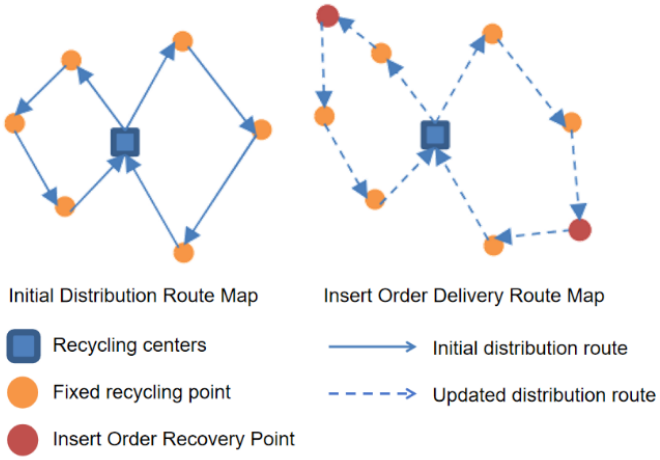

Figure 1. Schematic diagram of path update before and after order insertion.

2.2. Modeling

2.2.1. Model assumptions

This study develops its model based on the following integrated set of assumptions, aligned with standard practices in vehicle routing literature while incorporating the specifics of our dynamic recycling context. We assume the recycling quantity and time window for each fixed collection point are known a priori, and the depot is equipped with a sufficient homogeneous fleet. Upon receiving an online order, customers specify a desired collection time window, starting no less than one hour from the order placement; the system then prioritizes inserting these new requests into the routes of vehicles already in operation. Each vehicle has a maximum load capacity Q, and the cumulative load on any route must never exceed this limit, a constraint that must also be satisfied during dynamic insertion. The fuel consumption and carbon emissions on an arc (i, j) depend on the distance traveled, the vehicle's average speed, and its current load, with congestion and weather effects

captured through their influence on speed and modified energy coefficients. Travel times on network arcs are time-dependent, modeled as a step function of the departure time to reflect peak and off-peak periods. All collected tableware is fully recyclable without damage. For modeling clarity, we assume a uniform grease level for tableware from newly inserted orders (representing immediate post-use state) and a separate uniform level for tableware at fixed points. Finally, to handle operational uncertainty, external factors like weather are represented by a finite set of scenarios $s \in S$, where each scenario defines specific impacts on vehicle speed and energy consumption—an approach consistent with prior studies examining environmental disruptions such as rainfall.

2.2.2. Description of model parameters

Let V be the set of all nodes including the depot, fixed-point requests, and inserted requests. Indices $\mathrm{i,j} \in \mathrm{V}$ denote nodes, and $\mathrm{k} \in \mathrm{K}$ denotes vehicles, where K is the set of available vehicles and $K = |K|$ is the fleet size. Decision variable $x_{ijk} \in \{0,1\}$ equals 1 if vehicle k travels from node i to node j, and 0 otherwise. t_{ik} denotes the service start time of vehicle k at node i. The desired time window is $[e_i, l_i]$, while the acceptable time window is $[E_i, L_i]$, which is used to compute waiting and penalty terms.

2.2.3. Cost function analysis

(1) Fixed costs

The recycling center generates fixed costs whenever it dispatches recycling vehicles to carry out recycling activities. Fixed costs include purchasing or leasing recycling vehicles, personnel wages, and annual inspection fees for operating vehicles, among others. Usually, fixed costs can be taken as fixed values. Assuming that the recycling center dispatches a total of K_{used} vehicles for recycling in the whole recycling process, the fixed cost can be expressed as Eq (1):

$$C_1 = \sum_{k=1}^{K} \sum_{i=1}^{m} x_{0i}^{k} f_k \tag{1}$$

(2) Vehicle wear and tear costs

Vehicle wear and tear costs refer to the costs incurred due to wear and tear during vehicle operation. This part of the cost includes the depreciation cost of the vehicle with the increased distance travelled, the maintenance cost for each fixed distance travelled, and other costs caused by violations, accidents, and other unavoidable factors. The cost of wear and tear is proportional to the distance travelled and can be expressed as Eq (2):

$$C_1 = \sum_{k=1}^{K} \sum_{i,j=1}^{m} c_{ij}^{k} d_{ij} x_{ij}^{k} \tag{2}$$

(3) Cleaning costs

Recovered tableware needs to be cleaned, whether it can continue to be used as takeaway tableware or be processed into other products. The cleaning cost will increase with the side length of the recovery time, as the takeaway tableware in the fixed recycling point has been set aside for a long

time, so it is considered that the unit cleaning cost is a fixed value. The tableware that needs to be recovered for new orders is the tableware that has just been used, and the cleaning cost will vary significantly with time. The cleaning cost can be expressed as Eqs (3)–(5):

$$C_{31} = p_{c1}\sum_{i=1}^{n} r_i \tag{3}$$

$$C_{32} = p_{c2}\sum_{i=n+1}^{m} r_i = (\beta_1 + \beta_2 \mathrm{tc}_i^{'})\sum_{i=1}^{n} r_i \tag{4}$$

$$C_3 = p_{c1}\sum_{i=1}^{n} r_i + (\beta_1 + \beta_2 tc_i^{'})\sum_{i=1}^{n} r_i \tag{5}$$

(4) Time penalty costs

Suppose the recycling vehicle does not recycle within the customer's desired time window. In that case, the customer's satisfaction is reduced, affecting the motivation for later participation in tableware recycling, and a specific penalty cost will be incurred. Similarly, suppose the recycling vehicle arrives at the recycling location early. In that case, it will need to wait until the time when recycling is acceptable to the customer, which will also incur a specific waiting cost. The time penalty cost can be expressed as Eq (6):

$$C_4 = \sum_{i=1}^{m}(\alpha_1 \max\{e_i - t_i, 0\} + \alpha_2 \max\{t_i - l_i, 0\}) \tag{6}$$

(5) Fuel and carbon costs

Recovery vehicles consume fuel during recovery activities, and the amount consumed is related to the route length of the recovery activity. In addition, due to the "dual carbon target", carbon emissions are included in the cost considerations in this paper. Only one gas, CO_2, is considered in this paper. This paper uses the following formula to calculate the carbon generated by the fuel consumed in the recycling process: carbon emission = fuel consumption emission factor. Then, the cost of fuel and carbon emissions can be expressed as Eqs (7)–(9):

$$C_{51} = p_f \sum_{k=1}^{K}\sum_{i,j=1}^{m}(f_0 + f_1 L_i^k) d_{ij} x_{ij}^k \tag{7}$$

$$C_{52} = \omega c_0 \sum_{k=1}^{K}\sum_{i,j=1}^{m} t_{ij}(\tau, s) x_{ij}^k \tag{8}$$

$$C_5 = C_{51} + C_{52} \tag{9}$$

where $f_p(L_i^k) = f_0 + f_1 L_i^k$, and the greater the load, the higher the fuel consumption per unit mileage. Also, $t_{ij}(\tau, s) = K_{ij}(\tau)(1 + \delta_s)t_{ij}^0$, t_{ij}^0 is the benchmark travel time (sunny/unobstructed), $K_{ij}(\tau)$ is the congestion coefficient (peak > 1), and δ_s is weather scenario coefficient (rainy days > 0).

(6) Customer satisfaction

Since customer satisfaction is one of the most critical constraints, this paper adopts a hybrid time window to carry out the measure of customer satisfaction for timeliness. The functional relationship between timeliness satisfaction and time can be expressed as Eq (10):

$$S_i = \begin{cases} \dfrac{t_i - E_i}{e_i - E_i}, E_i \le t_i < e_i \\ 1, \quad e_i \le t_i \le l_i \\ \dfrac{L_i - t_i}{L_i - l_i}, l_i \le t_i < L_i \\ 0, \quad other \end{cases} \tag{10}$$

2.2.4. Cost function analysis

In summary, the model of this paper is shown as follows:

$$\min Z = C_1 + C_2 + C_3 + C_4 + C_5 \tag{11}$$

s.t.

$$\frac{1}{n}\sum_{i=1}^{m} S_i \ge S \tag{12}$$

$$\sum_{k=1}^{K}\sum_{i=1}^{m} x_{0i}^k \le K, \forall k \in K, i \in \{1,2,3,...,m\} \tag{13}$$

$$\sum_{k=1}^{K}\sum_{i=1}^{m} y_i^k \le 1, \forall k \in K \tag{14}$$

$$\sum_{i=1}^{m} x_{0i}^k = \sum_{i=1}^{m} x_{i0}^k \le 1, \forall k \in K \tag{15}$$

$$\sum_{i=1}^{m} y_i^k q_i \le Q, \forall k \in K \tag{16}$$

$$t_j = \sum_{k=1}^{K}\sum_{i=1}^{m} x_{ij}^k \left[\max(t_i, E_i) + ts_i + \frac{d_{ij}}{v} \right], j \in \{1,2,...,m\} \tag{17}$$

$$t_{sj} = \sum_{k=1}^{K} \sum_{i=1}^{m} x_{ij}^{k} \max(ts_i + tc_i + \frac{d_{ij}}{v}, E_j), j \in \{1,2,...,m\} \tag{18}$$

$$E_i \le ts_i \le L_i, i \in \{1,2,...,m\} \tag{19}$$

$$\sum_{k=1}^{K} y_i^k = 1, i \in \{1,2,...,m\} \tag{20}$$

where Eq (11) is an objective function that indicates that the total cost of the entire recycling activity is minimized; Eq (12) indicates the customer's satisfaction constraints concerning the timeliness; Eq (13) indicates that there are enough recycling vehicles at the recycling center to perform the recycling activity; Eq (14) indicates that each customer is served; Eq (15) indicates that each recycling vehicle departs from the recycling center and returns to the recycling center at the end; Eq (16) indicates that each recycling vehicle does not exceed the load capacity; Eq. (17) indicates the time the vehicle arrives at the recycling site; Eq (18) indicates the time the recycling vehicle starts its recycling activities at the recycling site; Eq (19) indicates that all recycling activities take place within a time window that is acceptable to the customer; and Eq (20) indicates that each recycling site is visited only once.

2.3. Algorithm design

Compared with a standard genetic algorithm, our method introduces two problem-driven enhancements to handle dynamic order insertion in recycling operations. First, we incorporate a sorted insertion (SI) procedure during decoding/repair to insert newly arrived pickup requests into existing vehicle routes by ranking candidate insertion positions according to incremental cost while enforcing feasibility. This step reduces the search space and prevents infeasible offspring from dominating the population. Second, we apply a local search (LS) intensification to each offspring after crossover/mutation using fast neighborhood moves to eliminate detours and improve route quality. Together, SI improves feasibility and responsiveness to new orders, while LS accelerates convergence and improves solution quality. After reviewing the relevant literature on vehicle routing problems both domestically and internationally, it is evident that different algorithms can be employed to address these problems, depending on the specific objective functions, decision variables, and function constraints. Seven heuristic algorithms are commonly used to solve the vehicle routing problem with time windows, and among these, the genetic algorithm and the ant colony algorithm are the most frequently utilized. Next, we will compare the advantages and disadvantages of some commonly used heuristic algorithms.

Both the genetic algorithm and ant colony algorithm are more suitable for the problem proposed in this paper. As the robustness of the genetic algorithm is higher, this paper chooses it to solve the model, because traditional genetic convergence is slower, and local search ability needs to be improved. This paper uses the time window information to group the data. It introduces the local search operation during the generation of the initial solution to make the algorithm converge faster and find a better solution.

The main body of the algorithm consists of two parts: “Initial Recovery Path Planning” and

"Insertion Order Path Planning". The algorithm's schematic diagram and the genetic algorithm's flowchart is shown in Figures 2 and 3.

(1) Chromosome coding

The text is encoded using natural number coding for the chromosomes. Take the example of a recycling center with four recycling carts and 8 fixed recycling locations.

If there are 2 codes 1-2-9-3-5-4-4-10-6-7-11-8-12 and 1-2-3-9-4-5-6-10-7-8-11-12, the information in the two chromosomes is shown in Table 2. The corresponding vehicle assignments and recycling routes for these two chromosome examples are compared in Table 3.

The natural numbers in the chromosome greater than 8 represent the recycling centers. The length of the chromosome is n+K-1 when the recycling point where the recycling activity is to be performed is n, and the maximum number of vehicles available is K.

(2) Initial solution generation and population initialization

A superior initial solution can significantly accelerate the convergence rate. This paper uses the following method to generate a more favorable initial solution:

Step 1: Arrange the recovery points according to the order of the time window and divide them into m segments, denoted as far as possible to satisfy $n = qm$.

Step 2: Randomly select one recovery point and arrange it.

Step 3: Repeat step 2 until all recovery points are aligned, generating a customer sequence.

Step 4: Traverse the sequence of customers sequentially with k = 1 and i = 1; place the sequence of customers in the kth path and determine if the load limit of the recovered vehicle is exceeded. If not, i = i + 1 and repeat.

Step 5: When the load constraint is not satisfied, store the kth path so that k = k + 1, and repeat step 4.

The initial solution is generated by traversing all the recovery points, and each sub-path formed by the initial solution is directly represented by a natural number greater than n as the recovery center to link the sub-paths into the initial solution. Repeat the above method to generate an initial population of 50.

(3) Adaptation function

The larger the degree of adaptation, the better the result. The objective function of this paper is to find the minimum value, so its inverse is used as the adaptation degree function.

(4) Selection

The selection operation is performed using a tournament strategy, where two days of chromosomes are randomly selected from the parent population to compare fitness sizes, and the larger ones go into the next generation. This selection method only compares the relative fitness size, which, to some extent, circumvents the situation in which convergence is too early, leading to the emergence of locally optimal solutions.

(5) Crossover

In this paper, crossover is performed using the method of order crossover (OX). The order crossover creates new offspring by swapping gene segments between parents. This enhances genetic algorithms' flexibility and convergence speed. In the paternal generation, chromosomes labeled P1 and P2 are chosen, with loci a and b selected randomly. A segment from P1 between a and b is removed to form C1, while a corresponding segment from P2 is inserted into C1, creating offspring C1. This process is similarly applied to create offspring C2. This is illustrated in Figure 4 and detailed in the following steps:

1) Randomly select two paternal chromosomes P1 and P2, randomly select a and b.
2) Different gene loci in P2 are sequentially inserted before and after the a and b segments in P1 to generate zygotic chromosome C1.
3) Generate another child C2 in the same way.

(6) Variation

In this paper, we use a two-point mutation method for mutation operation, i.e., two gene points on the chromosome are randomly selected for exchange operation.

(7) Local search

Genetic algorithms (GAs) provide strong global exploration, but the population tends to become homogeneous in the late stage, causing slow improvement and oscillation around local optima. To enhance late-stage intensification and solution quality, we embed a local search operator into the GA framework to refine promising solutions while maintaining feasibility, thereby further reducing the total cost. Our problem involves soft time-window penalties (customer satisfaction) and real-time order insertion, where small neighborhood moves (e.g., simple swaps or 2-opt) may be insufficient to escape local structures. The destruction-repair scheme "removes a small subset of customers (destroy) and reinserts them (repair)" to partially reshape route structures with a controllable perturbation. During the repair phase, feasibility checks (time windows and, if considered, capacity constraints) can be explicitly enforced, which improves the effectiveness of local improvements under complex constraints. In routing problems with time-window penalties, some customers are "hard-to-insert" due to remote locations or tight service windows; inserting them late often leads to large penalties or infeasibility. Farthest insertion prioritizes customers with the largest distance (or insertion difficulty) relative to the current partial routes, effectively placing the most challenging nodes first. This strategy reduces the risk of infeasible reinsertion and mitigates penalty accumulation, resulting in higher-quality reconstructed routes and improved algorithm stability.

To further enhance the ability of the algorithm to find the optimal solution, this algorithm adds the step of local search. The specific operation is as follows: first, eliminate a randomly selected gene point using the destruction operator to carry out the operation. Then, compute the correlation (distance) between the remaining and removed gene points. Third, remove the gene points with higher correlation individually. Finally, we will reintroduce the gene points into the chromosome using the farthest insertion method, which entails inserting the gene point with the highest minimum insertion cost first.

(8) Termination conditions

This paper's algorithm stops the operation when a set number of iterations is reached.

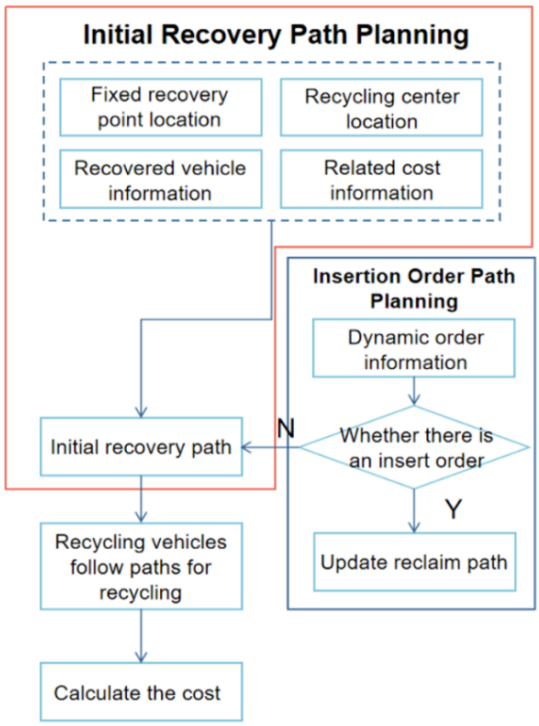

Figure 2. Schematic of the insert order algorithm.

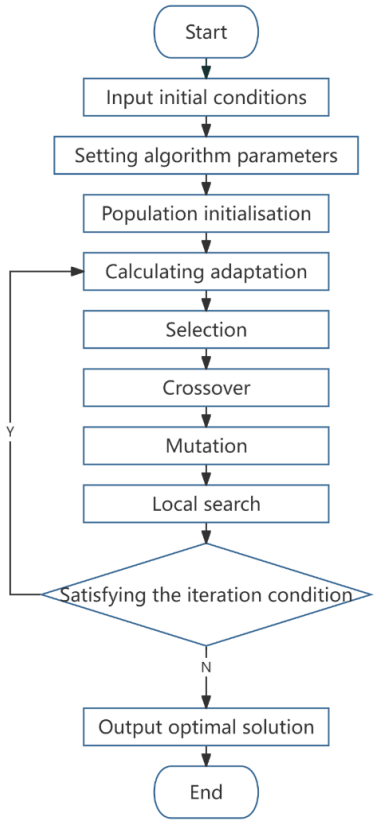

Figure 3. Flowchart of the genetic algorithm.

Table 2. Algorithm suitability comparison.

Algorithm	Advantage	Disadvantage	Applicable question
Genetic algorithm	Strong global search ability; strong adaptability; high robustness	Slow convergence; limited accuracy; parameter adjustment is complicated	Complex combinatorial optimization problem, NP-hard problem, function optimization problem, engineering design optimization, and so on.
Ant colony optimization	Positive feedback mechanism; distributed computing; adaptable to dynamic environments	Easy to fall into local optimality; slow convergence; the algorithm parameters are complex	Route optimization problems, such as traveling salesman problem (TSP), vehicle routing problem (VRP), network routing optimization, etc.
Simulated annealing	Simple and easy to implement; jumping out of local optimal; global optimal	Slow convergence speed; sensitive to initial value; parameter setting is complex	Global optimization problems, especially complex optimization problems with multiple extreme values, such as combination optimization, process parameter optimization, etc.

Table 3. Comparison of two chromosomes.

Chromosomes	1-2-9-3-5-4-10-6-7-11-8-12	1-2-3-9-4-5-6-10-7-8-11-12
Total vehicles	4	4
Used vehicles	4	3
Recovery path	0-1-2-0	0-1-2-3-0
(0 for return)	0-3-5-4-0	0-4-5-6-0
(collection centers)	0-6-7-0	0-7-8-0
	0-8-0	

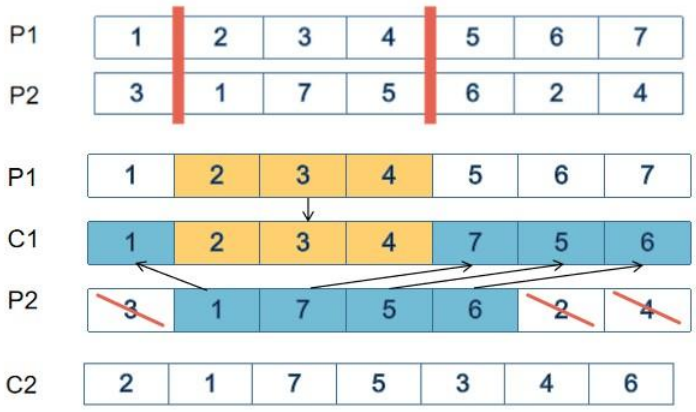

Figure 4. Schematic of order crossover.

3. Results

3.1. Initial recovery path planning

We calibrate the vehicle fixed utilization cost F as the per-shift (or per-day) fixed expense of deploying one recycling vehicle, including driver labor, vehicle depreciation/lease, insurance/registration, and routine maintenance. Specifically, $F = C_{labor} + C_{depr} + C_{ins} + C_{maint} + (+C_{overhead})$. The adopted value is obtained from local market quotations and/or operational accounting records, and we further test F in a sensitivity range of $\pm 20\%$ to verify robustness. The unit carbon price c_0 is referenced from the prevailing carbon trading price in the selected market/period. Since carbon prices fluctuate, we adopt the average (or end-of-period average) price over the study window and convert it to the unit used in our model (e.g., Yuan/kg) through standard unit conversion. A sensitivity analysis on c_0 is also conducted to ensure the conclusions are not driven by a single price point.

This study confirms the proposed model and algorithm through a series of experiments., and we provide information about the values of the experimental parameters. Due to the confidentiality of real-world business data and for the purpose of methodological validation, the numerical case study employs a set of simulated data generated based on classic VRP benchmarks and realistic assumptions, detailed in Table 4, including initial conditions, boundary conditions, and physical parameters, to better build the model. We have also recorded the recovery point information in Table 5, which captures a system state snapshot at a specific time during the numerical simulation. This information enables subsequent analysis and comparison. Moreover, we list the specific settings of the algorithm parameters in Table 6 to demonstrate the algorithm's performance and optimization effects. These parameters cover the algorithm's convergence conditions, the number of iterative steps, step size, and key optimization parameters. By providing these details, we aim to ensure a transparent and reliable experimental framework for numerical simulation and algorithm research.

Table 4. Parameter assignment table.

Parameters	Numerical value
Maximum number of vehicles to be recovered	20 vehicles
Vehicle fixed utilization cost F_k	100 Yuan (USD14.04)/per vehicle
Unit cleaning cost of tableware recovered from fixed collection points p_{c1}	0.5 Yuan (USD0.07)/kg
Cost of basic cleaning of dishes recovered from order insertion recycling points β_1	0.1 Yuan (USD0.014)/kg
Cost of basic cleaning of dishes recovered from order insertion recycling points β_2	0.002 Yuan (USD0.00028)/kg ·min
Maximum load capacity Q	100 kg
Waiting cost α_1	0.2 Yuan (USD0.028)/min
Late arrival cost α_2	1 Yuan (USD0.14)/min
Fuel consumption per unit distance traveled f_p	0.1 L/km
Price per unit of fuel volume p_f	7.06 Yuan (USD0.99)/L
Price per unit of carbon emissions c_0	1 Yuan (USD0.14)/kg

Continued on next page

Parameters	Numerical value
Carbon emission factor ω	2.63 kg/L
Minimum desired customer satisfaction S	85%
Distance correction coefficient	1.3
Recovery vehicle travel speed v_0	30 km/h

Table 5. Basic information on fixed recycling points.

Num	x	y	$[e_i, l_i]$	$[E_i, L_i]$	r_i	t_{ci}
0	30	40	7:00–21:00	7:00–21:00	0	0
1	41	49	17:00–17:30	16:00–18:30	30	20
2	35	17	16:30–17:00	15:30–18:00	10	5
3	55	45	9:00–9:30	8:00–10:30	30	20
4	55	20	16:00–16:30	15:00–17:30	10	5
5	15	30	7:30–8:00	7:00–9:00	20	10
6	25	30	15:00–15:30	14:00–16:30	10	5
7	20	50	9:40–10:10	8:40–11:10	30	20
8	10	43	10:40–11:10	9:40–12:10	10	5
9	55	60	14:40–15:20	13:40–16:20	10	5
10	30	60	12:00–12:30	11:00–13:30	20	10
11	20	65	13:30–14:00	12:30–15:00	20	10
12	50	35	15:10–15:40	14:10–16:40	20	10
13	30	25	9:00–9:30	8:00–10:30	15	5
14	15	10	14:10–14:40	13:10–15:40	30	20
15	30	5	13:10–13:40	12:10–14:40	25	15
16	10	20	11:00–11:30	10:00–12:30	20	10
17	5	30	14:10–15:10	13:10–16:10	25	15
18	35	35	14:30–15:00	13:30–16:00	30	20
19	15	60	17:00–17:30	16:00–18:30	25	15
20	45	65	16:30–17:00	15:30–18:00	20	10

Table 6. The initial recycling route.

Recycling vehicle number	Recycling routes
1	0-6-11-3-5-2-2-11-9-6-9-5-0
2	0-11-6-1-11-3-15-6-1-11-8-0
3	0-3-4-9-3-3-1-7-7-7-7-4-0
4	0-8-1-1-3-9-7-6-10-3-5-11-0

Moreover, to facilitate accurate calculation and analysis, the mathematical tool of Euclidean distance is used to measure and calculate the distance between two points. Although the Euclidean distance is simple and easy to use in theory, in practical application, considering the complexity and uncertainty of the road network, other measurement methods need to be adopted or modified to improve the accuracy of the model. According to the historical data of this region, the actual distance

is generally 1.3 times the Euclidean distance, that is, the correction coefficient $\rho = 1.3$, so the corrected distance is calculated by Eq (22). Specifically, the Euclidean distance is calculated as shown in Eq (21):

$$d_{ij} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{21}$$

$$d_{ij}^{'} = \rho d_{ij} \tag{22}$$

The provided text outlines a pathway scheme for an initial recovery effort, detailed in Figure 5, offering a clear view of its structure and flow. Additionally, it includes an economic analysis through Tables 6 and 7, listing the recovery routes and their costs—covering transportation, processing, and other related expenses—which aids in optimizing the recovery process for cost efficiency.

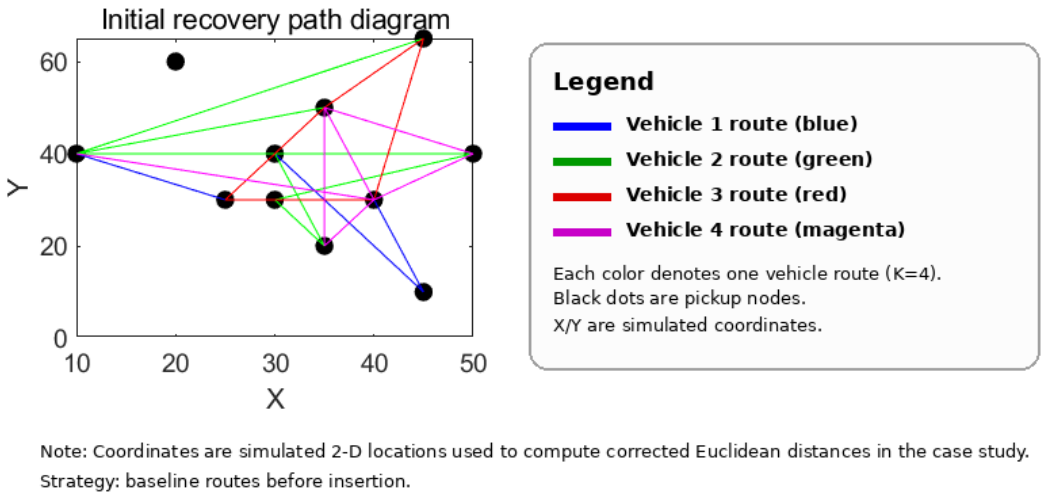

Figure 5. The initial recovery path.

Table 7. Cost table under the initial recovery path planning.

Cost classification	Amount/yuan (USD)
total cost	742.64 (USD104.22)
fixed costs	400 (USD56.14)
vehicle wear and tear costs	102.14 (USD14.33)
cleaning costs	67.22 (USD9.43)
time penalty cost	42.77 (USD6.00)
fuel and carbon costs	130.51 (USD18.32)

3.2. Inserting order routing

At 11:00, five new individuals are added to the recycling order, and the specific information is shown in Table 8. After inserting the order, the route for sending new vehicles from the recycling

center to perform the recycling task can be calculated, as shown in Table 9 and Figure 6; at the same time, it is possible to calculate a route where the original vehicle is still deployed directly from the midway to perform the recovery task, as shown in Table 10 and Figure 7.

In Figure 7, if the new vehicle is recycled separately according to the insertion sequence, the optimized path diagram is shown when compared to the direct deployment of the car in the initial path. In this way, we can observe the optimized path map, which shows the efficiency and cost-effectiveness of sending new vehicles separately in the recovery mission. This display of the control group-optimized pathway map helps us better understand the advantages of the original vehicle call in the recovery task. Through this detailed comparative analysis, we can further verify whether sending new vehicles alone for recovery tasks can result in cost optimization providing a scientific basis for decision-makers.

Table 8. Order insertion recovery point basic information.

Num	x	y	$[e_i, l_i]$	$[E_i, L_i]$	r_i	t_{ci}
21	40	55	12:00–13:00	11:30–13:30	5	5
22	10	50	13:00–14:00	12:30–14:30	5	5
23	12	48	13:00–14:00	12:30–14:30	5	5
24	40	25	12:00–13:00	11:30–13:30	5	5
25	20	20	12:00–13:00	11:30–13:30	5	5

Table 9. Insert order path.

Recycling vehicle number	Recovery routes
1	0- 1-1-3-1-3-2-2-10-6-1-11-12-8-0
2	0- 9-1-12-10-6-12-13-11-7-6-9-9-9-0
3	0- 2-8-13-13-1-1-9-9-3-6-1-12-9-0
4	0- 10-1-7-3-1-11-12-12-13-13-13-12-0

Table 10. The initial recycling route.

Recycling vehicle number	Recycling routes
1	0- 3-2-1-12-12-10-1-15-7-7-5-8-11-9-5-0
2	0-10-6-14-7-3-1-1-14-13-6-13-3-13-13-2-0
3	0- 2-12-10-3-3-5-2-9-3-1-13-12-12-0
4	0- 4-10-2-12-5-6-3-13-11-6-5-13-12-0

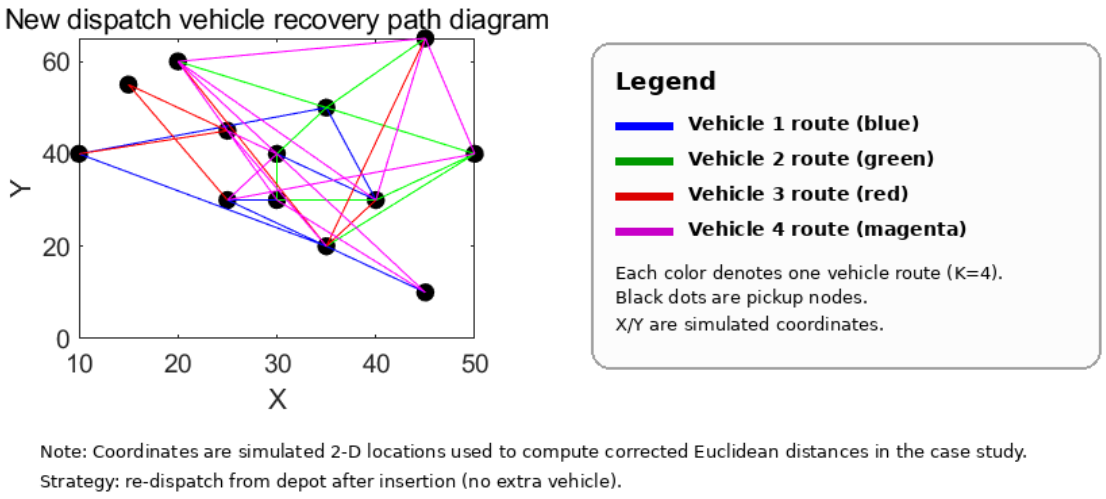

Figure 6. New dispatch vehicle recovery path.

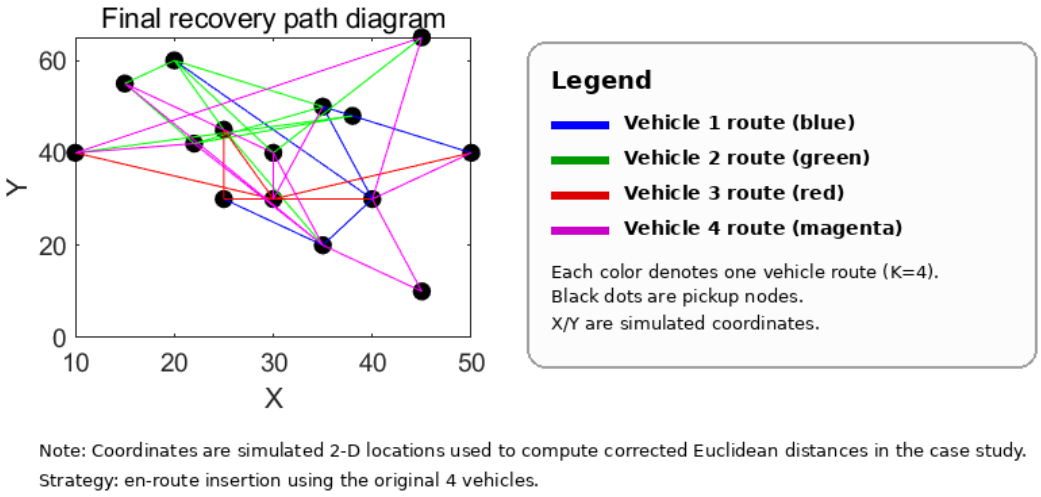

Figure 7. Optimized path map using the original vehicle.

On this basis, consider directly deploying the vehicles that are already in the middle of the recovery task. The specific information and data for this route are detailed in Table 10 and presented graphically in Figure 7. These charts show the complete process of starting the vehicle from the starting point, going through a series of path choices, and finally reaching the mission site.

Table 11 shows that the recycling method we suggest, which prioritizes sorting and uses vehicles in a specific order, saves more money overall than the method where a different vehicle is assigned for recycling. Specifically, there are savings in four areas: fixed costs, vehicle wear and tear costs, time penalty costs, and fuel and carbon emission costs. The time penalty costs are particularly optimized, as using the original vehicle will save more than 200%. While sending another car can reduce cleaning costs by fulfilling orders promptly, cleaning costs are a smaller percentage of the total cost and are reduced by only about 10%. Hence, prioritizing vehicles for tasks has a more significant advantage.

Table 11. Cost comparison of the two recovery modes.

Cost classification	Use the original vehicle/Yuan (USD)	Send another vehicle/Yuan (USD)	Increased percentage (Another/Original)
total cost	939.17 (USD131.79)	1650.48 (USD231.67)	75.74%
fixed costs	400 (USD56.14)	500 (USD70.17)	25%
vehicle wear and tear costs	235.40 (USD33.04)	283.54 (USD39.80)	20.45%
cleaning costs	112.26 (USD15.75)	100.64 (USD14.13)	–10.35%
time penalty cost	70.70 (USD9.92)	216.80 (USD30.43)	206.65%
fuel and carbon costs	356.21 (USD49.99)	549.50 (USD77.13)	54.26%

Due to the complexity of the real-world scenarios, there is still room for further optimization of this kind of model: (1) Compared to adding new orders, fixed-point recycling, which happens every day, discusses whether the recycling spot used by the vehicle should stay the same even after new orders are added, to suit the staff's usual habits. (2) The cost of recycling is also determined by other factors, such as weather, road conditions, etc., especially during the peak hours of commuting, when the speed of the recycling vehicles will vary greatly. (3) Since the amount of recycling for dynamic orders is relatively tiny, is it feasible to consider using different models of recycling vehicles to save costs.

3.3. Comprehensive benefit comparison across recycling modes

To demonstrate the comprehensive advantages of the proposed fixed-point recycling + dynamic order insertion hybrid mode, we extend the comparison beyond the two insertion strategies and benchmark our approach against two widely used operational modes. Specifically, we consider: Mode A (fixed-point recycling only), where vehicles follow the pre-planned fixed-point routes and newly arrived requests are postponed to the next cycle (or treated as unserved within the current cycle); Mode B (on-demand re-dispatch), where the fleet is re-dispatched from the depot and routes are re-optimized upon order arrivals; and Mode C (hybrid mode), where new requests are inserted into the ongoing fixed-point tours (en-route insertion), which corresponds to the proposed strategy.

We evaluate these modes using a multi-dimensional performance set: economic performance (total cost and cost breakdown), environmental performance (total fuel consumption and carbon emissions, reported as physical emissions E in addition to the carbon cost), social performance (service quality measured by customer satisfaction and participation rate), and long-term operational performance (cost per served request and fleet utilization over multiple insertion rounds). For social benefits, we compute the average satisfaction $\bar{S}$ based on the satisfaction function in Section 2 and define the participation rate as the proportion of requests whose satisfaction exceeds a predefined threshold S_{min}. For long-term benefits, we conduct a rolling-horizon simulation with repeated order insertions and report aggregated metrics over the entire operating horizon.

Table 12 summarizes the comparative results. Compared with Mode A, the hybrid mode improves service quality by serving newly arrived requests within the ongoing cycle, thereby increasing satisfaction/participation while keeping costs controlled. Compared with Mode B, the hybrid mode reduces redundant vehicle repositioning and avoids frequent re-dispatching, leading to lower emissions intensity and more stable long-run operational performance.

Table 12. Comprehensive performance comparison across recycling modes.

Metric	Mode A: Fixed-point only (Yuan / USD)	Mode B: Dispatch additional vehicle (Yuan / USD)	Mode C: Hybrid—reuse original vehicles (Yuan / USD)
Operational and social metrics			
Orders served / total	20 / 25	25 / 25	25 / 25
Participation rate PR (current-cycle fulfillment)	80.00%	100.00%	100.00%
Vehicles activated (implied by fixed costs)	4	5	4
Cost classification			
Total cost CCC	742.64 (USD 104.22)	1650.48 (USD 231.67)	939.17 (USD 131.79)
Fixed costs	400.00 (USD 56.13)	500.00 (USD 70.17)	400.00 (USD 56.14)
Wear and tear costs	102.14 (USD 14.34)	283.54 (USD 39.79)	235.40 (USD 33.04)
Cleaning costs	67.22 (USD 9.43)	100.64 (USD 14.13)	112.26 (USD 15.75)
Time-penalty cost	42.77 (USD 6.00)	216.80 (USD 30.43)	70.70 (USD 9.92)
Fuel and carbon costs	130.51 (USD 18.32)	549.50 (USD 77.13)	356.21 (USD 49.99)
Environmental and long-run operation metrics			
Estimated total distance DDD (km)*	134.69	567.08	367.61
Fuel consumption (L)*	13.47	56.71	36.76
Carbon emissions (kg CO_2_22)*	35.42	149.14	96.68
Cost per served order (Yuan/order)	37.13	66.02	37.57
Time-penalty per served order (Yuan/order)	2.14	8.67	2.83
CO_2_22 per served order (kg/order)*	1.77	5.97	3.87
Distance per activated vehicle (km/vehicle)*	33.67	113.42	91.9

3.4. Algorithm performance analysis

The experiment was programmed with MATLAB 2023b. To facilitate the comparison, the data from the above case is used. After the order insertion, the simple genetic algorithm (SGA) is used to synchronize the experiment. To ensure reproducibility and to justify the key GA parameters, we

determine the algorithm configuration using a two-stage calibration procedure: (i) literature-informed screening and (ii) pilot tuning on a representative instance.

Stage 1: Literature-informed screening.

For permutation-based routing problems with OX crossover and mutation operators, typical values of crossover probability and mutation probability are often selected within moderate ranges to balance exploration and exploitation. Following this principle, we initially consider $p_c \in [0.6, 0.8]$ and $p_m \in [0.01, 0.10]$. The population size is set to be sufficiently larger than the chromosome length to maintain diversity while keeping runtime acceptable for real-time insertion scenarios. In our encoding, the chromosome length is 39, and candidate population sizes are selected as 50–100.

Stage 2: Pilot tuning and final configuration.

We further conduct a pilot parameter tuning experiment by jointly varying mutation rate, crossover rate, population size, and number of generations at three representative levels each (a full factorial $3^4 = 81$ combinations; provided in Appendix Table A1). Each configuration is evaluated using two criteria: (1) the best objective value achieved (total cost) and (2) runtime/convergence speed. Since the proposed method targets dynamic order insertion, we prioritize configurations that achieve stable convergence within a short computation time, rather than pursuing marginal improvements with excessively long runtime. Based on the above trade-off, we adopt the final parameter setting summarized in Table 13, which provides robust performance and satisfactory convergence in the case study.

Table 13. Algorithm parameter table.

Parameter	Value
population size	50
chromosome length	39
crossover probability	0.7
mutation probability	0.1
number of iterations	100

In addition, all stochastic runs are executed under controlled random seeds for fair comparison across algorithms, and the termination condition is set as a fixed number of iterations (generations).

In a comparative study of the improved genetic algorithm presented in this paper with conventional ones, we pay particular attention to the performance of these two algorithms during the first 50 iterations. Through comparative analysis, we find that the improved genetic algorithm proposed in this paper converges significantly faster than the traditional genetic algorithm under the given parameter conditions. To ensure the reliability and statistical significance of the results, we performed 20 independent running experiments on the two algorithms. The proposed improved genetic algorithm showed consistent efficient convergence performance in 20 runs in these experiments. In contrast, the traditional genetic algorithm converges relatively slowly under the same experimental conditions, which is 6 times more effective than the results of the improved algorithm. This comparative result fully proves that the enhanced GA presented here has significant advantages in convergence speed and performs well in stability, effectively improving the performance of the genetic algorithm.

Figure 8 shows the iteration diagram of the improved genetic algorithm, and Figure 9 shows the iterative process of the simple genetic algorithm, through which we can see that the SGA's

convergence rate is relatively slow. This means that the algorithm requires multiple iterations that consume more time and computational resources to reach the optimal solution. Moreover, the graph indicates that the simple genetic algorithm may be more effective in local search but may not consistently find the optimal solution. Ultimately, despite several iterations, the convergence effect of SGA has not improved significantly, and there is still a gap between the IGA in the ideal state.

In order to verify the applicability of the model, a systematic sampling method was used to simulate the experimental data. The values of 0.01, 0.055, and 0.1 were extracted from mutation probability, 0.6, 0.7, and 0.8 from crossover probability, 50, 75, and 100 from population size, and 100, 550, and 1000 from the number of iterations. Due to space constraints, the detailed calculation results are presented in Appendix A. To assess the robustness of the above findings, we conduct a brief sensitivity analysis on key model parameters (time-window penalty and major cost coefficients). Detailed results are reported in Appendix B.

After adjusting the parameters, it can be seen that the convergence rate and the optimal cost of the model are within a reasonable range, indicating that the model has a good adaptive ability to different parameters.

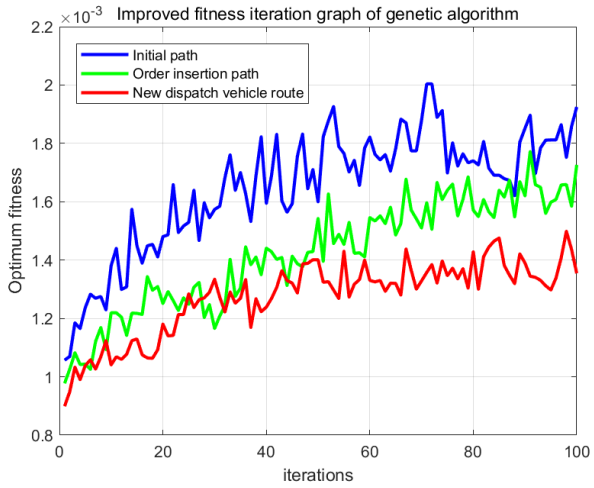

Figure 8. IGA iteration diagram.

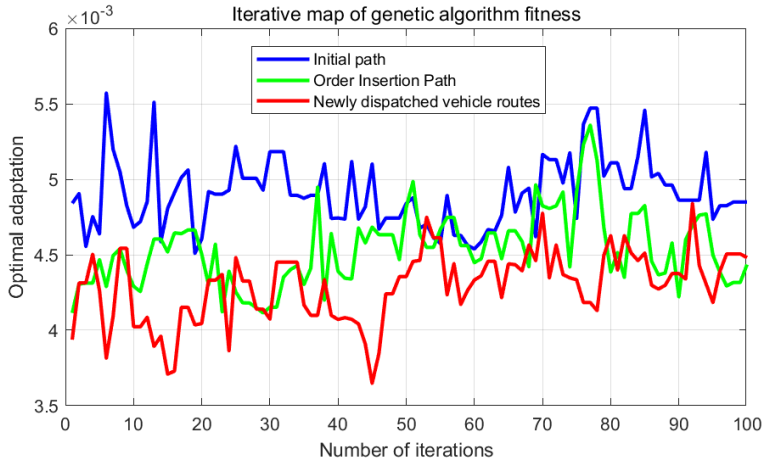

Figure 9. SGA iteration diagram.

4. Discussion

4.1. Cost-saving mechanisms of en-route insertion

This study shows that prioritizing en-route insertion (i.e., inserting newly arrived pickup requests into vehicles that are already executing routes) can reduce total operating cost relative to the conventional strategy of dispatching an additional vehicle. The cost advantage comes from four mechanisms.

First, fixed deployment costs are avoided when insertion is handled by active vehicles. In practice, dispatching an additional vehicle trigger fixed costs such as vehicle activation, driver assignment, and operational coordination. When the marginal insertion demand can be absorbed by en-route vehicles, these fixed costs are eliminated.

Second, en-route insertion improves time-related performance, thereby reducing waiting/lateness penalties under the (soft) time-window setting. Compared with waiting for a new vehicle to depart and travel to the service area, insertion into an ongoing tour typically shortens response time, which directly lowers the time penalty component.

Third, en-route insertion can decrease variable operating costs (e.g., energy and emissions) by reducing redundant vehicle movements. Even when the inserted requests add some detours, the total distance and energy consumption can still be lower than operating an additional vehicle for a separate tour, especially when the inserted requests are geographically close to the current route.

Finally, the two strategies may differ in cleaning-related costs. Although dispatching a new vehicle may sometimes reduce cleaning time due to shorter task chains, this component is generally a smaller share of total cost and is often insufficient to offset the additional fixed and time-related costs of activating a new vehicle. Overall, the insertion-first strategy offers a consistent cost-control advantage in the case study, mainly through fixed-cost avoidance and improved time performance.

The magnitude of cost reduction is partly structure-driven and partly scenario-dependent. Structurally, reusing the original vehicles avoids additional vehicle activation and associated fixed and operating costs, and reduces dispatch-related delays. However, the savings percentage depends on the insertion context, including the spatial dispersion of new orders, the tightness of time windows, baseline route utilization, and vehicle capacity slack. In scenarios with highly dispersed new orders or very tight time windows, dispatching an additional vehicle may become competitive, which defines the boundary conditions of the proposed hybrid strategy. In addition to cost savings, the proposed hybrid mode delivers environmental and service-level benefits by reducing redundant vehicle movements and improving customer satisfaction/participation within the current operating cycle. Moreover, rolling-horizon results indicate that the hybrid mode achieves more stable long-term performance in terms of cost per served request and fleet utilization compared with purely fixed-point or purely on-demand operations.

4.2. Managerial implications and implementation considerations

The results suggest a practical decision rule for operators: prioritize en-route insertion whenever feasibility constraints allow, and use additional vehicles only when insertion would violate service commitments or operational limits. This has three managerial implications.

Decision support for real-time dispatch: The proposed model can be integrated as a

decision-support module that evaluates, at each arrival of new requests, whether insertion into an ongoing tour is feasible and cost-effective. This supports time-sensitive recycling operations and reduces unnecessary vehicle activation.

Operational feasibility in real settings: Implementation should explicitly account for factors that drive feasibility and cost in practice, including vehicle capacity, time windows, and route stability. A phased rollout is recommended: start with a pilot region and limited time periods, monitor key KPIs (total cost, response time, on-time rate, and emissions), and then scale up to larger areas once the decision logic is validated.

Data and system requirements: Effective deployment requires basic interoperability with operational data sources (order streams, vehicle states, and location data). To improve reliability, the dispatch system should support (i) periodic route re-optimization, (ii) exception handling for congestion/weather disruptions, and (iii) clear operational rules for when a driver can accept insertion tasks. Training and lightweight user interfaces are important to ensure that dispatch recommendations can be executed smoothly. Finally, because real-time insertion is time-sensitive, adequate computation resources (or simplified evaluation rules) should be ensured to maintain fast response times during peak demand.

4.3. Limitations and future research

Several limitations should be noted. First, travel distance/time is approximated using corrected Euclidean distances, while real travel time and fuel consumption are affected by road network structure, congestion, and weather. Second, the current cost model assumes that fuel consumption and related losses are independent of carried load; however, payload varies along the route and may significantly affect fuel use and emissions. Third, when multiple insertion requests arrive within a short time window, capacity feasibility and route stability may become binding and reduce the reliability of insertion-based decisions.

Future work can extend the current framework in four directions: (i) Replace Euclidean-distance approximation with road-network-based shortest paths and time-dependent travel speeds using GIS/traffic data, (ii) incorporate load-dependent fuel/emission functions and heterogeneous vehicle types to better reflect operational reality, (iii) consider stochastic or robust optimization to handle uncertain travel times and bursty order arrivals in real-time insertion, (iv) validate the model on larger-scale operational datasets and explore additional objectives (e.g., service equity across regions and long-term station planning), potentially combining prediction models for insertion demand with routing optimization.

5. Conclusions

This paper addresses the problem of efficiently delivering tableware recycling, with customer satisfaction as the constraint. Fixed recovery costs, vehicle loss costs, cleaning costs, time penalty costs, fuel costs, and carbon emission costs are incorporated into the target function. A path optimization model for tableware delivery that considers dynamic order insertion is developed, and an improved genetic algorithm is designed to solve the model. The numerical experimental results show that: (1) compared with the traditional genetic algorithm, the proposed improved genetic algorithm presents significant advantages regarding convergence speed and stability; and (2) in the

case of order insertion, the scheme proposed in this paper is to prioritize the use of vehicles in the recovery task, which reduces the cost compared with the traditional approach of sending another vehicle for recovery.

Relevant research directions and challenges are discussed, and some suggestions for improvement are provided. It is hoped that, through further experiments and analysis, more practical solutions for takeaway tableware recycling can be developed to support environmental protection policies.

Author contributions

Chenghan He: Conceptualization, Methodology, Software, Writing–original draft, Writing–review & editing; Dexin Huang: Formal analysis, Validation, Data curation, Visualization; Chengyin Wang: Investigation, Resources, Project administration, Supervision.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

We would like to thank you for following the instructions above very closely in advance. It will definitely save us lot of time and expedite the process of your paper's publication.

Conflict of interest

The authors declare no conflicts of interest.

References

1. Y. Q. Mo, L. X. Li, Gig economy and entrepreneurship: Evidence from the entry of food delivery platforms, (Chinese), *Journal of Management World*, **38** (2022), 31–45. https://doi.org/10.19744/j.cnki.11-1235/f.2022.0018
2. B. Yang, X. J. Xia, Y. Y. Cheng, Catering industry during the COVID-19 pandemic: Impact and differentiation, (Chinese), *Journal of Hohai University (Philosophy and Socail Sciences)*, **23** (2021), 31–40. https://doi.org/10.3876/j.issn.1671-4970.2021.01.005
3. G. Blanca-Alcubilla, A. Bala, N. de Castro, R. Colomé, P. Fullana-i-Palmer, Is the reusable tableware the best option? Analysis of the aviation catering sector with a life cycle approach, *Sci. Total Environ.*, **708** (2020), 135121. https://doi.org/10.1016/j.scitotenv.2019.135121
4. V. Linderhof, F. H. Oosterhuis, P. J. H. van Beukering, H. Bartelings, Effectiveness of deposit-refund systems for household waste in the Netherlands: applying a partial equilibrium model, *J. Environ. Manage.*, **232** (2019), 842–850. https://doi.org/10.1016/j.jenvman.2018.11.102

5. Q. Y. Xu, Z. Shao, Y. He, Optimal delivery strategies for packing box recycling in online platforms, *J. Clean. Prod.*, **276** (2020), 124273. https://doi.org/10.1016/j.jclepro.2020.124273
6. K. Govindan, M. Palaniappan, Q. H. Zhu, D. Kannan, Analysis of third-party reverse logistics provider using interpretive structural modeling, *Int. J. Prod. Econ.*, **140** (2012), 204–211. https://doi.org/10.1016/j.ijpe.2012.01.043
7. D. Pamucar, K. Chatterjee, E. K. Zavadskas, Assessment of third-party logistics provider using multi-criteria decision-making approach based on interval rough numbers, *Comput. Ind. Eng.*, **127** (2019), 383–407. https://doi.org/10.1016/j.cie.2018.10.023
8. N. Zarbakhshnia, Y. Wu, K. Govindan, H. Soleimani, A novel hybrid multiple-attribute decision-making approach for outsourcing sustainable reverse logistics, *J. Clean. Prod.*, **242** (2020), 118461. https://doi.org/10.1016/j.jclepro.2019.118461
9. Z.-S. Chen, X. Zhang, K. Govindan, X.-J. Wang, K.-S. Chin, Third-party reverse logistics provider selection: a computational semantic analysis-based multi-perspective multi-attribute decision-making approach, *Expert Syst. Appl.*, **166** (2021), 114051. https://doi.org/10.1016/j.eswa.2020.114051
10. M. Agovino, M. D'Uva, A. Garofalo, K. Marchesano, Waste management performance in Italian provinces: efficiency and spatial effects of local governments and citizen action, *Ecol. Indic.*, **89** (2018), 680–695. https://doi.org/10.1016/j.ecolind.2018.02.045
11. P. Rathore, S. P. Sarmah, Modeling and identification of suitable motivational mechanism in the collection system of municipal solid waste supply chain, *Waste Manag.*, **129** (2021), 76–84. https://doi.org/10.1016/j.wasman.2021.05.011
12. A. Taweesan, T. Koottatep, C. Polprasert, Effective measures for municipal solid waste management for cities in some Asian countries, *Expo. Health*, **9** (2017), 125–133. https://doi.org/10.1007/s12403-016-0227-5
13. J. Heydari, K. Govindan, A. Jafari, Reverse and closed-loop supply chain coordination by considering government role, *Transp. Res. D: Transp. Environ.*, **52** (2017), 379–398. https://doi.org/10.1016/j.trd.2017.03.008
14. J. H. Yang, R. Y. Long, H. Chen, Q. Q. Sun, A comparative analysis of express packaging waste recycling models based on the differential game theory, *Resour. Conserv. Recy.*, **168** (2021), 105449. https://doi.org/10.1016/j.resconrec.2021.105449
15. X. Tong, D. Y. Tao, The rise and fall of a "waste city" in the construction of an "urban circular economic system": The changing landscape of waste in Beijing, *Resour. Conserv. Recy.*, **107** (2016), 10–17. https://doi.org/10.1016/j.resconrec.2015.12.003
16. G.-E.-K. Berthomé, A. Thomas, A context-based procedure for assessing participatory schemes in environmental planning, *Ecol. Econ.*, **132** (2017), 113–123. https://doi.org/10.1016/j.ecolecon.2016.10.014
17. Q. Song, Application of an optimized beam-PSO algorithm in multi-trip vehicle routing problem, (Chinese), *Computer Engineering & Science*, **41** (2019), 1882–1891.
18. C. Wang, C. Liu, D. Mu, Y. Gao, VRPSPDTW problem solving by discrete cuckoo search, (Chinese), *Computer Integrated Manufacturing Systems*, **24** (2018), 570–582. https://doi.org/10.13196/j.cims.2018.03.004
19. H. M. Fan, J. X. Wu, J. Geng, Y. Li, Hybrid genetic algorithm for solving fuzzy demand and time windows vehicle routing problem, (Chinese), *Journal of Systems & Management*, **29** (2020), 107–118.

20. G. W. Huang, Y. G. Cai, Y. H. Qi, H. R. Chen, S. H. Wang, Adaptive genetic grey wolf optimizer algorithm for capacitated vehicle routing, (Chinese), *Acta Electronica Sinica*, **47** (2019), 2602–2610.
21. S. Majidi, S.-M. Hosseini-Motlagh, J. Ignatius, Adaptive large neighborhood search heuristic for pollution-routing problem with simultaneous pickup and delivery, *Soft Comput.*, **22** (2018), 2851–2865. https://doi.org/10.1007/s00500-017-2535-5
22. T. Zhang, P. Z. Lai, Q. F. He, Z. H. Jin, Optimization of dynamic vehicle routing of urban distribution based on real-time information, (Chinese), *Systems Engineering*, **33** (2015), 58–64.
23. J. L. Zhang, Y. W. Zhao, H. Y. Wang, J. Jie, W. L. Wang, Modeling and algorithms for a dynamic multi-vehicle routing problem with customers' dynamic requests, (Chinese), *Computer Integrated Manufacturing Systems*, **16** (2010), 543–550. https://doi.org/10.13196/j.cims.2010.03.97.zhangjl.026
24. J. L. Shi, J. Zhang, Optimization on simultaneous pick-up and delivery vehicle routing problem with split delivery and stochastic travel and service time, (Chinese), *Control and Decision*, **33** (2018), 657–670. https://doi.org/10.13195/j.kzyjc.2017.0317
25. L. Y. Zhang, J. Zhang, B. Xiao, Multi-objective O2O take-out instant delivery routing optimization considering customer priority, (Chinese), *Industrial Engineering and Management*, **26** (2021), 196–204. https://doi.org/10.19495/j.cnki.1007-5429.2021.02.024
26. X. M. Zhang, J. M. Chen, J. Xiao, Stochastic dynamic multi-vehicles pick-up and delivery problem with heavy traffic and its solution policy, (Chinese), *Journal of Systems Engineering*, **27** (2012), 61–68.
27. H. T. Lei, G. Laporte, B. Guo, The vehicle routing problem with stochastic demands and split deliveries, *INFOR: Information Systems and Operational Research*, **50** (2012), 59–71. https://doi.org/10.3138/infor.50.2.059

Appendix A

Table A1. Model parameters.

Mutation rate	Crossover rate	Population size	Number of generations	Rate of convergence (s)	Optimal solution/Yuan (USD)
0.1	0.6	50	100	2.216	988.4004 (USD139.39)
			550	8.389	837.1398 (USD118.06)
			1000	14.289	777.3294 (USD109.63)
		75	100	3.010	934.6655 (USD131.81)
			550	10.090	547.8486 (USD77.26)
			1000	22.140	548.9114 (USD77.41)
		100	100	3.835	934.5324 (USD131.80)
			550	16.845	662.8409 (USD93.48)
			1000	30.373	222.6159 (USD31.40)
	0.7	50	100	2.288	939.1742 (USD131.79)
			550	8.816	683.4309 (USD96.38)
			1000	15.230	813.174 (USD114.68)
		75	100	3.085	1088.868 (USD153.56)

Continued on next page

Mutation rate	Crossover rate	Population size	Number of generations	Rate of convergence (s)	Optimal solution/Yuan (USD)
0.055	0.8	100	550	12.777	473.5608 (USD66.79)
			1000	17.551	349.5841 (USD49.30)
		75	100	3.907	840.4376 (USD118.53)
			550	17.507	418.5536 (USD59.03)
			1000	30.321	182.1538 (USD25.69)
		50	100	2.462	965.9659 (USD136.23)
			550	8.320	547.6708 (USD77.24)
			1000	14.779	683.2396 (USD96.36)
	0.6	100	100	3.028	919.7602 (USD129.71)
			550	12.607	415.8298 (USD58.64)
			1000	32.850	382.1596 (USD53.90)
		75	100	3.370	936.3321 (USD132.05)
			550	17.942	514.8307 (USD72.61)
			1000	31.587	441.6423 (USD62.28)
		50	100	2.284	719.9238 (USD101.53)
			550	8.595	663.2232 (USD93.57)
			1000	14.587	522.4058 (USD73.70)
	0.7	100	100	3.319	906.2887 (USD127.86)
			550	10.429	538.7476 (USD76.01)
			1000	22.754	534.2266 (USD75.37)
		75	100	3.110	878.0127 (USD123.87)
			550	19.466	483.9952 (USD68.28)
			1000	47.741	251.5756 (USD35.49)
		50	100	2.372	1225.3498 (USD172.88)
			550	8.815	657.5642 (USD92.77)
			1000	15.008	638.106 (USD90.03)
	0.8	100	100	3.214	724.8134 (USD102.26)
			550	11.524	638.092 (USD90.02)
			1000	23.619	367.5053 (USD51.85)
		75	100	4.135	867.1107 (USD122.34)
			550	18.211	350.1929 (USD49.41)
			1000	24.122	303.1264 (USD42.77)
		50	100	2.451	971.1642 (USD137.02)
			550	8.965	699.3819 (USD98.67)
			1000	15.557	371.2569 (USD52.38)
		100	100	3.222	923.0593 (USD130.23)
			550	11.473	524.5995 (USD74.01)
			1000	23.979	416.4266 (USD58.75)
		75	100	4.391	967.9873 (USD136.57)
			550	18.663	455.5048 (USD64.26)

Continued on next page

Mutation rate	Crossover rate	Population size	Number of generations	Rate of convergence (s)	Optimal solution/Yuan (USD)
0.01	0.6	50	1000	32.608	312.6343 (USD44.11)
			100	2.201	790.4193 (USD111.52)
			550	9.106	872.7977 (USD123.14)
			1000	15.249	674.1007 (USD95.1)
		75	100	2.668	800.1414 (USD112.89)
			550	13.480	704.1061 (USD99.34)
			1000	25.330	619.9645 (USD87.47)
		100	100	4.001	724.991 (USD102.28)
			550	14.154	557.514 (USD78.66)
			1000	31.786	482.0295 (USD68.01)
	0.7	50	100	2.396	967.9751 (USD136.57)
			550	26.934	794.0319 (USD112.02)
			1000	30.659	714.444 (USD100.8)
		75	100	3.237	905.1715 (USD127.7)
			550	13.590	549.2548 (USD77.49)
			1000	19.156	508.0898 (USD71.68)
		100	100	4.105	758.6384 (USD107.03)
			550	17.893	461.3124 (USD65.08)
			1000	32.363	426.43 (USD60.16)
	0.8	50	100	2.609	951.8067 (USD134.28)
			550	9.135	662.2082 (USD93.43)
			1000	15.264	613.6152 (USD86.57)
		75	100	3.185	928.9397 (USD131.06)
			550	14.345	542.93 (USD76.6)
			1000	18.918	625.215 (USD88.21)
		100	100	4.307	762.6635 (USD107.6)
			550	19.095	465.049 (USD65.61)
			1000	32.500	518.6527 (USD73.17)

Appendix B

Table A2. Sensitivity analysis of key parameters.

Parameter	Multiplier	B_Total Cost	C_Total Cost	Gap_Bminus C	B_Time Penalty	C_Time Penalty
	0.8	1000856.848	780.655264	1000076.193	257.137926	189.221239
alpha2	1	1000873.014	769.478278	1000103.536	273.304283	190.15129
	1.2	1000889.181	769.478278	1000119.703	289.470639	190.15129
	0.8	1000833.014	729.478278	1000103.536	273.304283	190.15129
F	1	1000873.014	769.478278	1000103.536	273.304283	190.15129
	1.2	1000913.014	809.478278	1000103.536	273.304283	190.15129
	0.8	1000872.514	768.978278	1000103.536	273.304283	190.15129
beta1	1	1000873.014	769.478278	1000103.536	273.304283	190.15129
	1.2	1000873.514	769.978278	1000103.536	273.304283	190.15129