mathematics

MDPI

Article

Bi-Objective Mixed Integer Nonlinear Programming Model for Low Carbon Location-Inventory-Routing Problem with Time Windows and Customer Satisfaction

Lihua Liu [1,2], Aneng He [1], Tian Tian [3], Lai Soon Lee [1,3,*] and Hsin-Vonn Seow [4]

[1] Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia; gs62124@student.upm.edu.my (L.L.); gs64722@student.upm.edu.my (A.H.)
[2] Faculty of Science, Guangxi University of Technology and Science, Liuzhou 545000, China
[3] Laboratory of Computational Statistics and Operations Research, Institute for Mathematical Research, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia; gs66197@student.upm.edu.my
[4] Faculty of Arts and Social Sciences, Nottingham University Business School, University of Nottingham Malaysia, Semenyih 43500, Selangor, Malaysia; hsin-vonn.seow@nottingham.edu.my
* Correspondence: lls@upm.edu.my

Abstract: In order to support a low-carbon economy and manage market competition, location–inventory–routing logistics management must play a crucial role to minimize carbon emissions while maximizing customer satisfaction. This paper proposes a bi-objective mixed-integer nonlinear programming model with time window constraints that satisfies the normal distribution of stochastic customer demand. The proposed model aims to find Pareto optimal solutions for total cost minimization and customer satisfaction maximization. An improved non-dominated sorting genetic algorithm II (IMNSGA-II) with an elite strategy is developed to solve the model. The model considers cost factors, ensuring that out-of-stock inventory is not allowed. Factors such as a carbon trading mechanism and random variables to address customer needs are also included. An entropy weight method is used to derive the total cost, which is comprised of fixed costs, transportation costs, inventory costs, punishment costs, and the weight of carbon emissions costs. The IMNSGA-II produces the Pareto optimal solution set, and an entropy–TOPSIS method is used to generate an objective ranking of the solution set for decision-makers. Additionally, a sensitivity analysis is performed to evaluate the influence of carbon pricing on carbon emissions and customer satisfaction.

Keywords: location inventory routing; carbon trading scheme; customer satisfaction; NSGA-II; entropy–TOPSIS

MSC: 90B06; 90C29

Citation: Liu, L.; He, A.; Tian, T.; Lee, L.S.; Seow, H.-V. Bi-Objective Mixed Integer Nonlinear Programming Model for Low Carbon Location-Inventory-Routing Problem with Time Windows and Customer Satisfaction. *Mathematics* **2024**, *12*, 2367. https://doi.org/10.3390/math12152367

Academic Editors: Shiv Raj Singh, Dharmendra Yadav and Himani Dem

Received: 20 June 2024
Revised: 23 July 2024
Accepted: 27 July 2024
Published: 29 July 2024

1. Introduction

The emergence and advancement of the cold chain logistics (CCL) network have significantly contributed to meeting the demand for fresh food from various parts of the world. However, customers now have increasingly high expectations, expecting fresh food orders to arrive within specific time frames. The carbon emissions from the CCL transportations are relatively high. Therefore, optimizing the distribution network in the CCL must include being accountable for the overall cost of the fresh food, customer satisfaction, and carbon emissions, as highlighted by [1].

Customer satisfaction (CS) measures how well customers feel about the products and services they receive [2]. Numerous studies have demonstrated the positive relationship between CS and important business outcomes such as customer loyalty, repeat purchase intention, positive word of mouth, and market share [3]. Ensuring high levels of CS can

generate significant value for enterprises. In order to achieve this, especially with intense market competition, it is crucial to prioritize CS by considering the customer time window (TW) in the integrated optimization of the location–inventory–routing problem (LIRP).

Rapid economic growth and increased consumption have significantly increased carbon emissions [4]. Reducing greenhouse gas emissions has become a pressing global concern [5]. Statistical data show that more than 70% of the global transportation sector's carbon emissions come from land vehicles (see details at [6]). The CCL network, in particular, is recognized as a high-energy and high-carbon emission industry [7]. Therefore, optimizing distribution paths to minimize energy consumption and carbon emissions is of utmost importance.

This paper primarily focuses on determining the optimal transfer distribution center (DC) from a set of alternative distribution centers, taking into account the soft time window (STW) and other specified constraints. This involves establishing the transportation volume from the factory to the selected DCs, representing the ordered quantity for each DC. Subsequently, an optimal path is identified from each DC to every distributor to minimize the total supply chain cost (TSCC) and maximize CS.

In short, the CCL network process necessitates consideration of both CS and environmental benefits, prompting several crucial research questions: (i) How can CS be evaluated within the context of LIRP? (ii) How can the STW factor be incorporated, potentially through the implementation of a time penalty strategy? (iii) How can the cost associated with carbon emissions be accurately calculated? (iv) How can the optimization model that considers TSCC, CS, carbon emissions and environmental impacts (CEEI), and STW factors be solved?

The remainder of this paper is organized as follows: the related works of LIRP are presented in Sections 2–4, which define the fundamental assumptions and model parameters of the proposed model. Section 5 formulates the bi-objective mixed-integer nonlinear programming (MINLP) model with CS and stochastic demand following a normal distribution. In Section 6, an improved nondominated sorting genetic algorithm II (IMNSGA-II) with an elite strategy is proposed to solve the LIRP model, along with the introduction of the entropy–TOPSIS method. Section 7 investigates the effectiveness of the proposed IMNSGA-II and utilizes it to validate the MINLP model using benchmark data. Additionally, a case study of a CCL enterprise in Jinan City, Shandong Province, China is used to verify the model. The sensitivity analysis of CEEI and CS in the stochastic demand of the LIRP model decision outcomes is discussed in Section 8. Finally, conclusions and future research are provided in Section 9.

2. Literature Review

CCL is a specialized supply chain that maintains strict temperature requirements during product handling, storage, and transportation. In 2016, ref. [8] proposed a P-center selection method to determine the transshipment point between customers in a CCL LIRP. This method was evaluated using both benchmark and real data, including a case study of frozen chicken delivery by a local company in Selangor, Malaysia. Ref. [9] proposed a new LIRP model for CCL management, using the NSGA-II algorithm to solve the nonlinear integer programming model and optimize multiple objectives such as location selection, inventory, and transportation cost simultaneously. Ref. [10] introduced the LIRP optimization model with a path STW constraint in CCL and developed an improved multi-objective ant colony optimization (MACO) algorithm to solve it.

Moreover, ref. [11] studied a low-carbon vehicle routing problem considering CS in a practical CCL network and proposed an adaptive genetic algorithm (AGA) method to solve it. However, there was no comprehensive integration of LIRP. Ref. [12] investigated a bi-objective programming model that included LIRP. It considered total cost minimization and carbon emission cost minimization using a multi-objective hybrid harmony search–simulated annealing (MOHS-SA) algorithm to produce well-distributed Pareto-optimal solutions. Ref. [13] considered the time window and carbon trading mechanism to establish

a low-carbon environmental protection cold chain multi-objective LIRP model, solved using an improved GA. These studies did not incorporate CS based on time.

Other research, such as [14], proposed a CCL LIRP model considering carbon emissions and product freshness with hard time window constraints. They used the YALMIP toolbox to solve the model and obtain optimal solutions for this complex multi-objective problem. Product freshness in this study was also viewed as a type of CS research. Ref. [15] constructed a CCL LIRP model based on a broken-line STW and CS, solving it using an improved ant colony algorithm. These studies considered CS and carbon emissions but assumed deterministic demand for the fresh product.

In 2023, ref. [16] established a bi-objective LIRP model with the total cost and lost sales, respectively, in which stochastic demand is considered and carbon tax policy is applied to reduce emissions. The augmented epsilon constraint method is used to solve this model. The model still does not consider CS and does not use a metaheuristic algorithm to solve the model. The contribution of this paper is that it will simultaneously consider CS under the constraint of soft time windows and the cost of minimizing carbon emissions, with customer demand modeled as a stochastic variable in the cold chain LIRP.

As global warming worsens, China has implemented various measures to reduce carbon emissions. On 5 January 2021, China's Ministry of Ecology and Environment issued the "Carbon Emission Trading Management Measures (Trial)", and on 16 July 2021, China's carbon emission trading market officially began online trading. The carbon emission trading market allows for the public trading of carbon emission rights as an asset. The carbon trading system (Figure 1) regulates this market, promoting the reduction of global greenhouse gas and carbon dioxide emissions. The carbon trading mechanism is one of the most effective measures to control carbon emissions [17].

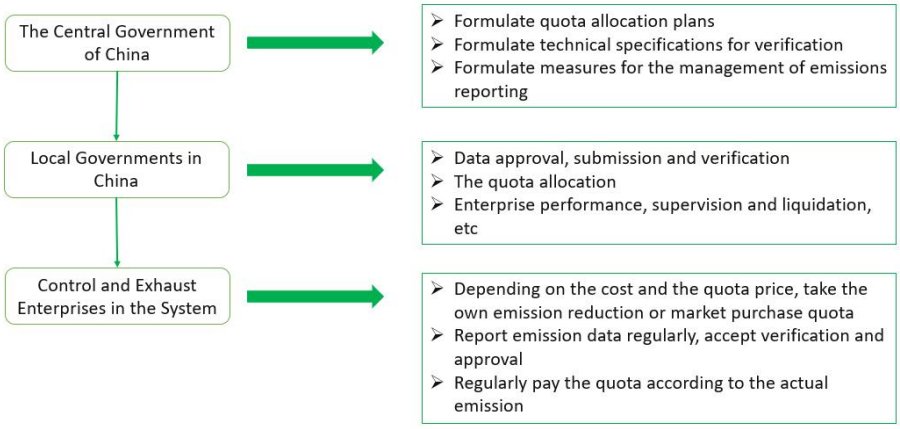

Figure 1. China's carbon trading system.

The research by ref. [18] focused on the optimized single-objective planning of carbon emissions, including operating costs, risks, and paths. The carbon emission cost calculation formula in this study is directly based on carbon emissions per kilometer and is obtained using CPLEX Opl version 12.5.1. Ref. [19] proposed a stochastic LIRP model for the biofuel supply chain aimed at minimizing total costs and carbon emissions, which is solved using a metaheuristic algorithm. The carbon emission formula in Asadi's model depends on the vehicle's weight, as well as the length and time of transportation. Additionally, ref. [20] proposed a LIRP model for a green supply chain network design, incorporating vehicle fuel consumption as a substitute for carbon emissions in the objective function, which was solved using the GAMS 24.1 software.

Ref. [13] comprehensively considered factors such as time windows and the carbon trading mechanism to establish a low-carbon ring. Given the high energy consumption, carbon emissions significantly impact CCL. Ref. [21] established a bi-objective programming model under uncertainty, including transportation costs due to fuel consumption, and the

performance evaluation was satisfaction. This problem was solved using a weighted fuzzy multi-objective solution method combined with CPLEX.

Recently, ref. [22] proposed the LRIP for CCL that incorporates a carbon trading mechanism. The model developed a multi-objective programming framework to optimize location, transportation, inventory, and carbon trading costs. Subsequently, the improved non-dominated sorting genetic algorithm II (NSGA-II) was employed to solve the multi-objective programming model. However, it should be noted that the model does not account for STW constraints and CS.

Based on the above literature, this paper will focus on establishing a correlation between carbon emissions, fuel consumption, and driving distance. Consequently, a carbon emission cost function based on the carbon trading mechanism is proposed. Table 1 presents a summary of key research on the LIRP synthesis problem based on relevant literature characteristics.

Table 1. Summary of related research on LIRP.

Author	Year	CCL	MOP	STW	CS	LC	(Meta) Heuristic	Stochastic	Case Study
Shariff et al. [8]	2016	✓							✓
Lerhlaly et al. [18]	2016					✓			
Zheng et al. [9]	2017	✓	✓				✓	✓	
Asadi et al. [19]	2018		✓			✓	✓	✓	✓
Li et al. [10]	2020	✓	✓	✓			✓	✓	
Gholipour et al. [20]	2020		✓			✓			✓
Liu, Zhu et al. [14]	2021	✓	✓			✓		✓	✓
Misni et al. [12]	2021	✓	✓			✓	✓	✓	
Zhu, Wen et al. [13]	2021	✓	✓	✓		✓	✓		✓
Shu et al. [15]	2021	✓		✓	✓		✓	✓	✓
Tavana et al. [21]	2021		✓		✓	✓			✓
Li et al. [22]	2022	✓	✓			✓	✓	✓	✓
Govidan et al. [16]	2023		✓			✓		✓	✓
This paper	2024	✓	✓	✓	✓	✓	✓	✓	✓

CCL: cold chain logistic; MOP: multi-objective programming; STW: Soft Time Window ; CS: customer satisfaction; LC: low carbon.

Back in 1995, ref. [23] introduced a genetic algorithm known as the non-dominated sorting genetic algorithm (NSGA). However, this algorithm had some shortcomings, such as high computational complexity when handling large amounts of data and the need to specify the shared radius. In the early 2000s, ref. [24] refined and enhanced the NSGA, leading to the development of the NSGA-II algorithm. The NSGA-II algorithm is widely recognized and utilized for solving multi-objective optimization problems, particularly those involving conflicting objectives like the ones discussed in this paper. It effectively addresses the main issues of the traditional NSGA algorithm by reducing the computational complexity from $O(nM^3)$ to $O(nM^2)$, where n represents the number of objective functions and M denotes the population size without needing a shared parameter and incorporating an optimal retention mechanism.

In 2018, ref. [25] established a bi-objective MINLP model for the LIRP, employing the NSGA-II and multi-objective particle swarm optimization (MOPSO) to solve large-scale problems. The comparison in this paper demonstrated that the NSGA-II outperformed MOPSO, indicating its superiority in uncovering high-quality solutions. Additionally, ref. [26] introduced three metaheuristic algorithms, NSGA-II, MOPSO, and the Pareto envelope selection algorithm (PESA-II), to solve a bi-objective LIRP model considering risk pool and STW. The NSGA-II exhibited a better performance than the other algorithms in terms of solution quality and convergence.

Similarly, ref. [27] utilized the NSGA-II algorithm to solve a bi-objective programming LIRP model. They explored different variants of the algorithm, including parallel cooperative and pure parallel methods. Compared to the strength Pareto evolutionary algorithm version II (SPEA-II), under the same conditions, the NSGA-II demonstrated

superior performance in terms of computation time (CT) and the number of non-dominated solutions (NDS).

This paper builds on these studies by leveraging the NSGA-II algorithm to optimize the cold chain LIRP model, considering customer satisfaction under STW constraints and minimizing carbon emissions while addressing stochastic customer demand.

3. Basic Assumptions

Some assumptions are given to better define the model's constraints:

1. Only a single variety of cold chain food is considered. Due to the wide variety of cold chain food, the temperature control, shelf life, and susceptibility of these foods are not the same. Thus, the items cannot be placed in the same type of vehicle.
2. Each customer is served by one DC.
3. Each developed DC provides a distribution service for the customer point it serves every working day, and the annual working days of all the developed DCs are the same.
4. The demand of each customer point is independent and normally distributed.
5. The delivery vehicles in the same store are of uniform specifications; that is, each vehicle has the same capacity, fuel consumption, and speed in transit.
6. Carbon emissions from transportation are considered.
7. The total carbon emission of the supply chain is limited.
8. Any unexpected situation that might occur during the distribution process is not considered, such as the change in production demand of the vehicle production center, traffic control, weather, etc.
9. Each distribution route can be served by multiple vehicles.
10. The STW factor is considered in the distribution, and the related penalty cost and time have some linear function relationship.
11. The sum of customer demand on any distribution line should not exceed the vehicle's load capacity, and the vehicle initiates its journey from the DC and completes its assigned tasks before returning to the same DC.

4. Model Parameters and Definitions

The following are the descriptions of the model parameters and decision variables.

Parameters

N : A factory node is denoted as $N = 0$.
H : A collection of potential locations of DCs.
J : A collection of distributors.
K : All delivery vehicles assemble.
g_h : Potential distribution center h fixed construction cost, $h \in H$.
d_{ij} : Distance between node i and node j, $i, j \in N \cup H \cup J$.
i_h : Distribution center h unit product inventory holding cost.
α : Probability of running out of stock, and $1 - \alpha$ is the corresponding service level.
z_α : Safety stock factor.
p : Unit price of goods ordered from the factory.
L : Lead time.
μ_j : Average demand of distributor j in the period.
σ_j : Standard deviation of demand of distributor j in the period.
c_h : Distribution center h maximum storage capacity service capacity.
C^{CAP} : Carbon allowances allocated along the supply chain.
C_1 : Fixed cost of delivery vehicles has nothing to do with the vehicle's carrying weight and the vehicle's driving distance.
FC_{ij} : Energy consumption cost of distribution vehicle.
T_c : Transportation cost of the delivery vehicle.
$[ET'_j, LT'_j]$: Expected delivery TW for distributor j in CS function.

α_1 : Penalty cost per unit time for the vehicle to arrive at distributor j before time ET'_j.
α_2 : Penalty cost per unit time for the vehicle to arrive at distributor j after time LT'_j.
$[eT_j, lT_j]$: Acceptable delivery TW for distributor j in CS function.
P_c : Penalty cost.
ρ : Vehicle fuel consumption per unit distance.
Q_X: Dead weight of vehicle.
$\rho(Q_{ij})$: In the process of the delivery vehicle from node i to node j, the load Q_{ij} is the fuel consumption per unit distance of the goods.
Q_0: Distribution vehicle weight.
Q_k: Maximum carrying capacity of the distribution vehicle.
ρ_0: Fuel consumption per unit distance at no load.
ρ^*: Fuel consumption per unit distance at full load.
Q_c: Total carbon emissions in distribution.
p_2: Unit price of fuel.
ω: Carbon emission coefficient.
C_c: Carbon emission cost.
C_e: Carbon tax, the environmental cost of consuming each unit of carbon emitted.
D: Maximum driving distance of delivery vehicles.

Decision Variables

x_h: Quantity ordered from the factory to the DC h.
x_{hj}: Transportation volume that DC h allocates to distributor j.

$$y_h = \begin{cases} 1, \text{ If establish the DC } h; \\ 0, \text{ Otherwise.} \end{cases}$$

$$z_{hj} = \begin{cases} 1, \text{ If distributor } j \text{ is served by DC } h; \\ 0, \text{ Otherwise.} \end{cases}$$

$$x_{ij}^k = \begin{cases} 1, \text{ If the delivery vehicle } k \text{ travels from node } i \text{ to node } j; \\ 0, \text{ Otherwise.} \end{cases}$$

$$y_i^k = \begin{cases} 1, \text{ If delivery vehicle } k \text{ serves node } i; \\ 0, \text{ Otherwise.} \end{cases}$$

$$S = \begin{cases} 1, \text{ If there is the shortage of stock and insufficient capacity;} \\ 0, \text{ Otherwise.} \end{cases}$$

5. Formulation of Bi-Objective LIRP with Stochastic Demand

5.1. Objective Function Design

1. Fixed Construction Cost
The LIRP model proposed by [28] stated that the establishment cost of a distribution center (DC) depends on whether the DC, h, is open. Therefore, the fixed construction cost of the DC location is given as follows:

$$G = \sum_{h \in H} g_h y_h. \tag{1}$$

2. Transportation Cost
The transportation cost in this model is mainly composed of two parts. The first part is the fixed cost of delivery vehicles, which is the cost of dispatching vehicles, as mentioned by [28]. The fixed cost, denoted as C_1, includes the fixed cost of vehicles, wages of delivery personnel, and other vehicle-related costs. This cost is constant and does not depend on the vehicle weight or driving distance.

The second part of the transportation cost, as discussed in previous literature such as [9,10,28,29], is traditionally measured by the unit distance cost multiplied by the distance traveled. Generally, the farther a vehicle travels, the greater its carrying

capacity and the higher the transportation cost [29]. However, unit distance cost is not as straightforward to determine as the fuel consumption on the road. Additionally, both refrigeration and the operation of cold chain vehicles are primarily powered by fuel oil. Thus, following the approach in the cold chain vehicle routing problem (VRP) literature [1], the transportation cost incorporates the energy consumption cost of the delivery vehicle, denoted as FC_{ij}.

The fuel consumption per unit distance, ρ, at customer node (i, j) shows a linear correlation with the vehicle load Q_X [30]. This relationship can be mathematically expressed as follows:

$$\rho(Q_X) = a(Q_0 + Q_X) + b. \tag{2}$$

Considering the dead weight (Q_0) and maximum load capacity (Q_k) of the distribution vehicle, it is imperative to determine both the fuel consumption per unit distance in unloaded conditions (ρ_0) and under full-load circumstances:

$$\rho_0 = aQ_0 + b, \tag{3}$$

$$\rho^* = a(Q_0 + Q_k) + b, \tag{4}$$

which obtain,

$$a = \frac{\rho^* - \rho_0}{Q_k}. \tag{5}$$

In short, the determination of fuel efficiency per unit distance for a vehicle, denoted as $\rho(Q_X)$, can be formulated.

$$\rho(Q_X) = \rho_0 + \frac{\rho^* - \rho_0}{Q_k} Q_X. \tag{6}$$

The combustion of fuel leads to the production of carbon emissions, and the consumption of fuel by the distribution vehicle is influenced not only by its distance but also by its cargo capacity. The total carbon emission, Q_c, in the distribution of the node (i, j) section can be obtained as follows:

$$Q_c = \rho(Q_{ij})\omega d_{ij} = (\rho_0 + \frac{\rho^* - \rho_0}{Q_k} Q_{ij})\omega d_{ij}. \tag{7}$$

By substituting Equation (6), the energy cost FC_{ij} is as follows:

$$FC_{ij} = \rho(Q_{ij}) p_2 d_{ij} = (\rho_0 + \frac{\rho^* - \rho_0}{Q_k} Q_{ij}) p_2 d_{ij}. \tag{8}$$

Therefore, the transportation cost T_c of delivery vehicles can be expressed as follows:

$$T_c = C_1 \sum_{i \in N \cup H \cup J} x_{0i}^k + \sum_{i \in N \cup H \cup J} \sum_{j \in H \cup J} \sum_{k \in K} FC_{ij} x_{ij}^k. \tag{9}$$

3. Inventory Cost

Ref. [31] used the normal distribution of demand in the LIRP model and considered the cost of the safety stock. In addition, the inventory cost considered in the LIRP model of [32] also included the expected cost of both the working stock and the safety stock.

The ordering cost is as follows:

$$p \sum_{h \in H} \sum_{j \in J} \frac{\mu_j z_{hj}}{x_h}.$$

The expected inventory cost is as follows:

$$\sum_{h \in H} i_h \frac{x_h}{2}.$$

Assume that the demand at each distributor is independent of each other and follows a standard normal distribution. Thus, the safe stock point where the demand at the DC matches the order lead time is given by the following:

$$\sum_{j \in J} L\sigma_j^2 z_{hj}.$$

The order point of the DC is as follows:

$$z_\alpha \sqrt{\sum_{j \in J} L\sigma_j^2 z_{hj}}.$$

Therefore, the inventory cost considered in this paper considers the order cost, expected inventory cost, and safety inventory cost:

$$W_c = p \sum_{h \in H} \sum_{j \in J} \frac{\mu_j z_{hj}}{x_h} + \sum_{h \in H} i_h \left(\frac{x_h}{2} + z_\alpha \sqrt{\sum_{j \in J} L\sigma_j^2}\right). \tag{10}$$

4. TW Penalty Cost Function

According to [29], the time-considering penalty cost is calculated similarly to the method described by [10]. The time window (TW) penalty cost function was derived based on the principle that during the delivery process, customers typically have specific time constraints for receiving frozen or refrigerated food. Failure to meet these agreed-upon delivery times can result in reduced CS, increased vehicle energy consumption, potential loss of goods, and subsequent penalty costs.

In this paper, the penalty cost associated with TWs is taken into consideration. It is assumed that each customer has an expected delivery time window, denoted as $[ET_j', LT_j']$. To avoid incurring penalty costs, an acceptable delivery time window, denoted as $[eT_j, lT_j]$, is determined. This acceptable window ensures that the delivery is made within the customer's specified time frame while minimizing the risk of penalties. If the delivery vehicle arrives at ET_j' ahead of time and waits, the idle cost of delivery personnel and the delivery vehicle will be generated. Arriving later than LT_j' will incur a penalty.

In general, when the delivery vehicle arrives earlier than ET_j', the impact on the customer is minimal. However, if the delivery vehicle arrives later than the expected time LT_j', the customer may refuse to accept the delivery, resulting in a significant penalty cost. The relationship between the penalty cost and the delivery time is illustrated in Figure 2. This graph highlights the increasing penalty cost, as the delivery time exceeds the agreed-upon TW.

Suppose the abscissa coordinate represents the delivery time t_j, and the ordinate coordinate $P_j(t_j)$ represents the penalty cost. Let α_1 and α_2 $(\alpha_1, \alpha_2 \geq 0)$ be the penalty coefficients for the intervals $[eT_j, ET_j']$ and $[LT_j', lT_j]$, respectively, where $\alpha_2 \geq \alpha_1$. M represents an infinite penalty cost, indicating that when the delivery time exceeds the acceptable time window for the customer, the customer will reject the goods, incurring an infinite penalty cost.

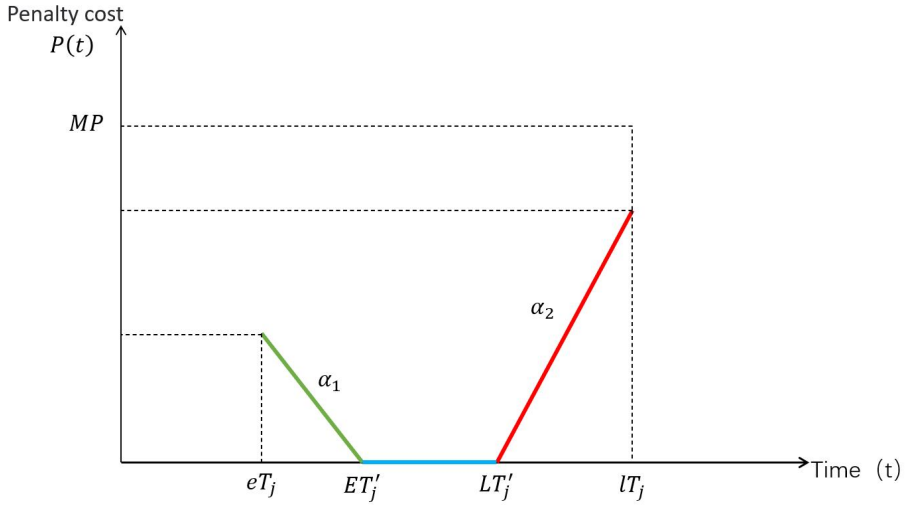

Figure 2. Diagram of TW penalty cost function.

From Figure 2, it can be observed that when the goods are delivered within the $[ET'_j, LT'_j]$, the penalty cost is zero. If the delivery falls outside the acceptable range of $[eT_j, lT_j]$, an infinite penalty cost is incurred. Furthermore, when the delivery occurs within the time frame of $[eT_j, ET'_j]$, the penalty cost gradually decreases as the deviation from the customer's expected earliest delivery time diminishes. If the goods are delivered within $[LT'_j, lT_j]$, the penalty cost will increase with a greater deviation from the latest expected delivery time by the customer. Therefore, the TW penalty function can be expressed as follows:

$$P_j(t_j) = \begin{cases} ET'_j - t_j, & eT_j \le t_j < ET'_j \\ 0, & ET'_j \le t_j < LT'_j \\ t_j - LT'_j, & LT'_j \le t_j \le lT_j \end{cases}. \tag{11}$$

$$P_c = \alpha_1 \sum_{j \in J} \max\left\{ET'_j - t_j, 0\right\} + \alpha_2 \sum_{j \in J} \max\left\{t_j - LT'_j, 0\right\}. \tag{12}$$

This penalty function effectively models the cost implications of deviating from the customer's preferred delivery time window, thus helping in optimizing the delivery schedules to enhance customer satisfaction and minimize penalties.

5. Carbon Cost Function

Carbon emissions are generated by burning fuel. The fuel consumption of delivery vehicles is influenced by two factors: the driving distance and the cargo weight. Both of these factors play a role in determining the overall fuel efficiency of the delivery vehicles. Therefore, the total carbon emission Q_c in the distribution node (i, j) section can be obtained as follows:

$$Q_c = \rho(Q_{ij})\omega d_{ij} = (\rho_0 + \frac{\rho^* - \rho_0}{Q_k} Q_{ij})\omega d_{ij}. \tag{13}$$

The carbon trading mechanism, initially introduced by [33] in the context of VRP, has been further explored by [34] to understand its implications on logistics distribution. These studies underscore that if actual carbon emissions fall below the allocated quota, companies can profit by selling surplus emission rights. Conversely, surpassing the carbon cap mandates purchasing additional allowances to offset excess emissions. Consequently, the carbon emission cost primarily signifies the environmental expense attributable to vehicle emissions during delivery operations. Therefore, the carbon emission cost C_c for the delivery at customer node (i, j) is expressed as follows:

$$C_c = C_e\left(\sum_{i\in N\cup H\cup J}\sum_{j\in J}\rho(Q_{ij})\omega d_{ij} - C^{CAP}\right). \tag{14}$$

6. Customer Satisfaction Function
Based on the approach proposed by [15], customers expect their delivery within the time window $[ET'_j, LT'_j]$, with an acceptable delivery window $[eT_j, lT_j]$ to avoid penalty costs. However, due to epidemic prevention measures, DCs often face shortages and insufficient capacity. Consequently, most customers tolerate delivery times that are slightly earlier or later than their preferred time window. Assume that $S = 1$ denotes goods shortages and capacity insufficiency. In this special case, the delivery time window acceptable to customers is $[eT_j, lT_j]$, where the abscissa represents the delivery time t_j, and the ordinate $V(t_j)$ represents the CS function. Figure 3 depicts the relationship between the delivery time and CS.

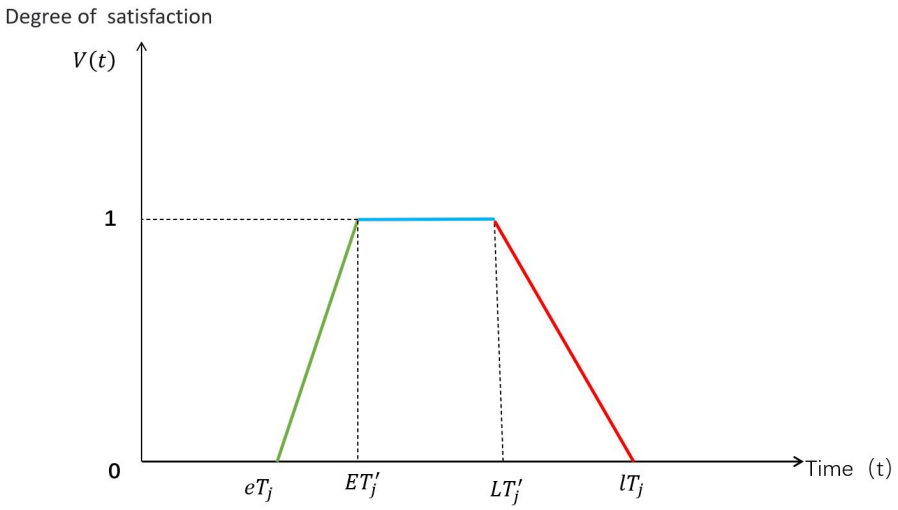

Figure 3. Diagram of customer satisfaction function.

As depicted in Figure 3, in special circumstances, when goods are delivered within the $[ET'_j, LT'_j]$ timeframe and the customer is served accordingly, his/her satisfaction is rated as 1. If deliveries occur outside the $[eT_j, lT_j]$ window, CS drops to 0. Within the $[eT_j, ET'_j]$ interval, CS increases as the delivery time approaches the earliest expected delivery time. Conversely, within $[LT'_j, lT_j]$, CS decreases as the delivery time extends beyond the latest expected delivery time. Therefore, the CS function can be formulated as follows:

$$V(t_j) = \begin{cases} 0, & t_j < eT_j \\ \frac{t_j - ET'_j}{eT_j - ET'_j}, & eT_j \le t_j < ET'_j \\ 1, & ET'_j \le t_j \le LT'_j \\ \frac{LT'_j - t_j}{LT'_j - lT_j}, & LT'_j < t_j < lT_j \\ 0, & lT_j < t_j \end{cases}. \tag{15}$$

5.2. The Bi-Objective LIRP Model

The proposed bi-objective optimization model aims to simultaneously minimize the TSCC and maximize CS. The MINLP formulation for this model is structured as follows, incorporating the stated objectives:

$$\min f_1 = \lambda_1 G + \lambda_2 T_c + \lambda_3 W_c + \lambda_4 P_c + \lambda_5 C_c \tag{16}$$

$$\max f_2 = \sum_{h\in H}\sum_{j\in J} V(t_j) x_{hj} / \sum_{h\in H}\sum_{j\in J} x_{hj} \tag{17}$$

subject to,

$$x_h + z_\alpha \sqrt{\sum_{j\in J} L\sigma_j^2 z_{hj}} \le C_h y_h, \qquad \forall h \in H \tag{18}$$

$$\sum_{i\in N\cup H\cup J} q_i y_i^k \le Q_k, \qquad \forall k \in K \tag{19}$$

$$\sum_{i\in H}\sum_{j\in J} d_{ij} x_{ij}^k \le D \tag{20}$$

$$\sum_{i\in H}\sum_{j\in J}\sum_{k\in K} x_{ij}^k \le K, \tag{21}$$

$$\sum_{j\in J} x_{ij}^k = y_j^k, \qquad \forall i \in H, k \in K \tag{22}$$

$$\sum_{k\in K} y_i^k = 1, \qquad \forall i \in N \cup H \tag{23}$$

$$ET_j \le t_j \le LT_j, \qquad \forall j \in J \tag{24}$$

$$eT_j \le t_j \le lT_j, \qquad \forall j \in J, S = 1 \tag{25}$$

$$x_h \ge \sum_{j\in J} x_{hj} z_{hj}, \qquad \forall h \in H \tag{26}$$

$$x_{hj} z_{hj} \ge u_j, \qquad \forall h \in H \tag{27}$$

$$\sum_{h\in H} z_{hj} = 1, \qquad \forall j \in J \tag{28}$$

$$z_{hj} \le y_h, \forall h \in H, j \in J \tag{29}$$

$$y_h = \begin{cases} 1, \text{ If establish the DC } h; \\ 0, \text{ Otherwise.} \end{cases} \tag{30}$$

$$z_{hj} = \begin{cases} 1, \text{ If distributor } j \text{ is served by DC } h; \\ 0, \text{ Otherwise.} \end{cases} \tag{31}$$

$$x_{ij}^k = \begin{cases} 1, \text{ If the delivery vehicle } k \text{ travels from node } i \text{ to node } j; \\ 0, \text{ Otherwise.} \end{cases} \tag{32}$$

$$y_i^k = \begin{cases} 1, \text{ If delivery vehicle } k \text{ serves node } i; \\ 0, \text{ Otherwise.} \end{cases} \tag{33}$$

In the above model, the objective function (16) represents the minimization of TSCC, encompassing fixed construction costs, transportation costs, inventory costs, TW penalty costs, and carbon emission costs. Objective function (17), on the other hand, seeks to maximize CS. These dual objectives are subject to the following constraints: Constraint (18) ensures the capacity limitations of DCs, while Constraint (19) guarantees the vehicle's service capacity. Constraint (20) imposes mileage restrictions on each vehicle, and Constraint (21) limits the number of distributed vehicles. Constraint (22) ensures that each vehicle's distribution route begins and ends at the same DC, and Constraint (23) mandates that each customer is serviced by exactly one vehicle. Constraint (24) manages the STW constraint, while Constraint (25) addresses the STW constraints in scenarios involving stock shortages and insufficient transportation. Constraint (26) ensures that the volume ordered by the DC exceeds the volume shipped from the DC to the customer, and Constraint (27)

ensures that the transportation volume from the DC to the customer meets the customer's demand. Constraint (28) guarantees that each customer is served by only one DC, and Constraint (29) ensures that the selected DC has an assigned delivery route. Constraints (30) to (33) define binary decision variables, ensuring their values are either 0 or 1.

6. Development of IMNSGA-II

The carbon trading and CS logistics of the LIRP represent a bi-objective optimization challenge that falls into the category of NP-hard problems. The NSGA-II is a widely used algorithm that is particularly effective in solving multi-objective optimization problems. Utilizing NSGA-II enables better decision-making based on the priorities set by decision-makers, as it provides a set of Pareto optimal solutions, offering a range of choices [19].

Building on the findings of the proposed bi-objective LIRP model and the demonstrated performance of NSGA-II in previous studies, this paper introduces an improved NSGA-II (IMNSGA-II) to solve the above-mentioned model while preserving the algorithm's superior capabilities. The IMNSGA-II incorporates three main enhancements over the standard NSGA-II: (i) an improved non-dominated sorting algorithm, (ii) the introduction of crowding degree and a crowding comparison operator, and (iii) the adoption of an elite strategy. These improvements make the algorithm particularly well-suited for addressing the multi-objective optimization model, yielding a set of Pareto optimal solutions. This provides decision-makers with superior choices, thereby enhancing the rationality and effectiveness of their decision-making process.

This section primarily discusses the coding scheme, population initialization method, selection process, crossover and mutation strategies, and population merging and optimization techniques used in the development of the algorithm.

6.1. Coding Scheme

In a genetic algorithm (GA), each individual represents a potential solution to the problem and must include the following information:

1. The vehicle distribution routing scheme and volume scheme for the first cycle.
2. The vehicle distribution for the second cycle.
3. The vehicle distribution routing scheme and volume scheme for the nth cycle, iterated through successive generations until the nth generation.

Based on the above principles, the following basic coding rules can be obtained: (i) each individual is divided into several segments according to the number of cycles; and (ii) each segment is further divided into a routing segment and a distribution volume segment. The routing segment code directly represents the vehicle distribution sequence by the number of customers, while the distribution volume segment code directly indicates the distribution volume of the corresponding DC.

Figure 4 illustrates the distribution routing scheme and distribution volume scheme from the first cycle to the third cycle. In the figure, the blue markers represent the customers, the yellow markers denote the distribution volume corresponding to the customers, and the green markers signify the DC, with its corresponding distribution volume represented by inf.

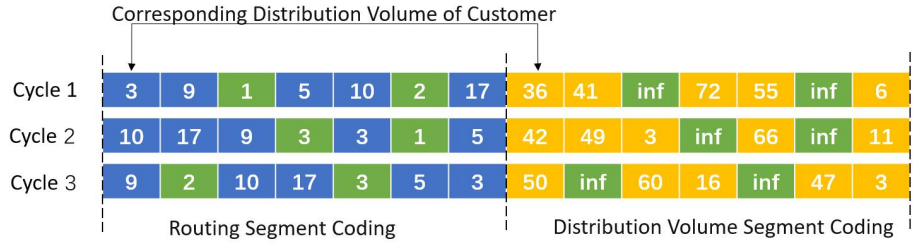

Figure 4. Schematic of the underlying coding rules.

Each DC is responsible for delivering to the various customer nodes on its left, while the final DC also delivers to the customer nodes on its right. Since the model assumes that vehicles will ultimately return to their departure DC, one gene position representing a DC is omitted at the beginning and end of each route segment code. This omitted gene position needs to be added when interpreting the code.

The specific meaning represented by the code can be elucidated as follows: (i) in the first cycle, vehicle *A* departs from DC 1, traverses through customers 3 and 9, and delivers quantities of 36 and 41 units, respectively, before returning to DC1; (ii) vehicle *B* departs from DC2, passes through customers 5 and 10, and delivers quantities of 72 and 55 units, respectively, before returning to DC2; (iii) vehicle *C* departs from DC2, passes through customer 6, and delivers a quantity of 17 units before returning to DC2. The delivery process in the first period is depicted in Figure 5.

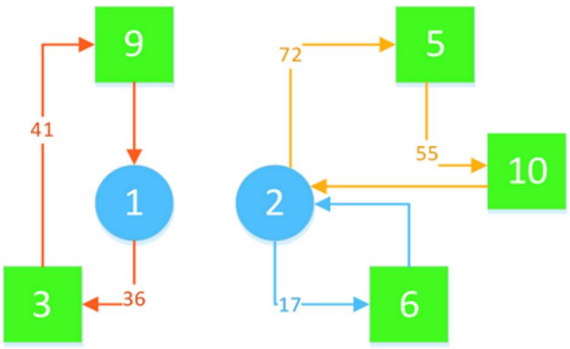

Figure 5. Schematic diagram of the first cycle delivery process.

In the preceding text, the three periods of an individual are displayed in three separate rows for clarity, but they actually belong to the same row in the coding scheme. Additionally, after encoding an individual, the gene positions for the calculated objective function values are appended at the end. This facilitates subsequent non-dominated sorting and crowding distance calculation.

In Figure 6, the complete individual code is shown with two terminal red markers. These loci record the values of objective functions 1 and 2, as calculated by the DC based on the solution provided by the individual.

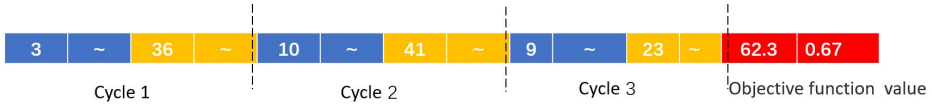

Figure 6. Schematic of the complete individual code.

6.2. Population Initialization

To streamline algorithm design and enhance computational efficiency, this model employs a customer division approach based on the proximity to the nearest DC before addressing the optimization problem. This transformation converts the original problem involving multiple DCs and customers into several instances of a single DC serving multiple customer sets. Post-division, each DC assumes responsibility for a distinct customer subset, with minimal interdependence. This segregation enables separate computation of TSCC and CS, significantly reducing model complexity and computational burden.

The division process begins by calculating the distances between each customer and all available DCs. Subsequently, each customer is assigned to the nearest DC based on these computed distances. This assignment forms customer sets that are specific to each DC, facilitating isolated calculations of TSCC and CS for each subset. To illustrate this process, consider DC1 as an example. The process unfolds in the following four steps:

Step 1: After assigning customers to DC1, obtain the set of customers for which DC1 is responsible for.

Step 2: DC1 randomly selects a number n of vehicles within the specified range (1, $number\_vehicle_{max}$), representing all vehicles dispatched by DC1 in the current cycle's distribution process. According to the coding scheme described earlier, the DC number's position in the individual denotes the boundary point for coding segments of different vehicles' distribution paths. Therefore, this process involves inserting $(n-1)$ DC numbers into the sequence of customer routes obtained in Step 1.

The value $(n-1)$ accounts for the default exclusion of the DC number at the beginning and end of the path segment coding, signifying that at least one vehicle has completed the distribution task and has been included in the selected n, necessitating a subtraction of 1. When inserting the DC numbers, selecting both the number of insertions and their positions randomly represents different distribution vehicle and route selection schemes. To maintain uniformity in the length of the coding array generated by each individual in this step, additional gene bits are appended to the end of this section after adding the number of distribution centers. Specifically, the number of additional bits will be $(number\_vehicle_{max} - n - 1)$, and the redundant gene bits will be filled with inf.

Step 3: The length of the distribution volume segment code obtained in Step 2 is used to generate a random array of the same length within the range of $(1, carriage_{max})$, where $carriage_{max}$ represents the maximum capacity limit of customer inventories. The distribution quantity code obtained in Step 2, where DC is coded as 1 and the filled redundant gene bits as inf, is then used to replace the corresponding gene bits of the random array with inf. Consequently, the distribution volume segment code for each customer's delivery vehicle is then determined.

Step 4: Combine the codes from Step 2 and Step 3 to generate a set of distribution paths and quantity plans for DC1 in the first cycle. The objective function values of this plan are computed based on the TSCC and CS provided by the model, with the last two gene bits sequentially filled.

Figure 7 notates the specific process used to obtain individual coding in the first cycle. Then, the process is iterated according to the specified number of cycles (e.g., each individual contains T cycles) and the nested population size (e.g., population size of 200). This will result in a population matrix with x columns. Each row represents an individual with T period segment codes and two function values.

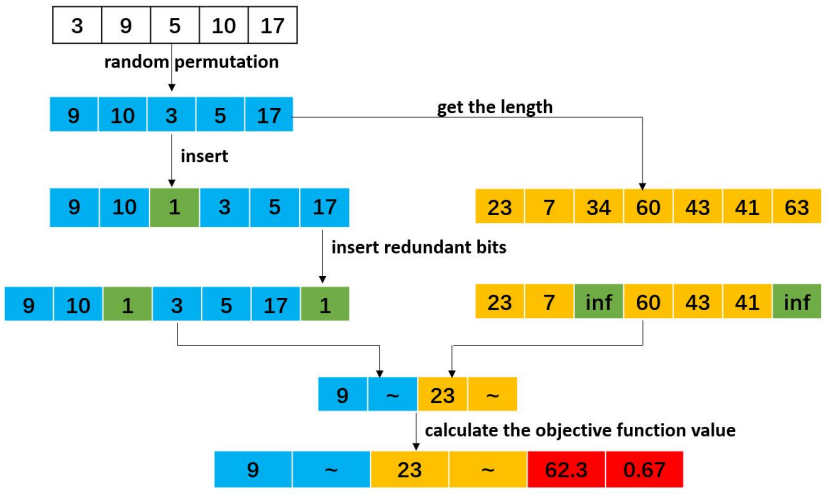

Figure 7. Initialization of a single individual.

The value of x is given by the following:

$$x = (length_{routing} + length_{shippingquantity}) \times T. \tag{34}$$

6.3. Selection, Crossover, and Mutation Strategies

6.3.1. Selection Strategy

This study employs the tournament selection method, which involves selecting the best individuals from the current population and adding them to the mating pool [35]. This step mimics natural selection, where individuals compete within a population for the opportunity to reproduce. Only individuals with genetic advantages are chosen to participate in the crossover and mutation processes of the next generation, ensuring continuous evolution of the population toward better solutions. The algorithm consistently explores improved solutions based on the current population. The competitive selection process is illustrated in Figure 8.

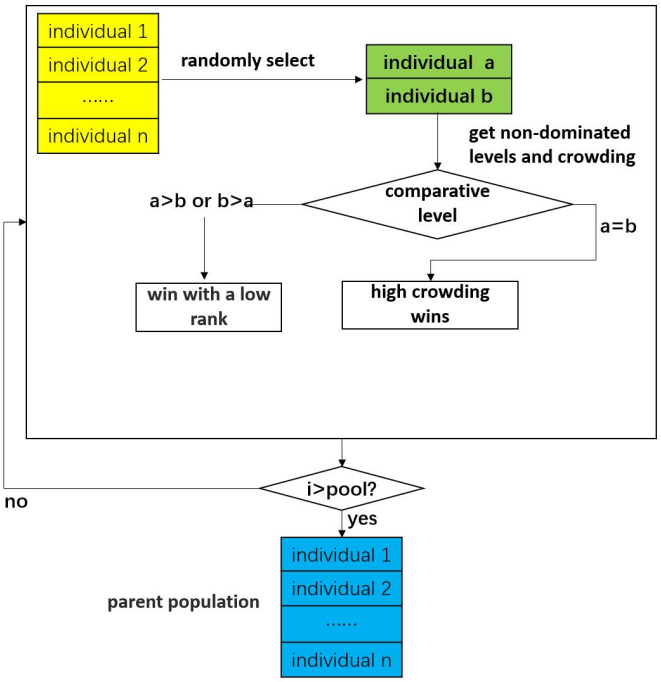

Figure 8. Schematic diagram of the tournament selection method.

The tournament selection process is outlined in the following steps: (i) Determine the number of individuals competing and the size of the mating pool. (ii) Compare non-dominated ranks to select direct winners; in cases of ties in rank, compare their crowding degree to determine superiority. By adhering to these competitive principles, the top individuals are chosen to join the parent population. (iii) Repeat this selection process iteratively until the parent population reaches the specified size of the mating pool.

6.3.2. Crossover Strategy

Crossover is a fundamental genetic operation where offspring chromosomes are derived from parent chromosomes during mating, thereby producing the next generation. In the context of the models developed in this study, crossover aims to optimize distribution routes, quantities, and vehicle assignments. To effectively exploit the solution space across these dimensions, this paper proposes several crossover methods that are tailored to the coding schemes employed.

The utilization of diverse crossover strategies serves to expand the exploitation of solution spaces, mitigate the risk of converging to local optima, and enhance the likelihood of identifying optimal solutions across broader ranges. In the proposed algorithm, different crossover methods are assigned varying probabilities while maintaining a consistent overall probability for the entire crossover process.

(a) Coding crossover of single-cycle failure segment

This method involves randomly selecting two parent individuals in each iteration and identifying a random crossover point in their routing or distribution quantity codes. To ensure compatibility, both code lengths must be equal during initialization, facilitating a straightforward crossover between the two sets of codes. The crossover point determines the boundary segment, where all non-DC location codes before this point are exchanged between the routing or distribution segments. Figure 9 illustrates this process.

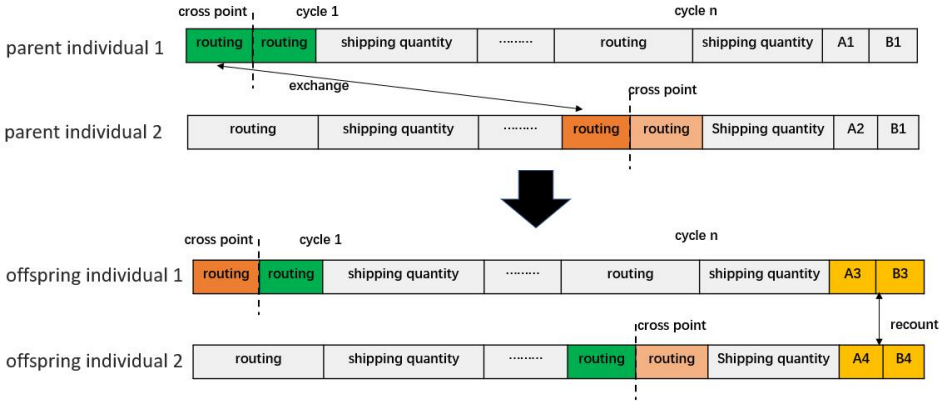

Figure 9. Coding crossover of single-cycle failure segment.

Based on the description above, this method primarily involves exchanging a portion of the routing segment code or distribution volume segment code between two individuals while keeping the location and number of DC genes unchanged, thereby preserving the vehicle selection. Importantly, the specific segments exchanged and the routing or distribution volume are randomly determined, with an equal probability of 0.5 for either occurrence.

(b) Dual-individual multi-period segment-wide crossover

In this method, n cycles are randomly selected among all individual cycles, then the path segment codes or distribution segment codes of non-DC of the two parent individuals on these n cycles are exchanged directly by taking the boundary between path segment codes and distribution segment codes as the crossing point.

Assuming a total of n cycles, the 1st and nth cycles are selected for crossover, as shown in Figure 10:

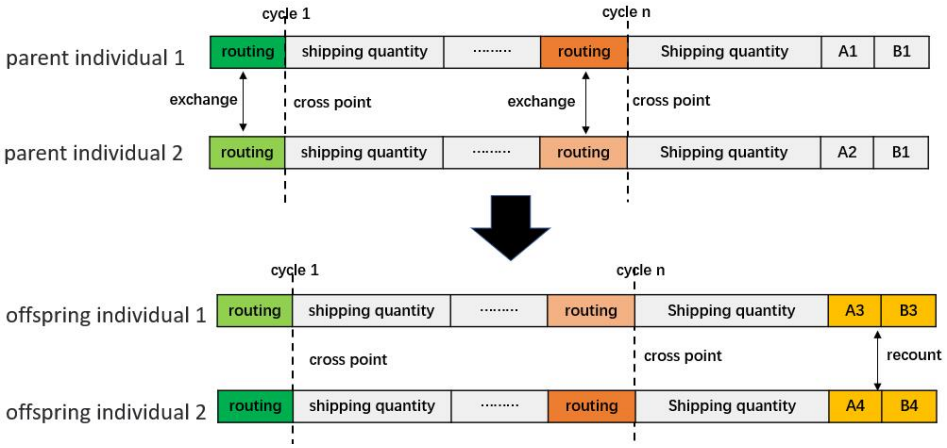

Figure 10. Dual-individual multi-period segment-wide crossover.

(c) Single-individual cycle crossover

In this method, a single-parent individual conducts self-reorganization through a specific process. Initially, the individual randomly reorders its cycle numbers to create a new sequence of cycles. Subsequently, it fills in the path segment and distribution segment codes for each cycle according to this new order.

As observed from the description, this method essentially involves reordering the cycles of the individual to generate offspring without altering the arrangement of the path segment and distribution segment codes. In the context of the model, this adjustment affects only the inventory cost and penalty cost, while the transportation cost remains unchanged.

It is assumed that there are 3 cycles in total, and the order after random rearrangement is $(3, 1, 2)$; that is, cycles 1 and 3 are exchanged. A schematic diagram of the crossing is shown in Figure 11.

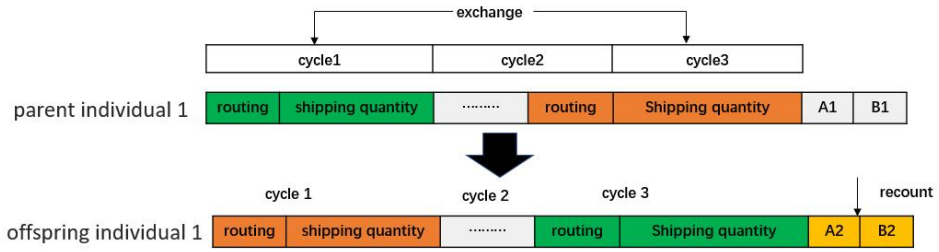

Figure 11. Single-individual cycle crossover.

6.3.3. Mutation Strategy

Mutation is an essential component of GAs that is crucial for escaping local optima and exploring new areas of the solution space. As previously discussed, the model in this study requires exploration in three key areas to generate offspring: adjusting delivery routes, modifying shipment quantities, and altering vehicle selections. While crossover methods address changes in delivery routes and quantities effectively, they do not inherently handle changes in vehicle selection, which are more suited to mutation operations.

Aligned with the encoding approach used, this subsection introduces various mutation techniques. These methods serve a role similar to crossover by promoting diversity and exploration in the solution space. To ensure effective exploration, this model employs a relatively high mutation probability (0.5–0.9) for individual genes, specifically targeting vehicle selection changes. This probability dictates the likelihood that each gene within an individual will undergo mutation sequentially during the algorithm's execution.

(a) Single-cycle gene exchange mutation

This mutation process involves interchanging the positions of two genes within the coding of a routing segment or shipping quantity segment within a single cycle for an individual. This interchanging is restricted to occur within the same cycle, because exchanging gene positions between different cycles could lead to repeated customers within the routing segments, violating the constraint that each customer can only be visited once per cycle in the model. However, interchanging gene positions within the same cycle does not pose this issue.

Interchange can only occur within a single coding segment; genes in routing segments and shipping quantity segments are not subject to exchange. These constraints limit the diversity of mutations, which justifies implementing a higher mutation probability. The specific process is illustrated in Figure 12.

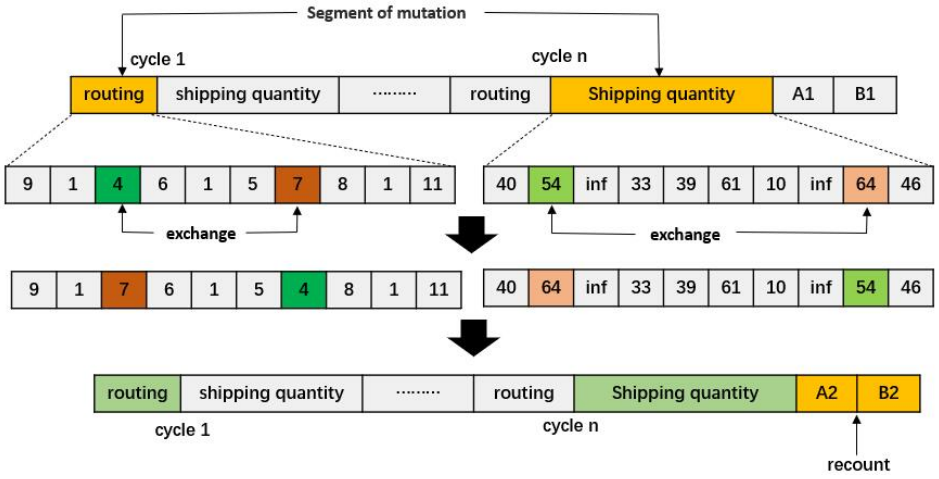

Figure 12. Single-cycle gene exchange mutation.

As described above, this mutation method modifies the coding of the routing segment and distribution volume segment. Therefore, it is not suitable for the DC location. When evaluating whether this mutation method should be applied to each gene, it is essential to skip the DC location.

(b) Vehicle selection mutation

This mutation method focuses on modifying the vehicle selection in each cycle delivery plan provided at initialization, specifically targeting the genes of the DC within the chromosome. This process can be broadly categorized into two directions: adding and reducing a vehicle. Adding a vehicle involves redistributing part of the workload from an existing vehicle to accommodate the newly added one, while reducing a vehicle entails transferring delivery responsibilities to another vehicle. It is important to note that for these mutations to occur, there must be available gene positions, which can manifest in three specific cases.

- Case 1: If only one vehicle is assigned to a specific cycle code, meaning there is no DC gene position in the routing segment and shipping quantity segment code, then this segment code is not eligible for a mutation to reduce the number of vehicles. However, it can still undergo a mutation to increase the number of vehicles. When this mutation occurs at cycle a, its schematic diagram is shown in Figure 13.

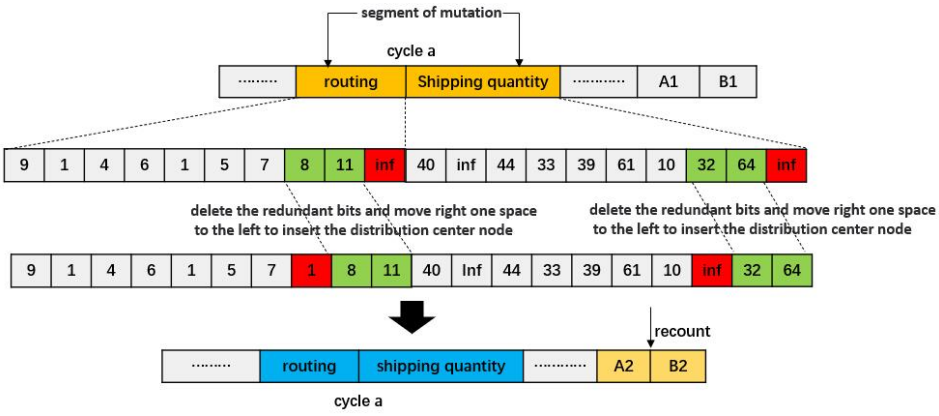

Figure 13. Increase vehicle mutation.

- Case 2: If a specific cycle code has reached its maximum vehicle capacity, this code cannot experience an increase in the number of vehicle mutations; only a decrease in the number of vehicle mutations can occur. When this mutation occurs at cycle b, its schematic diagram is depicted in Figure 14.

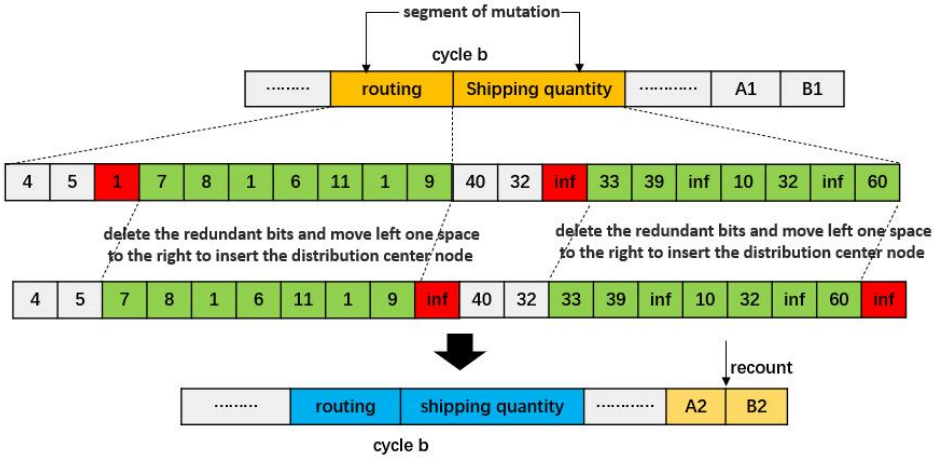

Figure 14. Decrease vehicle mutation.

- Case 3: If the number of vehicles called in a specific cycle exceeds 1 but falls below the upper limit, then this segment code is eligible for a random vehicle selection mutation.

The above restrictions are in place because the chromosomes of existing individuals have already determined the coding length during initialization. If a mutation were to exceed this length limit, it would disrupt the entire population's chromosomes. Only in cases where there are non-limit redundant bits at the end of the routing segment encoding (i.e., Case 3 mentioned above) can an arbitrary vehicle selection mutations occur. By adding a vehicle, it shifts the subsequent code backward and removes a redundant bit. On the other hand, by removing a vehicle, it shifts subsequent code forward and adds a redundant bit. Adjustments to the coding of the shipping quantity segment should be aligned with the subsequent routing segment coding.

6.4. Population Merging and Optimization

After the selection, crossover, and mutation processes, a parent population and an offspring population generated by the mating of the parent population are selected. However, in practice, the problems are usually highly complex. This complexity means that the excellence of an individual is derived from the collective quality of the genes on its chromosome, contributing to an overall better solution. It cannot be assumed that all chromosomes of high-performing individuals are uniformly excellent or that their superiority is due to a specific segment of their chromosomes. Consequently, while crossover and mutation between two high-quality individuals might have a higher probability of producing better offspring, it does not guarantee it.

To address this issue, the IMNSGA-II algorithm incorporates an elite selection strategy. This involves merging the offspring population with the parent population to form an intermediate population that is twice as large as the original. This combined population then undergoes non-dominated sorting and crowding distance calculations. Finally, the top-ranked non-dominated individuals are selected to fill the set population size from the top down, ensuring that the best individuals are retained for the next generation after reproduction [36]. The detailed process is illustrated in Figure 15.

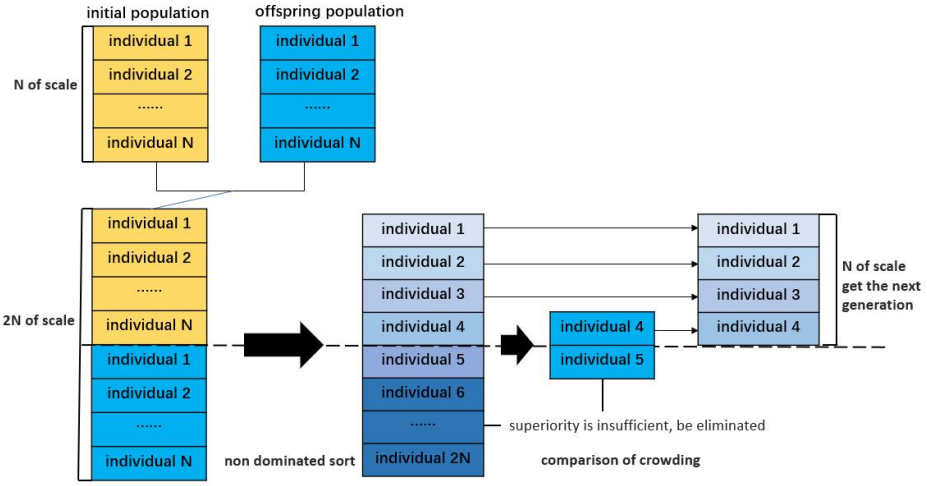

Figure 15. Population merging and optimization.

6.5. Dynamic Crowding Distance

In the algorithm, the advantages and disadvantages of individuals are judged by the crowding operator. However, the resulting crowding operator may deviate from the actual density of individuals, and if so, high-density individuals may be retained in the next generation, leading to local optima [22]. To improve the uniformity of individual distribution, diversify population distribution, and enhance the global search ability, this study uses the dynamic crowding distance n_{di}. Its algorithm is as follows:

Step 1: Let the parameter $n_d = 0, n \in 1 \ldots N$.

Step 2: For each objective function f_m:

① Individuals of this level are ranked according to the objective function. f_m^{max} is the maximum value of the individual objective function f_m, and f_m^{min} is the minimum value of the individual objective function f_m;

② Crowding degree of the two boundaries after sorting, 1_d and N_d are set to ∞;

③ The dynamic crowding distance of individual i is calculated as follows:

$$n_{di} = n_d / lg(1/V_i), \tag{35}$$

where

$$V_i = \frac{1}{M} \sum_{m=1}^{M} (\mid f_m(i+1) - f_m(i-1) \mid - n_d)^2,$$

$$n_d = \frac{(f_m(i+1) - f_m(i-1))}{f_m^{max} - f_m^{min}}.$$

$f_m(i+1)$ is the last bit of the objective function value after the individual is sorted, and M is the number of objective functions.

6.6. Steps of IMNSGA-II Algorithm

The algorithm starts by generating a random initial population of size N. The first-generation offspring population is obtained through non-dominated sorting, followed by selection, crossover, and mutation operations, as previously described.

From the second generation onwards, the parent and offspring populations are merged to perform fast non-dominated ranking. At the same time, the crowding degree of individuals in each non-dominated layer is calculated using Equation (35). Based on the non-dominance relationship and crowding degree, suitable individuals are selected to form the new parent population. This is followed by generating a new offspring population through the fundamental operations of the GA. This cycle repeats until the algorithm's termination conditions are met. The algorithm's flowchart is depicted in Figure 16.

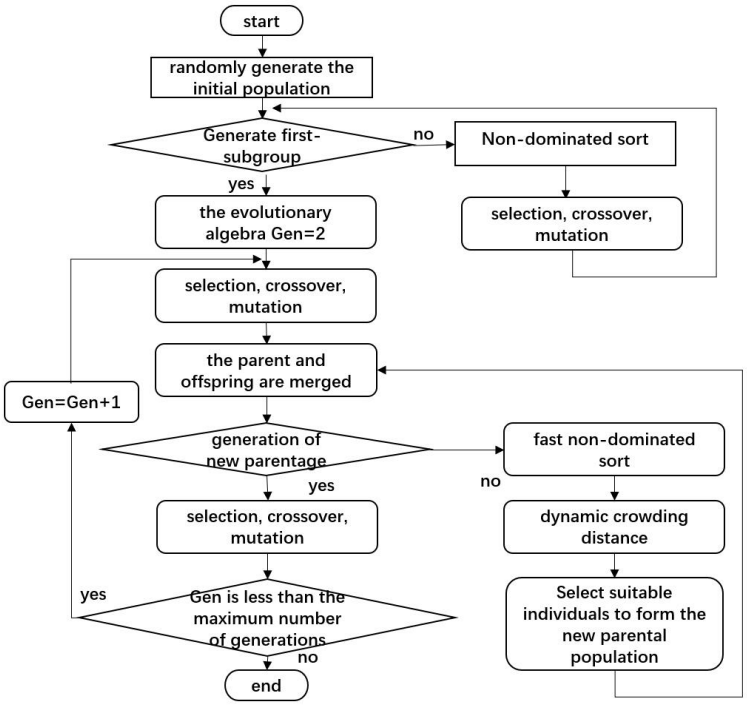

Figure 16. Flow chart of the IMNSGA-II algorithm.

6.7. The Entropy-TOPSIS Method: A Refined Approach for Decision Making

In a multi-objective Pareto-optimal solution set, objectively evaluating each solution's advantages and disadvantages can be challenging. In practice, decision-makers often rely on their experience or operational strategies to choose a final solution. Many research papers, such as [9,19,37], provide comparative conclusions or present various algorithms without providing an objective decision-making reference. In this study, the entropy–TOPSIS method by [38] is utilized to determine the optimal reference scheme by calculating the proximity index between each solution in the Pareto solution set and the optimal level to help decision-makers choose a Pareto solution that balances multiple objectives as the optimal reference solution. The specific methods are as follows:

- Firstly, for the cost, which has the same dimension, the entropy method is employed to assign weights to the fixed cost, transportation cost, inventory cost, STW penalty cost, and carbon emission cost. These weights are denoted as λ_1, λ_2, λ_3, λ_4, and λ_5, respectively. The entropy weight method is an objective weight method proposed by ref. [39]. Its calculation steps are as follows:
  - Check for negative numbers in the input matrix. If any negative numbers are found, re-normalize them to the non-negative interval. Assuming T solutions in the Pareto-optimal solution set and 5 evaluation indicators (i.e., 5 cost data) for forward maximization, the resulting forward matrix is as follows:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{15} \\ x_{21} & x_{22} & \dots & x_{25} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t1} & x_{t2} & \dots & x_{t5} \end{bmatrix}. \tag{36}$$

Let us denote the normalized matrix as Z, and each entry in Z can be calculated as follows:

$$z_{ij} = x_{ij} / \sqrt{\sum_{i=1}^{t} x_{ij}^2}. \tag{37}$$

This will determine whether there are any negative numbers in the Z matrix. If there are, we need to use another normalization method for X. Let us denote the normalized matrix of X as $\widetilde{Z}$. This normalized equation is as follows:

$$\widetilde{z}_{ij} = \frac{x_{ij} - min\{x_{1j}, x_{2j}, \ldots, x_{tj}\}}{max\{x_{1j}, x_{2j}, \ldots, x_{tj}\} - min\{x_{1j}, x_{2j}, \ldots, x_{tj}\}}. \tag{38}$$

– To calculate the proportion of the ith sample under the jth index, treat it as the probability used in the calculation of relative entropy. Given that there are t objects to be evaluated and 5 evaluation indices, the resulting non-negative matrix after the previous processing step can be denoted as $\widetilde{Z}$:

$$\widetilde{Z} = \begin{bmatrix} \widetilde{z}_{11} & \widetilde{z}_{12} & \cdots & \widetilde{z}_{15} \\ \widetilde{z}_{21} & \widetilde{z}_{22} & \cdots & \widetilde{z}_{25} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{z}_{t1} & \widetilde{z}_{t2} & \cdots & \widetilde{z}_{t5} \end{bmatrix}. \tag{39}$$

Calculate the probability matrix P, where each element of P can be calculated by the following formula:

$$p_{ij} = \widetilde{z}_{ij} / \sum_{i=1}^{t} \widetilde{z}_{ij}. \tag{40}$$

– To calculate the information entropy of each indicator and obtain the entropy weight, calculate the information entropy for each index (jth index):

$$e_j = -\frac{1}{lnt} \sum_{i=1}^{t} p_{ij} ln(p_{ij}), (j = 1, 2, \ldots, 5). \tag{41}$$

– The value of information utility for the jth index can be calculated as $d_j = 1 - e_j$, so the greater the value of information utility is, the more information it corresponds to. The information utility value is normalized to obtain the entropy weight of each indicator:

$$w_j = d_j / \sum_{j=1}^{5} d_j, (j = 1, 2, \ldots, 5). \tag{42}$$

- Secondly, the weights are substituted into the objective function f_1, and the IMNSGA-II algorithm is reused to calculate the Pareto-optimal solution set. Taking two optimization objectives as evaluation indexes, the entropy weight method is introduced to calculate the objective weight of the two indexes, which can effectively reduce the influence of the less-reliable solutions at both ends of the Pareto-optimal solution set on the objective weight.
- Finally, the TOPSIS method is used to evaluate and rank the solutions of the Pareto-optimal solution set. Ref. [40] first proposed the TOPSIS method in 1981. It can be translated to the approximate ideal solution ranking method, often referred to as the good and bad solutions distance method. The TOPSIS method is a widely used comprehensive evaluation technique that effectively utilizes the information from the original data. Its result provides an accurate reflection of the differences between evaluation schemes. The specific steps (S1–S6) of the entropy–TOPSIS method used to select the optimal scheme by [40] are given below:

S1: The evaluation matrix is constructed and normalized. The Pareto-optimal solution set has t solutions, and the two objective functions are used to construct the evaluation index judgment matrix.

S2: The entropy weight w_j of the jth target is calculated by using the entropy weight Formula (42).
S3: The comprehensive weight β_j, of each objective function is determined.
S4: A weighted normalized matrix is constructed.
S5: The ideal solution and negative ideal solution are determined. The ideal solution v_j^+ and the negative ideal solution v_j^- are represented by the maximum and minimum values of each index in the weighted normalized matrix.
Distance d_t^+ from the ideal solution:

$$d_t^+ = \sqrt{\sum_{j=1}^{2}(v_{tj} - v_j^+)^2}. \tag{43}$$

Distance d_t^- from the ideal solution:

$$d_t^- = \sqrt{\sum_{j=1}^{2}(v_{tj} - v_j^-)^2}. \tag{44}$$

S6: The proximity index R_t of the tth solution to the optimal level in the Pareto-optimal solution set is calculated and sorted in descending order (the larger R_t is, the closer it is to the optimal level):

$$R_t = \frac{d_t^-}{d_t^- + d_t^+}. \tag{45}$$

7. Results and Discussion

7.1. Evaluation of IMNSGA-II Based on the Performance Metrics of the Multi-Objective Algorithm

The performance metrics of multi-objective algorithms can be categorized into three main types as follows:

(i) Convergence: This evaluates how closely the obtained solution set approximates the real Pareto front (PF). For example, the generational distance (GD) metric measures convergence, where a smaller GD value indicates better convergence of the approximate solution set S to the true PF [41];
(ii) Diversity: This assesses the distribution of the solution set across the entire PF, encompassing both spread and uniformity. Examples of diversity metrics include the hypervolume (HV) [42], maximum spread (MS) [31], coverage over Pareto front (CPF) [43], and pure diversity (PD) [44]. Higher values for these metrics suggest that the approximate solution set S covers a larger portion of the true PF, indicating better diversity.
(iii) Combined metrics: These consider both convergence and diversity. An example is the inverted generational distance (IGD) [45]. A smaller IGD value signifies that the solution set S has better convergence and diversity, allowing it to more effectively approximate the entire PF.

To assess the effectiveness of the proposed IMNSGA-II, the performance is benchmarked against other established algorithms, such as the Pareto envelope-based selection algorithm II (PESA-II) and NSGA-II in [46]. To ensure a fair comparison, several parameters are kept constant: (i) maximum iterations, $Gen = 300$, (ii) number of variables, $v = 500$, (iii) population size, $P = 50$, and (iv) crossover and mutation probabilities are fixed at 0.9 and 0.1, respectively. The experimental platform used for this comparative analysis is the evolutionary multi-objective optimization V3.5 (PlatEMO V3.5), as detailed in [47], which provides a standardized environment for evaluating algorithms based on multi-objective optimization problems (MOPs).

The specific MOP used for evaluation is the multi-objective traveling salesman problem (MOTSP) [48], which involves k objective TSP instances characterized by k distance matrices, each representing a distinct objective function. This setup allows for a comprehensive assessment of how well each algorithm balances competing objectives and produces Pareto-optimal solutions across different optimization criteria. These matrices are created as follows:

(i) Generate the TSP for objective 1 by assigning a uniform random number between 0 and 1 to each distinct pair of cities.
(ii) The TSP for objective $i + 1$ is generated using the following formula:

$$distance_{i+1}(a, b) = TSP_{cp} \cdot distance_i(a, b) + (1 - TSP_{cp}) \cdot rand(), \quad (46)$$

where TSP_{cp} is a correlation parameter introducing negative, zero, or positive inter-objective correlations when less than 0, equal to 0, or greater than 0, respectively.

In this study, a 500-city TSP is defined within PlatEMO V3.5 to generate a MOTSP using Equation (46). The PESA-II, NSGA-II, and IMNSGA-II algorithms are employed to solve and evaluate the MOTSP, comparing their performance across the following performance metrics: GD, HV, CPF, PD, MS, and IGD.

Figures 17–22 illustrate the performance trajectories of the three algorithms on the six performance metrics mentioned above as the number of function evaluations increases. Table 2 presents the average performance metric values for each algorithm on the test problem. Additionally, Table 2 also includes the computational time (CT) required by each algorithm, providing insights into their efficiency alongside their optimization effectiveness.

Table 2. Comparison of average results based on the performance metrics.

Evaluation Metric	NSGA-II	PESA-II	IMNSGA-II
GD	83.121	101.490	**79.900**
HV	0.51372	**0.54788**	0.51238
CPF	5.3658	5.5130	**5.6845**
PD	35615	16616	**38328**
IGD	466.61	493.65	**464.63**
MS	0.98967	**0.99488**	0.98726
CT	10.904	**10.465**	10.496

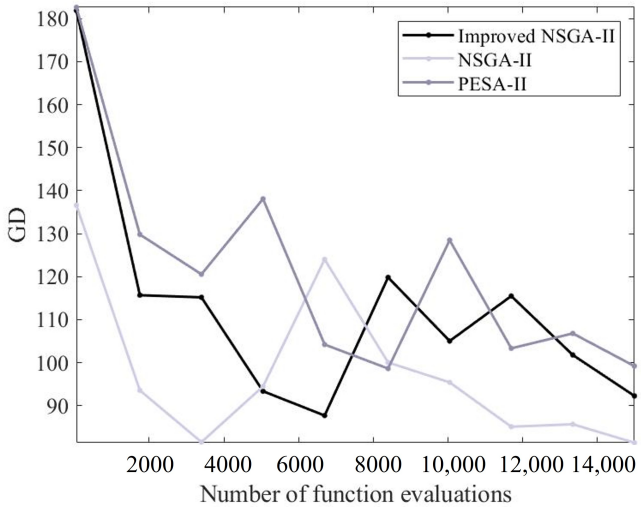

Figure 17. Performance of algorithms on GD.

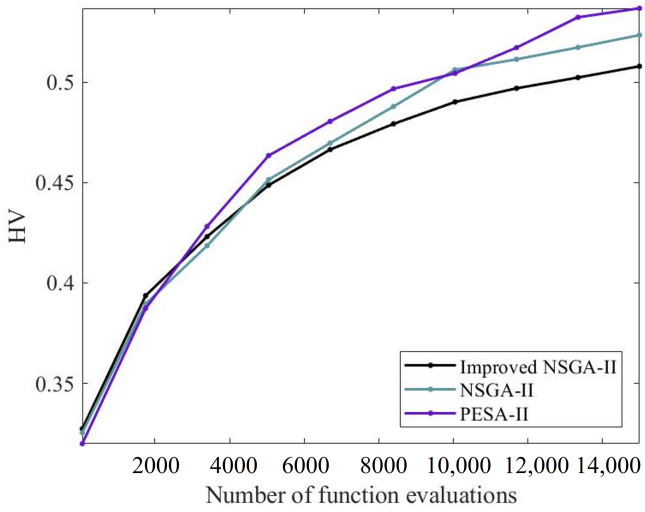

Figure 18. Performance of algorithms on HV.

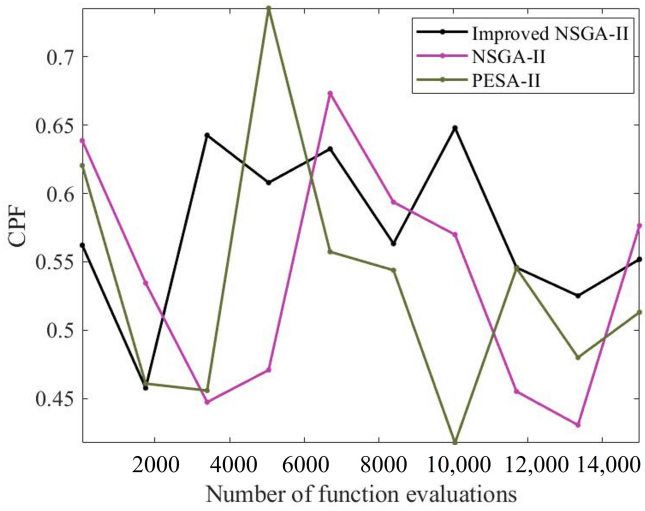

Figure 19. Performance of algorithms on CPF.

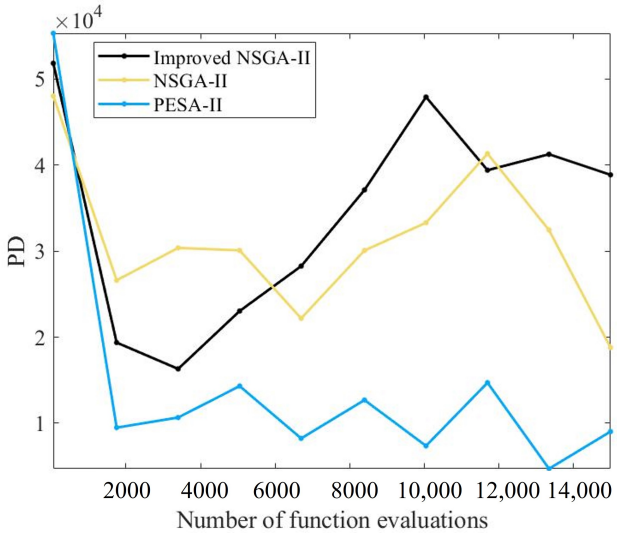

Figure 20. Performance of algorithms on PD.

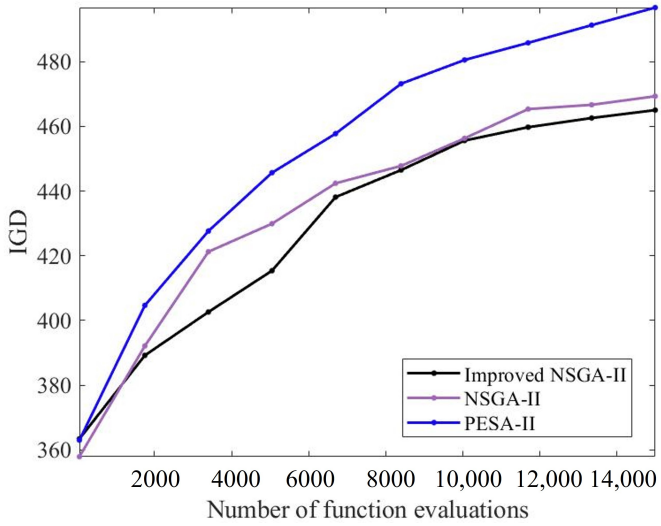

Figure 21. Performance of algorithms on IGD.

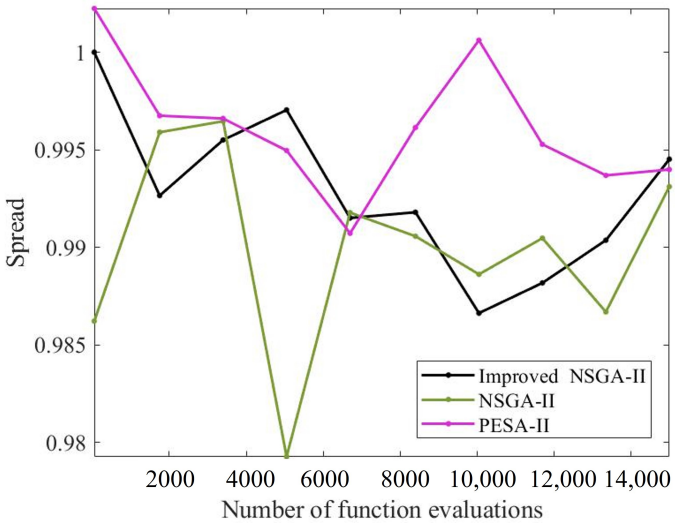

Figure 22. Performance of algorithms on MS.

The values highlighted in bold in Table 2 indicate superior performances across the evaluated metrics. The table clearly illustrates that the proposed IMNSGA-II outperforms the other two algorithms in four key indicators: GD, CPF, PD, and IGD. Moreover, the CT of IMNSGA-II is only 0.31 seconds slower than the fastest algorithm, PESA-II. These results underscore the efficacy of IMNSGA-II, particularly when compared to NSGA-II, demonstrating its superiority in optimizing multi-objective problems efficiently.

7.2. Validation of IMNSGA-II Using Benchmark Data

To address the current lack of benchmark datasets for the LIRP model incorporating STW and CS, we utilize the Solomon VRPTW benchmark available on Solomon's website (http://web.cba.neu.edu/~msolomon/problems.htm (accessed on 1 February 2024)). Specifically, we select datasets R101, R201, RC101, RC201, C101, and C201 for potential DCs and customers. These datasets represent various configurations that are suitable for illustrating the proposed model.

The experiments utilize datasets containing 25, 40, and 92 customers, each representing different distribution scenarios. The demand variance for each distributor is generated randomly following a normal distribution. A time window of 30 minutes is allowed

for both early and late deliveries, with start times set to 0 if they fall within this window. Additionally, the parameter settings used in the model are summarized in Table 3 as follows:

Table 3. Parameter setting of the model.

Parameter	Value	Parameter	Value
Inventory holding cost (i_h)	4	Fuel consumption per unit distance at full load (ρ^*)	0.388
Probability of being out of stock (α)	0.05	Carbon emission coefficient (ω)	0.0028
Safety stock factor (z_α)	0.95	Carbon tax C_e	0.0075
Order lead time (L)	7	Maximum driving distance of delivery vehicles (D)	1600
Unit order price (p)	9	Weight of fixed construction cost (λ_1)	0.0035
Carbon cap (C^{CAP})	1000	Weight of transportation cost (λ_2)	0.2823
Fixed cost of delivery vehicle (C_1)	100	Weight of inventory cost (λ_3)	0.0681
Penalty cost per unit time (α_1)	60	Weight of STW penalty cost (λ_4)	0.3715
Penalty cost per unit time (α_2)	90	Weight of carbon cost (λ_5)	0.2746
Fuel consumption per unit distance with no load (ρ_0)	0.122	Average speed of vehicle (v)	50
Number of vehicles (K)	11	Load of vehicle (Q_k)	200

In this section, the entropy method is used to determine the intermediate weight of the first objective function in our model. The proposed IMNSGA-II algorithm is implemented in MATLAB 2021a and executed using a laptop equipped with an 11th Gen Intel® Core™ i7-1165G7 processor running at 2.80 GHz and 16 GB of RAM.

For the experiment, a total of 500 iterations with a population size set to 200 are conducted. The computational results of this experiment are summarized in Table 4.

Table 4. Experimental results based on benchmark data.

Benchmark	G	T_c	W_c	P_c	C_c	f_1	f_2	R_t
R101 (j = 25)	4280	168,147	55,692	75,259	44,035	91,326	41.69%	41.26
R101 (j = 40)	4280	255,329	55,119	246,630	98,413	194,495	30.43%	45.44
R101 (j = 92)	4280	644,537	52,939	2,028,270	404,469	1,050,142	13.00%	49.88
R201 (j = 25)	4280	223,304	55,692	22,663	56,528	90,788	62.76%	46.89
R201 (j = 40)	4280	378,866	55,119	51,429	108,254	159,555	60.28%	41.86
R201 (j = 92)	4280	779,667	52,939	983,383	570,541	745,718	45.66%	47.85
RC101 (j = 25)	4280	167,530	55,252	58,913	87,324	96,937	43.78%	45.22
RC101 (j = 40)	4280	269,042	54,423	282,168	195,517	238,186	37.12%	45.58
RC101 (j = 92)	4280	743,871	52,187	2,417,121	589,771	1,273,475	8.93%	55.21
RC201 (j = 25)	4280	265,399	55,252	20,467	105,033	115,145	73.54%	46.00
RC201 (j = 40)	4280	409,679	54,423	25,897	225,191	190,832	80.75%	46.48
RC201 (j = 92)	4280	985,728	52,187	1,690,181	829,135	1,137,423	39.41%	47.99
C101 (j = 25)	4280	238,980	55,292	30,810	82,118	105,240	73.49%	42.73
C101 (j = 40)	4280	314,168	54,623	72,533	104,334	148,021	55.71%	44.74
C101 (j = 92)	4280	883,651	51,683	1,371,359	613,671	930,963	20.11%	47.75
C201 (j = 25)	3380	191,846	40,892	203,321	59,701	148,882	52.00%	39.24
C201 (j = 40)	3060	376,691	39,423	242,066	185,574	249,921	54.04%	39.55
C201 (j = 92)	4280	1,262,553	51,683	839,053	819,160	896,603	38.10%	60.81

G: fixed construction cost; T_c: transportation cost; W_c: inventory cost; P_c: TW penalty cost; C_c: carbon emission cost; f_1: TSCC with weights; f_2: CS; R_t: CPU time.

Based on Table 4, the reduction in CS can be attributed to the constrained availability of vehicles as the customer base expands, resulting in longer wait times and potential delays in deliveries. The fixed construction cost is directly influenced by the number of chosen DCs operationalized. Typically, benchmark datasets provide a choice of eight potential DCs, resulting in consistent fixed construction costs across various scenarios. It is crucial for companies to meticulously evaluate their DC selection and vehicle allocation

strategies to maintain optimal levels of customer satisfaction while efficiently managing operational costs.

7.3. Performance of MINLP Model and IMNSGA-II Based on a Case Study

To validate the efficacy of the proposed LIRP model and IMNSGA-II algorithm, a case study is conducted using an enterprise located in Jinan City, Shandong Province, China. The logistics distribution network of this enterprise will be optimized based on its specific developmental context and operational requirements. The W Enterprise primarily deals in fruits, vegetables, poultry, egg, and milk products, serving multiple urban areas within Jinan City.

Detailed information about the factories, alternative DCs, and distributors can be found in Appendix A (Tables A1 and A2). The experiments focus on optimization with eight potential DCs and fourty distributors. The vehicles have a capacity of 300 tons, and a carbon allowance cap (C^{CAP}) of 200 kgs is allocated along the supply chain. The other parameters match those listed in Table 3 in Section 7.2. All objective function weights are set to 1.

Optimizing the logistics distribution network involves considering factors like inventory capacity, distribution capability, and actual demand from distributors. This decision-making process should align with the company's strategic goals and market positioning, aiming to strike a balance between TSCC and CS. Since TSCC seeks to minimize its value and CS aims to maximize it, the CS objective function is transformed into its negative form when using the IMNSGA-II algorithm to find the minimum value. The W Enterprise can then select the most suitable decision plan by carefully evaluating these factors. This subsection aims to provide a comprehensive analysis of the decision results using the entropy–TOPSIS method, considering the specific needs and requirements of W Enterprise.

Based on the numerical experiments described above, the Pareto-optimal solution sets are obtained for fixed construction costs, transportation costs, inventory costs, time window penalty costs, and carbon emission costs. These solution sets are combined to form the decision matrix X. To determine the weights of fixed construction costs, transportation costs, inventory costs, time window penalty costs, and carbon emission costs, we utilize the entropy–TOPSIS method. The resulting weights are as follows:

$$\lambda_1 = 0.0035, \lambda_2 = 0.0013, \lambda_3 = 0.0681, \lambda_4 = 0.8746, \lambda_5 = 0.0524. \quad (47)$$

The weights obtained from Equation (47) are then incorporated into the model to generate the Pareto-optimal solution set. The Pareto-optimal solution set results obtained by IMNSGA-II are shown in Figure 23 below.

An analysis of the Pareto solution set reveals several key insights:

1. There exists a discernible trade-off between TSCC and CS.
2. Generally, an increase in TSCC correlates with a higher CS. In logistics distribution network optimization, prioritizing cost reduction often leads to decreased CS and potential delays in delivery times (as reflected in penalty costs), which can compromise service levels and operational efficiency.
3. Pursuing a higher CS involves meeting customer's expected TW more closely, typically resulting in increased total costs and potentially longer delivery times due to scheduling complexities.

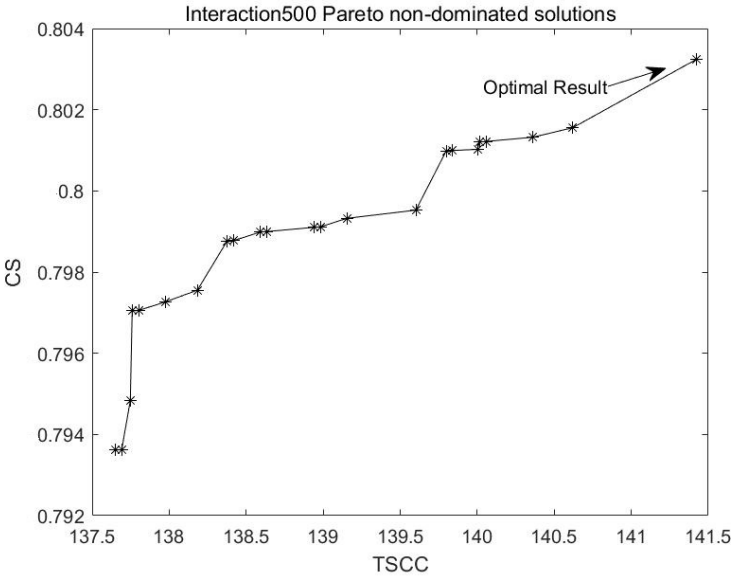

Figure 23. Pareto curve of TSCC with weight versus CS.

The entropy–TOPSIS method is employed to select the top 10 solutions from the Pareto-optimal solution set, as illustrated in Table 5.

Table 5. Top 10 solutions in the Pareto-optimal solution set.

Sort	G	T_c	W_c	P_c	C_c	f_1	f_2	R_t
1	4280	1505.379	1706.068	0.00	72.353	136.9115	0.8049	0.0371
2	4280	1556.070	1706.068	0.00	195.694	143.4405	0.7750	0.0371
3	4280	1667.489	1706.068	0.00	144.250	140.8897	0.7994	0.0370
4	4280	1715.731	1746.068	7.92	151.041	150.9576	0.7854	0.0368
5	4280	1594.770	1706.068	0.00	175.419	142.4284	0.7796	0.0367
6	4280	1557.522	1706.068	0.00	203.691	143.8614	0.7901	0.0367
7	4280	1586.902	1706.068	0.00	162.026	141.7163	0.7887	0.0366
8	4280	1774.005	1706.068	9.07	117.475	147.5548	0.7763	0.0366
9	4280	1617.298	1706.068	0.00	183.706	142.8919	0.7715	0.0365
10	4280	1773.187	1706.068	0.00	124.446	139.9893	0.7882	0.0363

The solution that is ranked first in Table 5 is taken as the best solution of the proposed LIRP model, and it is shown in the following Table 6. The layout in Figure 24 shows the routing of the nine vehicles for the 40 customers, where the points in red are the selected DCs.

Table 6. Optimal reference scheme of the proposed LIRP model.

Location of DCs: C_1, C_2, C_3, C_4.	**Ordered Quantity of DCs:** **265, 311, 326, 297.**
Distribution route:	
Vehicle 1: $C_1 \rightarrow 28 \rightarrow 3 \rightarrow 15 \rightarrow 8 \rightarrow 27 \rightarrow 24 \rightarrow C_1$	
Vehicle 2: $C_1 \rightarrow 39 \rightarrow 17 \rightarrow 32 \rightarrow 40 \rightarrow C_1$	
Vehicle 3: $C_2 \rightarrow 1 \rightarrow 20 \rightarrow 34 \rightarrow 18 \rightarrow 31 \rightarrow 23 \rightarrow C_2$	
Vehicle 4: $C_2 \rightarrow 30 \rightarrow 2 \rightarrow 4 \rightarrow 35 \rightarrow C_2$	Deliveries per vehicle:
Vehicle 5: $C_3 \rightarrow 19 \rightarrow C_3$	195, 150, 265, 175, 50, 180, 175, 300, 185.
Vehicle 6: $C_3 \rightarrow 33 \rightarrow 21 \rightarrow 37 \rightarrow 13 \rightarrow C_3$	Total distance: 302.6897.
Vehicle 7: $C_3 \rightarrow 7 \rightarrow 14 \rightarrow 11 \rightarrow 9 \rightarrow 29 \rightarrow C_3$	
Vehicle 8: $C_4 \rightarrow 5 \rightarrow 36 \rightarrow 12 \rightarrow 10 \rightarrow 26 \rightarrow 38 \rightarrow C_4$	
Vehicle 9: $C_4 \rightarrow 22 \rightarrow 25 \rightarrow 16 \rightarrow 6 \rightarrow C_4$	

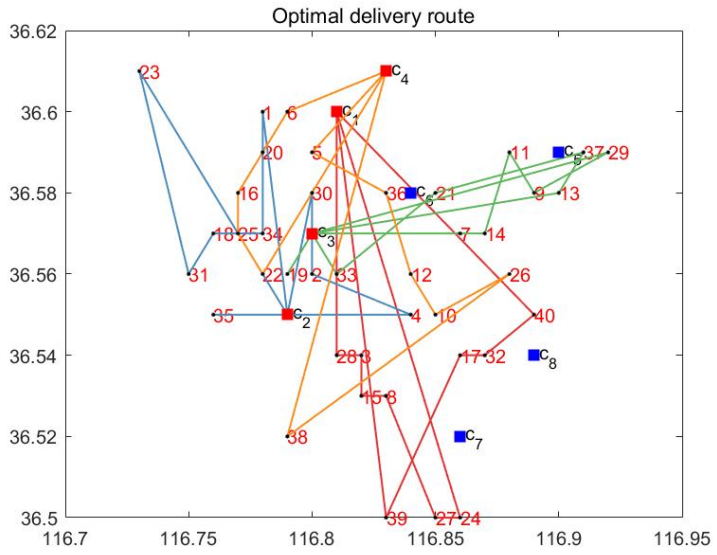

Figure 24. Optimal distribution roadmap of the proposed LIRP model.

8. Sensitivity Analysis

To evaluate the impact of different carbon quota thresholds on the solution quality and assist decision-makers in making informed choices, a sensitivity analysis is conducted. This paper begins with a smaller threshold value of 20 kg for an insufficient carbon quota and incrementally increases the step size by 20 kg until reaching a larger value of 280 kg, representing a sufficient carbon quota.

Figure 25 illustrates the trends in carbon emission cost and customer satisfaction (CS) as the carbon quota increases. Figure 26 shows the variations in carbon emission costs and total costs with changing carbon quota levels. The trend plots reveal that the carbon quota significantly influences the differences in carbon emissions and economic cost targets, while its effect on customer satisfaction is relatively minor. Figure 27 demonstrates that as the carbon quota increases from 20 kg to 280 kg, the economic cost decreases.

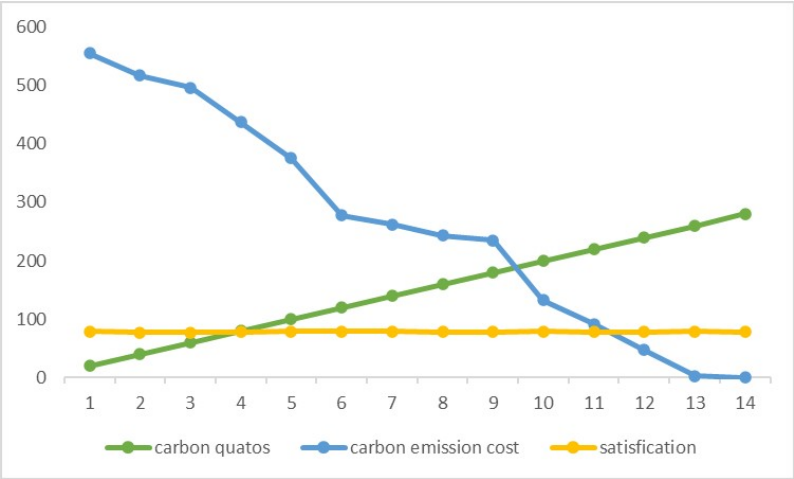

Figure 25. Trends in carbon emission costs and CS with increased carbon quotas.

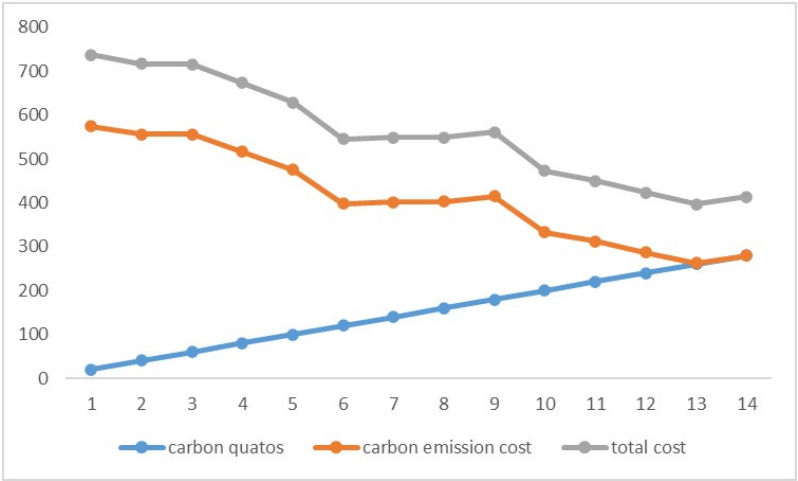

Figure 26. Trends in carbon emission costs and total costs with increased carbon quotas.

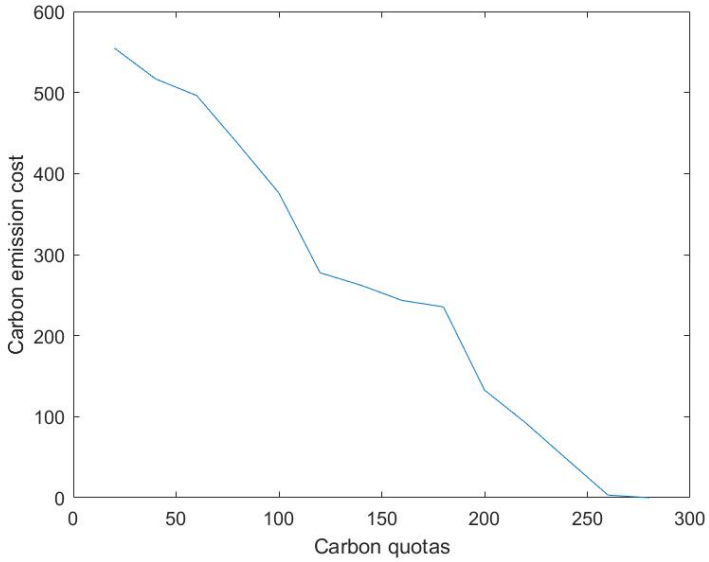

Figure 27. Trends in economic costs with increased carbon quotas.

Therefore, by combining the carbon emission quota and the trading data in the carbon emission mechanism, the model can recalculate the cost of carbon emissions, including considering the impact of the carbon trading mechanism on the cost. This analysis also indicates that a loose cap on carbon emissions could stimulate economic growth, but at the expense of significant environmental harm. Conversely, a stringent cap would benefit environmental sustainability but might constrain economic performance. The externalities associated with carbon allowances are substantial. Therefore, from a business perspective, it is essential to select the optimal decision-making scheme based on the preferences of decision-makers under varying carbon quotas.

9. Conclusions

This paper presents a novel approach by proposing a bi-objective mixed-integer nonlinear programming model with time window constraints that satisfies the normal distribution of stochastic customer demand. The proposed model aims to find Pareto-optimal solutions that minimize total costs and maximize customer satisfaction. An entropy-weight method is employed to evaluate the relationships among fixed costs, transportation costs, inventory costs, penalty costs, and carbon emission costs.

Additionally, an improved non-dominated sorting genetic algorithm II (IMNSGA-II) with an elite strategy is developed to solve the model. As the IMNSGA-II generates a

Pareto-optimal solution set, decision-makers must select the most-suitable strategy from this set. To facilitate this process, the entropy–TOPSIS method is employed to objectively rank the solution set, offering decision-makers with a solid basis for their choices.

To apply the proposed LIRP model to practical logistics scenarios, a case study is conducted on an enterprise in Jinan City of Shandong Province, China, specializing in fresh product logistics and distribution. The results offer valuable insights into optimizing the distribution system, demonstrating the algorithm's efficacy in tackling decisions related to the location of DCs, inventory management, and vehicle routing while considering factors such as stochastic customer demand, transportation costs, and facility capacities.

Furthermore, a sensitivity analysis is performed to evaluate the impact of adjusting the carbon quota threshold on the supply chain's performance. This analysis helps decision-makers to understand the trade-offs between carbon emissions and other key performance indicators. By varying the carbon quota threshold, decision-makers can effectively balance sustainability objectives with other performance metrics, ensuring an optimal balance between environmental considerations and overall supply chain efficiency.

In conclusion, this paper contributes to the field of logistics and supply chain management by offering a comprehensive approach that integrates TSCC, CEEI, and CS. The proposed model and algorithm provide decision-makers with the necessary tools to make informed decisions that align with their sustainability goals while optimizing various performance indicators. The entropy–TOPSIS method and sensitivity analysis further enhance decision-makers' understanding of the trade-offs, enabling them to make well-informed choices for their supply chain's overall performance and sustainability.

Future research into this area would involve including other sources of carbon emission costs like those that come from the storage of the fresh food. The question of diversifying the type of vehicles as well as different types of products is an interesting one and would definitely improve the impact of the model. The same is true of the normal assumption of customer demand compared to a fuzzy demand structure.

Author Contributions: Conceptualization, L.L. and L.S.L.; methodology, L.L. and L.S.L.; software, L.L. and A.H.; validation, L.L. and L.S.L.; formal analysis, L.L. and L.S.L.; investigation, L.L., L.S.L. and H.-V.S.; writing—original draft preparation, L.L., A.H. and L.S.L.; writing—review and editing, L.L., T.T., L.S.L. and H.-V.S.; funding, L.L., A.H. and T.T.; supervision, L.S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially supported by Guangxi University of Science and Technology through Natural Science Foundation of Guangxi Province (2023GXNSFAA026028).

Data Availability Statement: The data presented in this study are available from the corresponding author upon request.

Acknowledgments: We would like to thank the reviewers for taking the necessary time and effort to review this manuscript. We sincerely appreciate all their valuable comments and suggestions, which helped to improve the quality of this manuscript.

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

Appendix A

Table A1. Distributor information.

Distributor No.	Coordinates	Demand	DCs Construction Expected Delivery Time Window	Acceptable Delivery Time Window
1	(116.78, 36.60)	[40, 49]	[06.50–08.50]	[07.00–07.50]
2	(116.80, 36.56)	[30, 64]	[08.50–10.50]	[09.00–09.50]
3	(116.82, 36.54)	[25, 25]	[06.50–08.50]	[07.00–07.50]
4	(116.84, 36.55)	[50, 49]	[08.50–10.50]	[09.00–09.50]
5	(116.80, 36.59)	[35, 64]	[06.50–08.50]	[07.00–07.50]

Table A1. *Cont.*

Distributor No.	Coordinates	Demand	DCs Construction Expected Delivery Time Window	Acceptable Delivery Time Window
6	(116.79, 36.60)	[60, 25]	[08.50–10.50]	[09.00–09.50]
7	(116.86, 36.57)	[45, 49]	[08.50–10.50]	[09.00–09.50]
8	(116.83, 36.53)	[20, 64]	[06.50–08.50]	[07.00–07.50]
9	(116.89, 36.58)	[40, 25]	[08.50–10.50]	[09.00–09.50]
10	(116.85, 36.55)	[55, 49]	[06.50–08.50]	[07.00–07.50]
11	(116.88, 36.59)	[30, 64]	[08.50–10.50]	[09.00–09.50]
12	(116.84, 36.56)	[70, 25]	[06.50–08.50]	[07.00–07.50]
13	(116.90, 36.58)	[65, 49]	[06.50–08.50]	[07.00–07.50]
14	(116.87, 36.57)	[35, 64]	[08.50–10.50]	[09.00–09.50]
15	(116.82, 36.53)	[20, 25]	[06.50–08.50]	[07.00–07.50]
16	(116.77, 36.58)	[40, 49]	[08.50–10.50]	[09.00–09.50]
17	(116.86, 36.54)	[30, 64]	[08.50–10.50]	[09.00–09.50]
18	(116.76, 36.57)	[25, 25]	[06.50–08.50]	[07.00–07.50]
19	(116.79, 36.56)	[50, 49]	[06.50–08.50]	[07.00–07.50]
20	(116.78, 36.59)	[60, 64]	[06.50–08.50]	[07.00–07.50]
21	(116.85, 36.58)	[30, 25]	[06.50–08.50]	[07.00–07.50]
22	(116.78, 36.56)	[20, 49]	[08.50–10.50]	[09.00–09.50]
23	(116.73, 36.61)	[70, 64]	[06.50–08.50]	[07.00–07.50]
24	(116.86, 36.50)	[35, 25]	[06.50–08.50]	[07.00–07.50]
25	(116.77, 36.57)	[65, 49]	[08.50–10.50]	[09.00–09.50]
26	(116.88, 36.56)	[40, 64]	[06.50–08.50]	[07.00–07.50]
27	(116.85, 36.50)	[50, 25]	[06.50–08.50]	[07.00–07.50]
28	(116.81, 36.54)	[45, 49]	[06.50–08.50]	[07.00–07.50]
29	(116.92, 36.59)	[25, 64]	[08.50–10.50]	[09.00–09.50]
30	(116.80, 36.58)	[60, 25]	[08.50–10.50]	[09.00–09.50]
31	(116.75, 36.56)	[30, 49]	[06.50–08.50]	[07.00–07.50]
32	(116.87, 36.54)	[55, 64]	[08.50–10.50]	[09.00–09.50]
33	(116.81, 36.56)	[20, 25]	[06.50–08.50]	[07.00–07.50]
34	(116.78, 36.57)	[40, 64]	[06.50–08.50]	[07.00–07.50]
35	(116.76, 36.55)	[35, 25]	[08.50–10.50]	[09.00–09.50]
36	(116.83, 36.58)	[75, 49]	[06.50–08.50]	[07.00–07.50]
37	(116.91, 36.59)	[65, 64]	[06.50–08.50]	[07.00–07.50]
38	(116.79, 36.52)	[30, 25]	[06.50–08.50]	[07.00–07.50]
39	(116.83, 36.50)	[40, 49]	[08.50–10.50]	[09.00–09.50]
40	(116.89, 36.55)	[25, 64]	[08.50–10.50]	[09.00–09.50]

Table A2. Plant and alternative DC information.

Plant and Alternative DCs No.	Coordinates	Maximum Service Capacity	DC Construction Cost
0	(116.82, 36.50)	——	——
1	(116.81, 36.60)	320	1240
2	(116.79, 36.55)	360	920
3	(116.80, 36.57)	380	1220
4	(116.83, 36.61)	360	900
5	(116.90, 36.59)	400	1100
6	(116.84, 36.58)	360	980
7	(116.86, 36.52)	340	1050
8	(116.89, 36.54)	320	960

References

1. Qin, G.; Tao, F.; Li, L. A vehicle routing optimization problem for cold chain logistics considering customer satisfaction and carbon emissions. *Int. J. Environ. Res. Public Health* **2019**, *16*, 576. [CrossRef]
2. Anderson, E.W.; Sullivan, M.W. The Antecedents and Consequences of Customer Satisfaction for Firms. *Mark. Sci.* **1993**, *12*, 125–143. [CrossRef]

3. Bearden, W.O.; Teel, J.E. Selected Determinants of Consumer Satisfaction and Complaint Reports. *J. Mark. Res.* **1983**, *20*, 21–28. [CrossRef]
4. Wang, K.; Zhang, X.; Wei, Y.-M.; Yu, S. Regional allocation of CO_2 emissions allowance over provinces in China by 2020. *Energy Policy* **2013**, *54*, 214–229. [CrossRef]
5. Wang, X. Changes in CO_2 Emissions Induced by Agricultural Inputs in China over 1991–2014. *Sustainability* **2016**, *8*, 414. [CrossRef]
6. "Carbon Neutrality" Promotes the Transformation of Transportation Mode, and the Global Rail Transit Development Momentum Is Strong. Available online: https://m.gmw.cn/baijia/2021-09/26/35189553.html (accessed on 23 March 2021).
7. Wang, S.; Tao, F.; Shi, Y.; Wen, H. Optimization of Vehicle Routing Problem with Time Windows for Cold Chain Logistics Based on Carbon Tax. *Sustainability* **2017**, *9*, 694. [CrossRef]
8. Shariff, S.S.R.; Kamal, N.S.; Omar, M.; Moin, N.H. Location Routing Inventory Problem with Transhipment Points Using p-center. *J. Ind. Eng. Manag. Sci.* **2016**, *1*, 59–72. [CrossRef]
9. Zheng, J.; Li, K.; Wu, D. Models for Location Inventory Routing Problem of Cold Chain Logistics with NSGA-II Algorithm. *J. Donghua Univ. (Engl. Ed.)* **2017**, *34*, 533–539.
10. Li, K.; Li, D.; Wu, D. Multi-objective Optimization for Location-Routing-Inventory Problem in Cold Chain Logistics Network with Soft Time Window Constraint. *J. Eur. Des Systèmes Autom.* **2020**, *53*, 803–809. [CrossRef]
11. Wang, Z.; Wen, P. Optimization of a Low-Carbon Two-Echelon Heterogeneous-Fleet Vehicle Routing for Cold Chain Logistics under Mixed Time Window. *Sustainability* **2020**, *12*, 1967. [CrossRef]
12. Misni, F.; Lee, L.S.; Jaini, N.I. Multi-objective hybrid harmony search-simulated annealing for location-inventory-routing problem in supply chain network design of reverse logistics with CO_2 emission. *J. Phys. Conf. Ser.* **2021**, *1988*, 012054. [CrossRef]
13. Zhu, A.; Wen, Y.; Kaplan, M. Green Logistics Location-Routing Optimization Solution Based on Improved GA A1gorithm considering Low-Carbon and Environmental Protection. *J. Math.* **2021**, *2021*, 1–16.
14. Liu, A.; Zhu, Q.; Xu, L.; Lu, Q.; Fan, Y. Sustainable supply chain management for perishable products in emerging markets: An integrated location-inventory-routing model. *Transp. Res. Part E Logist. Transp. Rev.* **2021 2021**, *150*, 102319–102337. [CrossRef]
15. Shu, B.; Pei, F.; Zheng, K.; Yu, M.; Zhang, D.; Mohammed, S.; Calvi, A. LIRP optimization of cold chain logistics in satellite warehouse mode of supermarket chains. *J. Intell. Fuzzy Syst.* **2021**, *41*, 4825–4839. [CrossRef]
16. Govindan, K.; Salehian, F.; Kian, H.; Hosseini, S.T.; Mina, H. A location-inventory-routing problem to design a circular closed-loop supply chain network with carbon tax policy for achieving circular economy: An augmented epsilon-constraint approach. *Int. J. Prod. Econ.* **2023**, *257*, 108771. [CrossRef]
17. Fan, J.; Li, J.; Wu, Y.; Wang, S.; Zhao, D. The effects of allowance price on energy demand under a personal carbon trading scheme. *Appl. Energy* **2016**, *170*, 242–249. [CrossRef]
18. Lerhlaly, S.; Lebbar, M.; Allaoui, H.; Ouazar, D.; Afifi, S. An integrated inventory location routing: Problem considering CO_2 emissions. *Contemp. Eng. Sci.* **2016**, *9*, 303–314. [CrossRef]
19. Asadi, E.; Habibi, F.; Nickel, S.; Sahebi, H. A bi-objective stochastic location-inventory-routing model for microalgae-based biofuel supply chain. *Appl. Energy* **2018**, *228*, 2235–2261. [CrossRef]
20. Gholipour, S.; Ashoftehfard, A.; Mina, H. Green supply chain network design considering inventory-location-routing problem: A fuzzy solution approach. *Int. J. Logist. Syst. Manag.* **2020**, *35*, 436–452. [CrossRef]
21. Tavana, M.; Tohidi, H.; Alimohammadi, M.; Lesansalmasi, R. A location-inventory-routing model for green supply chains with low-carbon emissions under uncertainty. *Environ. Sci. Pollut. Res. Int.* **2021**, *28*, 50636–50648. [CrossRef]
22. Li, K.; Li, D.; Wu, D. Carbon Transaction-Based Location-Routing- Inventory Optimization for Cold Chain Logistics. *Alex. Eng. J.* **2022**, *61*, 7979–7986. [CrossRef]
23. Srinivas, N.; Deb, K.; Roy, P.; Mehta, S. Comparative study of vector evaluated GA and NSGA applied to multiobjective optimization. In Proceedings of the Symposium on Genetic Algorithms, Pittsburgh, PA, USA, 15–19 July 1995.
24. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
25. Forouzanfar, F.; Tavakkoli-Moghaddam, R.; Bashiri, M.; Baboli, A.; Molana, S.M.H. New mathematical modeling for a location-routing-inventory problem in a multi-period closed-loop supply chain in a car industry. *J. Ind. Eng. Int.* **2018**, *14*, 537–553. [CrossRef]
26. Mohammad, M.; Sadoullah, E.; Behnam, V. A bi-objective mathematical model for inventory distribution-routing problem under risk pooling effect: Robust metaheuristic approach. *Econ. Comput. Econ. Cybern. Stud. Res.* **2018**, *52*, 257–274. [CrossRef]
27. Monroy, A.G.A.; Díaz, H.L. A parallel programming approach to the solution of the location-inventory and multi-echelon routing problem in the humanitarian supply chain. *Transp. Res. Procedia* **2021**, *58*, 495–502. [CrossRef]
28. Liu, S.-C.; Lee, S.B. A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. *Int. J. Adv. Manuf. Technol.* **2003**, *22*, 941–950. [CrossRef]
29. Wang, M.; Wang, Y.; Liu, W.; Ma, Y.; Xiang, L.; Yang, Y.; Li, X. How to achieve a win–win scenario between cost and customer satisfaction for cold chain logistics? *Phys. A Stat. Mech. Its Appl.* **2021**, *566*, 125637. [CrossRef]
30. Xiao, Y.; Zhao, Q.; Kaku, I.; Xu, Y. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Comput. Oper. Res.* **2012**, *39*, 1419–1431. [CrossRef]

31. Wang, X. An integrated multi-depot location- inventory-routing problem for logistics distribution system planning of a chain enterprise. In Proceedings of the 2010 International Conference on Logistics Systems and Intelligent Management, Harbin, China, 9–10 January 2010; ICLSIM 2010.
32. Ahmadi Javid, A.; Azad, N. Incorporating location, routing and inventory decisions in supply chain network design. *Transp. Res. Part E Logist. Transp. Rev.* **2010**, *46*, 582–597. [CrossRef]
33. Kwon, Y.-J.; Choi, Y.-J.; Lee, D.-H. Heterogeneous fixed fleet vehicle routing considering carbon emission. *Transp. Res. Part D Transp. Environ.* **2013**, *23*, 81–89. [CrossRef]
34. Hsiao, Y.-H.; Chen, M.-C.; Chin, C.-L. Distribution planning for perishable foods in cold chains with quality concerns: Formulation and solution procedure. *Trends Food Sci. Technol.* **2017**, *61*, 80–93. [CrossRef]
35. Miller, B.L.; Goldberg, D.E. Genetic algorithms, tournament selection, and the effects of noise. *Complex Syst.* **1995**,*9*, 193–212.
36. Liu, J.; Chen, X. An improved NSGA-II algorithm based on crowding distance elimination strategy. *Int. J. Comput. Intell. Syst.* **2019**, *12*, 513–518. [CrossRef]
37. Rabbani, M.; Heidari, R.; Yazdanparast, R. A stochastic multi-period industrial hazardous waste location-routing problem: Integrating NSGA-II and Monte Carlo simulation. *Eur. J. Oper. Res.* **2019**, *272*, 945–961. [CrossRef]
38. Jingwen, H. Combining entropy weight and TOPSIS method for information system selection. In Proceedings of the 2008 IEEE Conference on Cybernetics and Intelligent Systems, Chengdu, China, 21–24 September 2008; pp. 1281–1284.
39. Zhu, Y.; Tian, D.; Yan, F. Effectiveness of Entropy Weight Method in Decision-Making. *Math. Probl. Eng.* **2020**, *2020*, 3564835. [CrossRef]
40. Hwang, C.-L.; Yoon, K. Methods for multiple attribute decision making. In *Multiple Attribute Decision Making;* Springer: Berlin/Heidelberg, Germany, 1981; pp. 58–191.
41. Schutze, O.; Esquivel, X.; Lara, A.; Coello, C.A.C. Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2013**, *16*, 504–522. [CrossRef]
42. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [CrossRef]
43. Tian, Y.; Cheng, R.; Zhang, X.; Li, M.; Jin, Y. Diversity Assessment of Multi-Objective Evolutionary Algorithms: Performance Metric and Benchmark Problems [Research Frontier]. *IEEE Comput. Intell. Mag.* **2019**, *14*, 61–74. [CrossRef]
44. Wang, H.; Jin, Y.; Yao, X. Diversity Assessment in Many-Objective Optimization. *IEEE Trans. Cybern.* **2017**, *47*, 1510–1522. [CrossRef] [PubMed]
45. Bosman, P.A.N.; Thierens, D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2003**, *7*, 174–188. [CrossRef]
46. Daroudi, S.; Kazemipoor, H.; Najafi, E.; Fallah, M. The minimum latency in location routing fuzzy inventory problem for perishable multi-product materials. *Appl. Soft Comput.* **2021**, *110*, 107543. [CrossRef]
47. Tian, Y.; Cheng, R.; Zhang, X.; Jin, Y. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]. *IEEE Comput. Intell. Mag.* **2017**, *12*, 73–87. [CrossRef]
48. Corne, D.W.; Knowles, J.D. Techniques for highly multiobjective optimization: Some nondominated points are better than others. In Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, UK, 7–11 July 2007; pp. 773–780.