Article

Joint Optimization of Delivery Routes for Perishable Goods from Multiple Distribution Centers in One-Echelon and Two-Echelon Networks Considering Time Satisfaction

Manqiong Sun , Yang Xu , Feng Xiao * , Hao Ji , Bing Su , Fei Bu

Posted Date: 19 November 2024

doi: 10.20944/preprints202411.1470.v1

Keywords: Joint distribution in one-echelon and two-echelon networks; soft and hard time windows; Perishable Goods Distribution

Article

Joint Optimization of Delivery Routes for Perishable Goods from Multiple Distribution Centers in One-Echelon and Two-Echelon Networks Considering Time Satisfaction

Manqiong Sun [1], Yang Xu [1], Feng Xiao [2,*], Hao Ji [1], Bing Su [1] and Fei Bu [1]

[1] School of Economics and Management, Xi'an Technological University, Xi'an 710021, China
[2] International Business School, Shaanxi Normal University, Xi'an 710119, China
* Correspondence: xiaofeng9727@snnu.edu.cn

Abstract: As the logistics industry modernizes, living standards improve, and consumption patterns shift, the demand for fresh food continues to grow, making cold chain logistics for perishable goods a critical component in ensuring food quality and safety. However, the presence of both soft and hard time windows among demand nodes can complicate single-network distribution of perishable goods. In response to these challenges, this paper proposes an optimization model for multi-distribution center perishable goods delivery, considering both one-echelon and two-echelon network joint distribution. The model aims to minimize total costs, including transportation, fixed, refrigeration, goods damage, and penalty costs, all while measuring customer satisfaction by the start time of service at each demand node. A two-stage heuristic algorithm is designed to solve the model. In the first stage, an initial solution is constructed using a greedy approach, based on the principles of the k-medoids clustering algorithm that considers both spatial and temporal distances. In the second stage, the initial routing solution is optimized using a linear programming approach from the Ortools solver combined with an Improved Adaptive Large Neighborhood Search (IALNS) algorithm. The effectiveness of the proposed model and algorithm is validated through a case study analysis. Results indicate that the enhanced algorithm achieves faster convergence and more effective overall cost optimization.

Keywords: joint distribution in one-echelon and two-echelon networks; soft and hard time windows; perishable goods distribution

1. Introduction

As living standards improve, the demand for perishable goods has progressively increased, making cold chain logistics a crucial component for ensuring these products reach consumers fresh and safe [1]. Beyond meeting customer expectations for product quality, service quality also plays a significant role in influencing customer satisfaction, which can be measured by the timeliness of product delivery. Consequently, delivering products within the time frames specified by customers is becoming increasingly important [2]. In real-world distribution, customers often specify desired delivery time intervals, requiring that delivery vehicles arrive within these periods. Depending on the strictness of the delivery time requirements, time windows are classified as either soft time windows or hard time windows [3]. Current research on cold chain distribution pathways for perishable goods typically considers either one-echelon or two-echelon networks alone. However, given that demand nodes frequently present varying and concurrent soft and hard time window requirements, single-network distribution in either one-echelon or two-echelon networks alone cannot adequately meet these time window demands at demand nodes. To address this issue more effectively, an optimization model for joint distribution of perishable goods across one-echelon and two-echelon networks is established. With the aim of minimizing total costs, the model is employed to determine the distribution routes of vehicles, considering the satisfaction levels at demand nodes.

The remainder of this paper is organized as follows: Section 2 provides a review of literature related to this paper. In Section 3, an optimization model for joint distribution of perishable goods across one-echelon and two-echelon networks is developed, addressing time satisfaction at both rigid and flexible demand nodes, a two-stage heuristic algorithm is introduced in Section 4. Section 5 details a case study that validates the effectiveness of the model and algorithm. Comparisons between the solutions from the first and second stages demonstrate that the initial solution obtained through the first stage is optimal, and the second stage shows improved optimization performance. Conclusions and potential future research avenues are discussed in Section 6.

2. Literature Review

Dantzig and Ramser were the pioneers in formulating the Vehicle Routing Problem (VRP) and developed its first heuristic algorithm [4]. As application scenarios evolved, research on VRP has expanded significantly. The Vehicle Routing Problem for Perishable Goods (VRPPG) is a subset of VRP, dealing specifically with products that rapidly deteriorate in quality [5]. The distribution of perishable goods was first addressed by Tarantilis and Kiranoudis who studied the Hybrid Fleet Vehicle Routing Problem (HFFVRP) and solved it using a metaheuristic threshold acceptance algorithm [6]. In the field of one-echelon network distribution paths for perishable goods, with the goal of minimizing unit costs while maximizing customer satisfaction, Gao Yuan Qin et al. assessed delivery punctuality as a measure of customer satisfaction. They developed a cold chain vehicle routing model and designed a genetic algorithm for its solution. Additionally, the impact of increased carbon pricing on emissions, customer satisfaction, and total costs was examined [7]. Anil Kumar Agrawal et al. addressed the vehicle routing problem for perishable goods using heterogeneous fleets. They incorporated time windows and quality requirements for perishable goods into the distribution process, established a mixed nonlinear programming model, and applied a genetic algorithm for its resolution [8]. With objectives of minimizing transportation costs, spoilage costs, and carbon emissions, and maximizing customer satisfaction, Zulvia et al. considered varying transportation times during peak and off-peak periods. They constructed a multi-objective perishable goods vehicle routing model and implemented a multi-objective gradient evolution algorithm for its solution [9]. Jiang Li et al. aimed to minimize the total cost and maximize customer satisfaction, taking into account customer appointment time windows. They crafted a multi-objective fresh produce delivery path model and employed an elitist strategy-based non-dominated sorting genetic algorithm for its resolution [10]. Focusing on minimizing total costs, with specific goals of minimizing quality loss costs and transportation costs, Partha Sarathi Barma and associates considered the distribution to two types of customers using refrigerated and standard vehicles. They established a multi-objective m-ring star-shaped distribution network model and utilized a non-dominated sorting genetic algorithm and strength Pareto evolutionary algorithm for its solution [11]. Huai and team aimed to minimize distribution costs and goods damage costs in their study. They constructed a multi-type vehicle routing model for perishable goods and solved it using a genetic algorithm based on two decoding methods [12].

The two-echelon VRP for perishable goods is a variant of the VRP that involves coordination between vehicular routes across two distinct hierarchical levels. In the realm of two-echelon networks for the distribution of perishable goods, objectives such as minimizing costs, reducing carbon emissions, and maximizing customer satisfaction are paramount. Wang Z et al. considered mixed time windows and carbon trading policies, developing a two-echelon cold chain vehicle routing model, and employed an adaptive genetic algorithm for its solution. Their research also examined the impact of variations in mixed time windows, the integration of customer satisfaction, and carbon trading policies on the model [13]. Aiming to minimize transportation and handling costs, reduce customer waiting times, and cut carbon emissions, Sahraeian R et al. considered customer satisfaction and distribution center capacity constraints. They developed a multi-objective mixed integer nonlinear programming model, which was solved using a non-dominated sorting genetic algorithm.

The effectiveness of this algorithm was validated through small, medium, and large-scale case studies [14]. In their research, Jaigirdar and colleagues aimed to minimize the total cost, average lead time for the delivery of fresh agricultural products, and initial investment cost for refrigerated distribution centers. They developed a two-echelon multi-objective mixed integer linear programming model for perishable goods. The CPLEX solver was employed to solve this model [15]. Targeting minimal total distribution costs, Ma Yanfang et al. considered the importance of customers, differentiating between regular and key customers. They constructed a two-echelon capacity-constrained vehicle routing optimization model and designed a two-stage heuristic algorithm for solving it. The first stage involved an enhanced genetic simulated annealing algorithm for the two-echelon distribution paths, while the second stage employed an exact method for the one-echelon distribution paths. The effectiveness of these algorithms was demonstrated through case study analysis [16].

Extensive research has been conducted on the vehicle route optimization problem for perishable goods distribution, yielding a wealth of theoretical results. However, most studies focus on one-echelon or two-echelon network distributions independently, predominantly considering single time window requirements, with few addressing both soft and hard time windows. In reality, demand nodes may have differing time requirements for services, potentially necessitating both soft and hard time windows for the delivery of perishable goods. A single network distribution may not suffice to meet these diverse needs. In response to this issue, an optimization model for joint distribution of perishable goods across multiple distribution centers within one-echelon and two-echelon networks is constructed, addressing demand node satisfaction within soft and hard time windows. To enhance solution efficiency, a two-stage heuristic algorithm is designed and its effectiveness is validated through a case study analysis.

3. One-Echelon and Two-Echelon Networks Model

An optimization model has been developed for the joint distribution of perishable goods, addressing satisfaction at two types of demand nodes within one-echelon and two-echelon networks. This model is noted for its complexity and numerous constraints, and is classified as an NP-hard problem. Due to the difficulties in obtaining optimal solutions with exact algorithms, a heuristic approach is employed for the solution. The model aims to minimize total costs and introduces a two-stage heuristic algorithm for this purpose. In the first stage, vehicle routing initial solutions are derived using a combination of k-medoids and greedy algorithms. The k-medoids algorithm, known for its robustness as it is not influenced by outliers and adapts well to various distributions, is utilized to cluster demand nodes. Considering the impact of initial cluster centers on clustering outcomes, the K-Means++ initialization method is employed to enhance the quality and efficiency of clustering. Subsequently, initial routing paths are constructed using a greedy algorithm. In the second stage, an improved Adaptive Large Neighborhood Search (ALNS) algorithm based on linear programming is proposed, leveraging the linear programming capabilities of the Ortools solver. This approach enhances the flexibility and local search capabilities, optimizing the initial paths obtained in the first stage.

3.1. Description of the Problem

A basic discrete undirected road traffic network $G=(V,E)$ is considered, where in $V_E=\{v_d\}\cup V_s\cup V_\mathrm{c}$ represents the set of nodes. Here, v_d denotes the supply point, V_s the collection of distribution center nodes, and V_c the set of customer nodes is subdivided into V_c^m where $V_c=V_c^1\cup V_c^2$; $m=1$ indicates nodes with hard time window requirements, and $m=2$ indicates nodes with soft time window requirements.Let $G_1=(V_1,E_1)$ represent first-echelon network, with $V_1=\{v_d\}\cup V_s\cup V_c^1$ being the nodes accessible for first-echelon vehicle distribution. $E_1=\{(i,j)\,|\,i,j\in V_1\}$ comprises the edges in the one-echelon network. $G_2=(V_2,E_2)$ is defined as second-echelon network, where $V_2=V_s\cup V_c$ encompasses nodes accessible for second-echelon vehicle distribution, and $E_2=\{(i,j)\,|\,i,j\in V_2\}$ constitutes the edge set of the second-echelon network. The supply point v_d has

doi:10.20944/preprints202411.1470.v1

K_1 fuel vehicles each with a carrying capacity of Q_1 and an average speed of v_1. The distribution center node v_s utilizes K_2 electric vehicles each with a carrying capacity of Q_2, an average speed of v_2, and a maximum travel distance T. The number of nodes with soft time window requirements is n_2, while those with hard time window requirements number n_1. Each customer node $i \in V_c$ has a demand d_i. The distance between any two points is d_{ij}. Soft time window nodes expect service within the time window $[ET_i, LT_i]$ and accept service within $[EET_i, ELT_i]$, whereas hard time window nodes accept service strictly within $[ET_i, LT_i]$. Given the limitations of single-network distribution in satisfying the time window requirements of some hard time window nodes, fuel vehicles in G_1 are scheduled to deliver perishable goods directly from the supply point to the nodes V_c^1. Concurrently, perishable goods are transported from the supply point to the distribution centers, where electric vehicles in G_2 continue the delivery to customer nodes V_c, ensuring all demands are met. In scenarios with multiple distribution centers, customer satisfaction regarding time is considered, aiming to minimize total costs (including fixed, refrigeration, transportation, goods spoilage, and time penalty costs). Strategies for joint one-echelon and two-echelon network distribution of perishable goods are developed and analyzed.

The model is based on the following assumptions:

(1) The locations and numbers of supply points, distribution centers, and demand nodes are known.
(2) The demands and service times at demand nodes are predefined and known.
(3) Upon arrival at a distribution center, one-echelon delivery vehicles unload their products, which are immediately loaded onto other one-echelon vehicles, with negligible handling time at the distribution center.
(4) Each demand node is visited only once.
(5) All soft time window demand nodes are serviced through distribution centers.
(6) There are no vehicle paths between distribution centers.

The joint distribution paths for perishable goods in the one-echelon and two-echelon networks are illustrated in Figure 1. In this cold chain network, there is one supply point, three distribution centers, and multiple types of demand nodes, serviced by several one-echelon fuel refrigerated vehicles and a specified number of two-echelon electric refrigerated vehicles. According to the demand at the nodes, perishable goods are either directly delivered to certain hard time window demand nodes by one-echelon vehicles or transported from the supply point to distribution centers, from where two-echelon vehicles distribute them to remaining demand nodes.

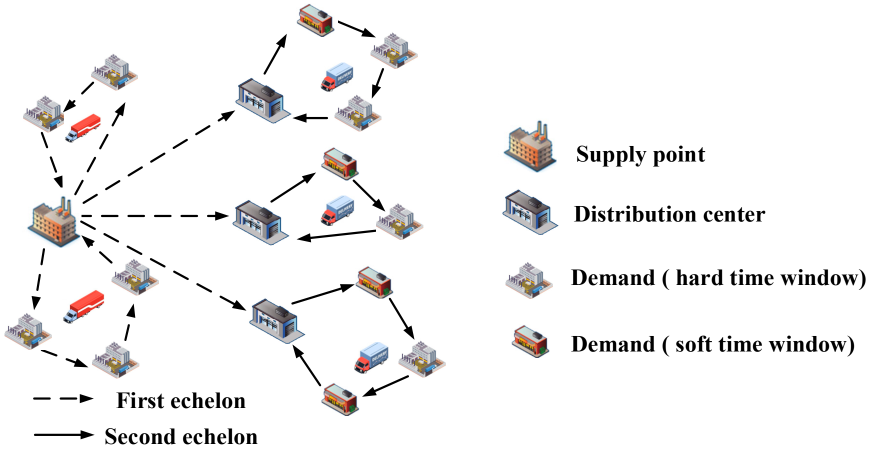

Figure 1. Joint Distribution Routes in One-Echelon and Two-Echelon Networks.

3.2. Notation Definition

The symbols and parameters used in the model are listed in Table 1, and the decision variables are shown in Table 2.

Table 1. Symbols and Parameters.

Set	Symbol Explanation
$V_E = \{v_d\} \cup \{v_s\} \cup V_c$	Set of all nodes
v_d	Supply point nodes
V_s	Set of Distribution center nodes
$V_c = V_c^1 \cup V_c^2$	Set of demand points
V_c^m	Subset of demand points with hard time windows(m=1), demand points with soft Time windows(m=2)
$V_1 = \{v_d\} \cup \{v_s\} \cup V_c^1$	Subset of nodes serviceable by primary vehicles
$V_2 = \{v_s\} \cup V_c$	Subset of nodes serviceable by secondary vehicles
$K_1 = \{1, 2, ..., k_1\}$	Set of primary transport vehicles
$K_2 = \{k_1 + 1, ..., k_1 + k_2\}$	Set of secondary transport vehicles

Set	Symbol Explanation
Q_1	Capacity of homogeneous primary transport vehicles
Q_2	Capacity of homogeneous secondary transport vehicles
fk_1	Fixed cost of primary vehicles
fk_2	Fixed cost of secondary vehicles
d_{ij}	Distance between node i and node j , $i, j \in V_E$
λ_i	Service time at node i , $i \in V_c$
v_1	Average speed of primary vehicles
v_2	Average speed of secondary vehicles
θ_1	Loss rate of perishable goods during transportation
θ_2	Loss rate of perishable goods during service
α_1	Unit time penalty cost for early arrival at soft time window customers
α_2	Unit time penalty cost for late arrival at soft time window customers
d_i	Demand at each demand point
C_r	Unit waiting cost for early arrival at demand points with hard time window
β_1	Unit refrigerant consumption rate during transport phase for primary vehicles
β_2	Unit refrigerant consumption rate during unloading phase for primary vehicles
ε_1	Unit refrigerant consumption rate during transport phase for secondary vehicles
ε_2	Unit refrigerant consumption rate during unloading phase for secondary vehicles
$[ET_i, LT_i]$	Service time window for demand points with hard time window
$[ET_i, LT_i]$	Expected service time window for demand points with soft time window
$[EET_i, ELT_i]$	Acceptable service time window for demand points with soft time window
P	Price of perishable products
a_1	Unit distance transportation cost for fuel vehicles
a_2	Unit distance transportation cost for electric vehicles

T	Maximum driving distance for electric vehicles
Q_{ij}^k	Load of vehicle k on (i,j), $i,j\in V_E, k\in K_1\cup K_2$
t_i^k	Arrival time of vehicle k at node i, $i\in V_E, k\in K_1\cup K_2$
t_{si}^k	Service start time of vehicle k at node i ， $i\in V_E, k\in K_1\cup K_2$

Table 2. Decision Variables.

Decision Variable	Explanation
x_{ij}^k	1 if vehicle k traverses (i,j), $k\in K_1\cup K_2, i,j\in V_E$; 0 otherwise
q_i^s	1 if the demand at customer node $i\in V_c$ is satisfied through a distribution center; 0 otherwise

3.3. Cost Analysis

(1) Fixed Vehicle Costs

Fixed vehicle costs chiefly refer to the expenses associated with operating each vehicle within both one-echelon and two-echelon distribution networks. These costs include driver salaries, vehicle depreciation, and maintenance fees, and are primarily related to the number of refrigerated vehicles utilized.

$$C_1=\sum_{i\in V_d}\sum_{j\in V_s\cup V_c^1}\sum_{k\in K_1}x_{ij}^k fk_1+\sum_{i\in V_s}\sum_{j\in V_c}\sum_{k\in K_2}x_{ij}^k fk_2 \tag{1}$$

(2) Variable Transport Costs

The transportation costs of a vehicle refer to the expenses incurred during the vehicle's operation, primarily fuel and electricity consumption. These costs are dependent on the distance traveled and the vehicle's load capacity.

$$C_2=\sum_{i\in V_1}\sum_{j\in V_1}\sum_{k\in K_1}a_1x_{ij}^k d_{ij}+\sum_{i\in V_2}\sum_{j\in V_2}\sum_{k\in K_2}a_2x_{ij}^k d_{ij} \tag{2}$$

(3) Refrigeration Costs

To ensure the freshness of perishable goods, refrigerated vehicles are employed for delivery. The refrigeration costs during delivery typically encompass the costs of refrigerants used to maintain the internal temperature of the cargo area. These costs include the energy consumed to keep the temperature low during transport and the additional energy required when the vehicle doors are opened during unloading, which causes an increase in the internal temperature.

$$C_3=\sum_{k\in K_1}\sum_{i\in V_1}\sum_{j\in V_1}x_{ij}^k(\beta_1\frac{d_{ij}}{v_1}+\varepsilon_1\lambda_i)+\sum_{k\in K_2}\sum_{i\in V_2}\sum_{j\in V_2}x_{ij}^k(\beta_2\frac{d_{ij}}{v_2}+\varepsilon_2\lambda_i) \tag{3}$$

(4) Cost of Goods Spoilage

The cost of goods spoilage refers to losses incurred due to the deterioration, rotting, and spoilage of products during transportation and service operations (such as when doors are opened) in cold chain logistics. A decay function is introduced from the literature. It models the quality degradation of perishable products, denoted as $D(t)=D_0e^{-\theta t}$. The spoilage rate varies because of the difference in the temperature inside the vehicle during delivery and when the doors are opened for unloading. It is posited that the spoilage rate during transportation is θ_1, and changes to $\theta_2(\theta_2>\theta_1)$ when the doors are opened, due to increased air convection and subsequent rise in temperature within the vehicle. The cost of goods spoilage can thus be represented as Equation (4).

$$C_4=P\left[\begin{array}{l}\sum_{i\in V_1}\sum_{j\in V_1}\sum_{k\in K_1}x_{ij}^kQ_{ij}^k\left(1-e^{-\theta_1\frac{d_{ij}}{v_1}}\right)+\sum_{i\in V_2}\sum_{j\in V_2}\sum_{k\in K_2}x_{ij}^kQ_{ij}^k\left(1-e^{-\theta_1\frac{d_{ij}}{v_2}}\right)\\ \sum_{i\in V_1}\sum_{j\in V_1}\sum_{k\in K_1}x_{ij}^kQ_{ij}^k\left(1-e^{-\theta_2\lambda_i}\right)+\sum_{i\in V_2}\sum_{j\in V_2}\sum_{k\in K_2}x_{ij}^kQ_{ij}^k\left(1-e^{-\theta_2\lambda_i}\right)\end{array}\right] \tag{4}$$

(5) Time Penalty Costs

Time window penalty costs include costs associated with early arrival within a hard time window, penalties for late arrival, early arrival penalties within a soft time window, and penalties for late arrival within a soft time window.

$$\begin{aligned} C_5 = \sum_{k\in K_1\cup K_2}\sum_{i\in V_c^1}\left[M\max\left(t_i^k - LT_i, 0\right) + C_r\max\left(ET_i - t_i^k, 0\right)\right] + \\ \sum_{k\in K_2}\sum_{i\in V_c^2}\left[\alpha_1\max\left(ET_i - t_i^k, 0\right) + \alpha_2\max\left(t_i^k - LT_i, 0\right)\right] \end{aligned} \tag{5}$$

(6) Satisfaction Function at Demand Nodes

The satisfaction function for demand nodes is incorporated through a membership function that accounts for service initiation times, assessing the satisfaction levels of demand nodes within a soft time window.

$$U\left(t_{si}^{k_2}\right) = \begin{cases} \dfrac{t_{si}^{k_2} - EET_i}{ET_i - EET_i} & t_{si}^{k_2} \in \left[EET_i, ET_i\right) \\ 1 & t_{si}^{k_2} \in \left[ET_i, LT_i\right] \\ \dfrac{ELT_i - t_{si}^{k_2}}{ELT_i - LT_i} & t_{si}^{k_2} \in (LT_i, ELT_i] \\ 0 & \text{others} \end{cases} \tag{6}$$

For demand nodes within a hard time window, the satisfaction function is defined as Equation (7).

$$S\left(t_{si}^{k}\right) = \begin{cases} 1 & t_{si}^{k_2} \in \left[ET_i, LT_i\right] \\ 0 & \text{others} \end{cases} \tag{7}$$

3.4. Model Development

The model is constructed considering the objectives and all constraints.

$$\min C_1 + C_2 + C_3 + C_4 + C_5 \tag{8}$$

$$s.t.\ \frac{1}{n_2}\sum_{i\in V_c^2}\sum_{k_2\in K_2} U\left(t_{si}^{k_2}\right) \geq A \tag{9}$$

$$\frac{1}{n_1}\sum_{i\in V_c^1}\sum_{k\in K_1\cup K_2} S\left(t_{si}^{k}\right) = 1 \tag{10}$$

$$\sum_{i\in v_d}\sum_{j\in V_c^1\cup V_s} x_{ij}^k Q_{ij}^k \leq Q_1 x_{ij}^k, \forall k\in K_1 \tag{11}$$

$$\sum_{i\in V_s}\sum_{j\in V_c} x_{ij}^k Q_{ij}^k \leq Q_2 x_{ij}^k, \forall k\in K_2 \tag{12}$$

$$\sum_{j\in V_E} x_{ji}^k = \sum_{j\in V_E} x_{ij}^k, \forall i\in V_c, \forall k\in K_1\cup K_2 \tag{13}$$

$$\sum_{k\in K_1\cup K_2}\sum_{i\in V_E} x_{ij}^k = 1, \forall j\in V_c \tag{14}$$

$$\sum_{i\in v_d}\sum_{j\in V_c^1\cup V_s} x_{ij}^k = \sum_{j\in V_c^1\cup V_s}\sum_{i\in v_d} x_{ji}^k, \forall k\in K_1 \tag{15}$$

$$\sum_{i\in V_s}\sum_{j\in V_c} x_{ij}^k = \sum_{j\in V_c}\sum_{i\in V_s} x_{ji}^k, \forall k\in K_2 \tag{16}$$

$$\sum_{i\in v_d}\sum_{j\in V_1} x_{ij}^k \leq 1,\ \forall k\in K_1 \tag{17}$$

$$\sum_{i\in V_s}\sum_{j\in V_2} x_{ij}^k \le 1,\ \forall k\in K_2 \tag{18}$$

$$\sum_{s\in V_s} q_i^s = 1, \forall i\in V_c \tag{19}$$

$$\sum_{i\in V_s}\sum_{j\in V_s} x_{ij}^k = 0, \forall k\in K_1\cup K_2 \tag{20}$$

$$x_{V_s v_d}^k = x_{v_d V_s}^k, \forall k\in K_1 \tag{21}$$

$$\sum_{i\in V_s}\sum_{j\in V_c}\sum_{k\in K_2} x_{ij}^k Q_{ij}^k = \sum_{i\in V_c}\sum_{s\in V_s} q_i^s d_i \tag{22}$$

$$\sum_{i\in v_d}\ \sum_{j\in\ V_s\cup\ V_c^1}\ \sum_{k\in K_1} x_{ij}^k Q_{ij}^k = \sum_{i\in V_c} d_i \tag{23}$$

$$\sum_{i\in\ V_E}\ \sum_{k\in K_1\cup\ K_2} x_{ij}^k Q_{ij}^k - \sum_{i\in V_E}\ \sum_{k\in K_1\cup K_2} x_{ij}^k Q_{ij}^k = d_j, \forall j\in V_c \tag{24}$$

$$\sum_{i\in V_c} q_i^s d_i \le Q_1 \sum_{k\in K_1} x_{v_d V_s}^k \tag{25}$$

$$\sum_{i\in V_2}\sum_{j\in V_2} x_{ij}^k d_{ij} \le T, \forall k\in K_2 \tag{26}$$

$$t_j^k = \sum_{k\in K_1}\sum_{i\in V_1}\left(t_{si}^k + \lambda_i + \tfrac{d_{ij}}{v_1}\right)\ x_{ij}^k, \forall j\in V_1 \tag{27}$$

$$t_j^k = \sum_{k\in K_2}\sum_{i\in V_2}\left(t_{si}^k + \lambda_i + \tfrac{d_{ij}}{v_2}\right) x_{ij}^k, j\in V_2 \tag{28}$$

$$t_{si}^k = \max\left(t_i^k, ET_i\right)\ ,\ i\in V_c^1, k\in K_1\cup K_2 \tag{29}$$

$$t_{si}^k = \max\left(t_i^k, EET_i\right)\ ,\ i\in V_c^2, k\in K_2 \tag{30}$$

$$EET_i \le t_{si}^k \le ELT_i, i\in V_c^2, k\in K_2 \tag{31}$$

$$ET_i \le t_{si}^k \le LT_i, i\in V_c^1, k\in K_1\cup K_2 \tag{32}$$

$$t_i^{k_1} = t_{si}^{k_2}, \forall i\in V_s, k_1\in K_1, k_2\in K_2 \tag{33}$$

$$x_{ij}^k \in \left\{0,1\right\}, i, j\in V_E, k\in K_1\cup K_2 \tag{34}$$

$$q_i^s \in \left\{0,1\right\}, i\in V_c, s\in V_s \tag{35}$$

$$Q_{ij}^k,, t_i^k, t_{si}^k \ge 0,\ \ i, j\in V_E, k\in K_1\cup K_2 \tag{36}$$

Equation (8) is aimed at minimizing the total costs. Equation (9) is designed to enforce satisfaction constraints at demand nodes with soft time windows. Equation (10) is formulated to enforce satisfaction constraints at demand nodes with hard time windows. Equations (11) and (12) are intended to limit the load capacity of one-echelon vehicles. Equations (13) and (14) ensure that each demand node is serviced by only one vehicle. Equation (15) stipulates that each one-echelon vehicle departing from a supply point must return to that supply point. Equation (16) dictates that each two-echelon vehicle departing from a distribution center returns to that distribution center. Equation (17) limits the use of each one-echelon vehicle to a single operation. Equation (18) limits the use of each two-echelon vehicle to a single operation. Equation (19) assigns each demand node to a distribution center. Equation (20) ensures that no vehicular paths exist between distribution centers. Equation (21) ensures connectivity between the supply point and distribution centers.

Equation (22) asserts that the total quantity of goods leaving distribution centers meets the demand of assigned demand nodes. Equation (23) ensures that the quantity of goods departing from the supply point is sufficient to meet the demands of all demand nodes. Equation (24) maintains the conservation of vehicle load weight, thereby eliminating subconstraints and satisfying the requirements at each demand node. Equation (25) asserts that the quantity of goods leaving the supply point is at least equal to that leaving from distribution centers. Equation (26) ensures that the travel distance of each two-echelon vehicle does not exceed specified limits. Equation (27) addresses the continuity of service times from the supply point to distribution centers and to demand nodes with hard time windows. Equation (28) concerns the continuity of service times from distribution centers to demand nodes. Equation (29) sets the start times for services at demand nodes with hard time windows. Equation (30) sets the start times for services at demand nodes with soft time windows. Equation (31) imposes constraints on the time windows at demand nodes with soft time windows. Equation (32) imposes constraints on the time windows at demand nodes with hard time windows. Equation (33) disregards the loading and unloading times at distribution centers. Equations (34) to (36) define the permissible ranges for variables.

4. Two-Stage Heuristic Algorithm

In the present study, a two-tier heuristic algorithm is employed to address the VRP for the joint distribution of perishable goods through one-echelon and two-echelon networks. This problem is characterized by its high model complexity and numerous constraints, classifying it as an NP-hard issue where exact algorithms struggle to find optimal solutions. Consequently, heuristic methods are utilized to derive solutions. For the problem at hand, a two-stage heuristic algorithm has been designed. In the first stage, the vehicle routing initial solution is obtained using the k-medoids clustering algorithm and a greedy algorithm. The k-medoids clustering algorithm is selected for its robustness, as it is not influenced by outliers and offers strong adaptability to different distributions. This algorithm also provides a relatively fast solution speed and accurate data set partitioning. Therefore, a time-space distance-based k-medoids clustering algorithm is applied to cluster the demand nodes. Following the clustering results, a greedy algorithm is employed to construct the initial route. Moreover, considering the impact of the initial clustering centers on the clustering outcome, the K-Means++ initialization method is utilized to select the initial clustering centers, thereby enhancing the quality and efficiency of the clustering. In the second stage, a hybrid approach that integrates the linear programming ideas of the Ortools solver with the flexibility and local search capabilities of the ALNS algorithm is proposed to optimize the initial routes. This approach leverages a linear-programming-enhanced ALNS algorithm, which offers a robust mechanism for addressing the complexities associated with the problem.

4.1. K-Center Clustering Algorithm Considering Spatio-Temporal Distance

In addressing routing optimization challenges within VRP, clustering algorithms are commonly employed, with distance serving as the primary factor. In this paper, it is argued that clustering demand nodes that are close geographically but have significant differences in their time windows can lead to unfeasible solutions or result in low transportation costs offset by high penalty costs related to time windows. Consequently, an approach that integrates both spatial and temporal distances has been adopted. The concept of "spatio-temporal distance" is introduced, and a k-center clustering algorithm based on this concept is proposed to cluster customers with similar spatio-temporal proximities, thereby preventing the aforementioned problems.

Spatio-temporal distance is calculated as a weighted average of the normalized spatial and temporal distances between customers. Considering both soft and hard time windows for customer demands, assume customers at nodes i and j have time windows $\left[ET_i, LT_i\right]$ ($\left[EET_i, ELT_i\right]$) and $\left[ET_j, LT_j\right]$ ($\left[EET_j, ELT_j\right]$), respectively. It is assumed that $ET_i(EET_i) \le ET_j(ELT_j)$, and vehicle k_2 arrives at node i at time $t_i^k \in \left[ET_i, LT_i\right]$ ($\left[EET_i, ELT_i\right]$). The arrival time of vehicle k_2 at node j then falls within the interval $\left[ET_i(EET_i)+\lambda_i+t_{ij}, LT_i(ELT_i)+\lambda_i+t_{ij}\right]$, where $ET_i(EET_i)+\lambda_i+t_{ij}=a$, $ET_i(ELT_i)+\lambda_i+t_{ij}=b$.

The temporal distance between points i and j is categorized as follows when point j is subject to a hard time window:

doi:10.20944/preprints202411.1470.v1

(1) If $b<ET_j$, then the vehicle is considered to have arrived at demand node j before the earliest acceptable service time. A penalty coefficient, C_r, is applied, and the temporal distance between points i and j is determined as $C_r(ET_j-b)$.

(2) If $a>LT_j$, then the vehicle is considered to have arrived after the latest acceptable service time at hard time window demand node j. A penalty coefficient, M, is applied, and the temporal distance is determined as $M(a-LT_j)$.

(3) If $a\le ET_j<b$ or $a<LT_j<b$, then the vehicle's arrival time at demand node j aligns with the expected time window. A deviation coefficient, δ, is applied, and the temporal distance is determined as $\delta(\lambda_i+d_{ij}/v_2)$.

(4) If $ET_j<a<b<LT_j$, then the vehicle's arrival at demand node j is considered to be perfectly within the desired time window, and the temporal distance is calculated as λ_i+t_{ij}.

For customer point j characterized by a soft time window, the following scenarios are considered:

(1) If $b<EET_j$, meaning the vehicle arrives before the earliest acceptable service time within the soft time window, a penalty coefficient M is applied. The temporal distance is calculated as $M(EET_j-b)$.

(2) If $a>ELT_j$, indicating that the vehicle arrives after the latest acceptable service time for a customer point j with a soft time window, the penalty coefficient M is again applied. The temporal distance is calculated as $M(a-ELT_j)$.

(3) If $EET_j<a<b<ET_j$, the vehicle's arrival time at customer point j falls between the expected and acceptable time windows. A penalty coefficient α_1 is applied, and the temporal distance between points i and j is calculated as $\alpha_1(ET_j-b)$.

(4) If $LT_j<a<b<ELT_j$, the vehicle's arrival at customer point j also falls between the expected and acceptable time windows. A penalty coefficient α_2 is applied, and the temporal distance is calculated as $\alpha_2(a-LT_j)$.

(5) If $EET_j<a\le ET_j<b<LT_j$ or $ET_j<a<LT_j<b<ELT_j$, the vehicle's arrival time at customer point j coincides with the expected time window. A deviation coefficient δ is applied, and the temporal distance between points i and j is calculated as $\delta(\lambda_i+d_{ij}/v_2)$.

(6) If $ET_j<a<b<LT_j$, the vehicle perfectly arrives within the expected time window at customer point j, and the temporal distance is computed as λ_i+t_{ij}.

In summary, when calculating the temporal distance between nodes i and j, with node j being a hard time window demand node, the formula for temporal distance is defined according to the aforementioned scenarios.

$$d_{ij}^T=\begin{cases}C_r(ET-b),b<ET_j\\ M(a-LT_j),a>LT_j\\ a\le ET_j<b\ or\ a<LT_j<b,\delta(\lambda_i+d_{ij}/v_2)\\ ET_j<a<b<LT_j,\lambda_i+t_{ij}\end{cases}\tag{37}$$

When calculating the temporal distance between demand nodes i and j, where node j has a hard time window requirement, the formula for temporal distance is as follows:

$$d_{ij}^{T}=\begin{cases} M\left(EET_j-b\right),b<EET_J \\ M\left(a-ELT_j\right),a>ELT_j \\ EET_j<a<b<ET_j,\alpha_1(ET_j-b) \\ LT_j<a<b<ELT_j,\alpha_2(a-LT_j) \\ EET_j<a\le ET_j<b<LT_j \text{ or } ET_j<a\le LT_j<b<ELT_j,\delta(\lambda_i+d_{ij}/v_2) \\ ET_j<a<b<LT_j,\lambda_i+t_{ij} \end{cases} \tag{38}$$

The spatial distance between demand nodes i and j is defined as $d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$.

Weights are assigned to spatial and temporal distances to derive the formula for spatio-temporal distance d_{ij}^{ST} as shown in Equation (39):

$$d_{ij}^{ST}=\partial d_{ij}+(1-\partial)d_{ij}^{T} \tag{39}$$

The specific implementation process of the k-center clustering algorithm is as follows:

Step 1: A random point from the demand nodes is selected as the first clustering center. The minimum spatio-temporal distance d_{ij}^{ST} from each demand node to the nearest cluster center is calculated. The point with the maximum d_{ij}^{ST} is chosen as the next cluster center. This process is repeated until three initial cluster centers are identified.

Step 2: The spatio-temporal distance from each demand node to the three cluster centers is calculated. According to the principle of nearest spatio-temporal distance, all demand nodes are assigned to the corresponding cluster center dataset, resulting in three clusters.

Step 3: For each cluster, the dataset is traversed assuming each data point as a cluster center. The sum of distances from all other points to this point is calculated, and the point with the smallest sum of distances is selected as the new cluster center.

Step 4: The spatio-temporal distance from each demand node to the new cluster center is recalculated, and the points are reassigned based on the nearest principle.

Step 5: After each reassignment, the cluster centers are updated until there are no changes in the cluster dataset. The distance from each cluster center to the distribution center is calculated. Each cluster is assigned to the nearest distribution center, and the clustering results are output.

4.2. Initial Solution Construction

After clustering the demand nodes, a greedy algorithm is used to generate the initial distribution route. The specific steps are as follows:

Step 1: First, construct direct network paths using a one-echelon network for demand nodes with hard time windows that cannot be served through a two-echelon network according to the principles of the greedy algorithm. Then, based on the clustering results, calculate the distance from each demand node in the distribution center cluster to its respective distribution center. Arrange these distances in descending order and sequentially insert the closest demand nodes into the route while satisfying load and time window constraints.

Step 2: Iterate through the cluster center collection, selecting the nearest demand node to the cluster center as the first service object. Insert the demand node into the path, and perform load and time window feasibility checks on the route. If the constraints are not met, output this path.

Step 3: Repeat Step 2 for the remaining unassigned demand nodes in the cluster.

Step 4: Check if all cluster data sets have been serviced. If not, return to Step 2; if so, output the initial delivery route.

doi:10.20944/preprints202411.1470.v1

4.3. Enhanced ALNS Algorithm Based on Linear Programming

In comparison to the conventional ALNS algorithm, the enhanced version developed in this paper incorporates four types of deletion operators and three types of repair operators based on the classification of demand nodes, along with two swap operators to expand the search space. Rather than using a fixed scoring strategy for operators, operators are scored based on the objective value of each iteration, which assists in escaping local optima. It has been observed that updating operator weights at each iteration results in superior outcomes by effectively avoiding local optima, increasing operator randomness, enhancing the diversity of solutions, and broadening the search space. Hence, a strategy of updating weights with each iteration is adopted.

The enhanced ALNS algorithm, which integrates the linear programming solution approach of the Ortools solver, operates on the principle of storing updated solutions from each iteration in a set of sub-tour solutions. When the iteration count i modulo 30 equals zero, to increase the diversity of solutions, a set of 1000 random sub-tours is generated. The Ortools solver's linear programming framework is then applied to both the stored and the randomly generated sub-tours to determine the optimal sub-tour combination, which forms a new solution. This new solution is established as the current solution. If the termination criteria are not met, this current solution continues to be used in subsequent iterations of the ALNS algorithm until the termination conditions are fulfilled.

4.3.1. Destruction Operators

(1) Random Removal Operator

This operator randomly selects q demand nodes for deletion. While it may generate lower-quality solutions, it increases neighborhood diversity and expands the search space, assisting in escaping from local optima.

(2) Worst Cost Saving Removal Operator

This operator removes q demand nodes that offer the greatest cost savings. Cost savings refer to the reduction in the objective function value when a demand node is removed from the route. Demand nodes are ranked by their cost-saving potential, and the top q are removed, effectively reducing the total distribution cost.

(3) Route Deletion Operator

In this operator, a route with the fewest number of service demand nodes or the highest cost is selected randomly, and all demand nodes on this route are deleted. If the number of deleted demand nodes is less than q, another route is randomly selected until the total number exceeds or equals q. This operator can enhance vehicle utilization and reduce overall transportation costs.

(4) Worst Satisfaction Removal Operator

This operator sorts demand nodes with soft time windows by decreasing satisfaction of time constraints and removes the top q nodes with the lowest satisfaction. This effectively removes demand nodes with poor time satisfaction from the route, potentially improving overall route efficiency and customer satisfaction.

4.3.2. Repair Operators Based on Demand Node Types

(1) Cost-Considerate Greedy Insertion Operator Based on Demand Node Type

Demand nodes are categorized into three types: those with hard time windows deliverable only via one-echelon routes, those with soft time windows deliverable through either first-echelon or second-echelon routes, and the remainder with hard time windows.

Step 1: Calculate the optimal insertion node for each removed demand node to be reinserted into the route, aiming to minimize the increase in the objective function.

Step 2: Select the demand node with the lowest integration cost, and assess the type of demand node. Subject to vehicle capacity constraints and the satisfaction of the time windows, demand nodes with hard time windows that are restricted to one-echelon routes are reinserted into one-echelon distribution paths; otherwise, they are inserted into two-echelon routes.

Step 3: Update the integration costs of the remaining removed demand nodes and repeat Step 2 until all removed demand nodes are reinserted into the route.

(2) Random Insertion Operator Based on Demand Node Type

Removed demand nodes are randomly inserted into the current solution within the constraints. Demand nodes with hard time windows that must be serviced via first-echelon routes are inserted into first-echelon distribution paths, while other demand nodes are randomly integrated into second-echelon routes.

(3) New Vehicle Insertion Operator Based on Demand Node Type

Under the constraints, demand nodes are categorized for distribution. For those with hard time windows that can only be serviced via first-echelon routes, a new first-echelon vehicle is added to distribute them in a random order. Other demand nodes are serviced by adding a second-echelon vehicle also distributing in a random order.

(4) Best Timing Insertion Operator Based on Demand Node Type

Before adding demand node i to a route, it is necessary to confirm that the service timing at this node falls within its acceptable time window. Additionally, the arrival times at subsequent nodes on the route must also meet their respective time windows. Based on these constraints, the time deviation $BT(i)=\left|t_i^k-ET_i\right|$ is calculated. The demand node i is then inserted into the position where $BT(i)$ is minimized.

4.3.3. Swap Operators

(1) Single-Vehicle Swap Operator

Randomly swap the order of demand nodes visited on a single vehicle's route to generate a new route, as depicted in the Figure 2.

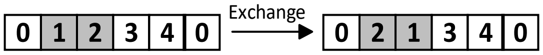

Figure 2. Schematic Diagram of the Single-Vehicle Exchange Operator.

(2) Multi-Vehicle Swap Operator

Randomly perform a swap of demand nodes between two vehicles' routes to generate a new route configuration, as shown in the Figure 3.

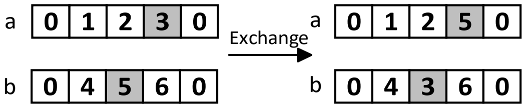

Figure 3. Schematic Diagram of the Multi-Vehicle Exchange Operator.

4.3.4. Acceptance Criteria, Operator Scoring Strategy, Operator Weight Update, and Termination Conditions

During the iterative process, non-improving solutions are accepted. These are updated solutions that are not superior to current solutions. Acceptance is based on the judgment criteria of the simulated annealing algorithm. This algorithm accepts solutions with a certain probability. In the presence of multiple deletion, repair, and exchange operators, a roulette wheel method is used for selection. Initially, all operators are assigned a weight of zero. Operator scores are updated in each iteration. Updates are based on the performance of deletion, repair, and exchange operators. Operators receive a score increment calculated as (1/objective function of this iteration)+a very small number. These increments allow for the updating of weights for deletion and repair operators based on the scores obtained.

$$w_i=w_{i-1}(1-r)+r\frac{\prod_i}{\theta_i} \tag{40}$$

where w_i represents the weight of the operator at iteration i, r is the weight update parameter, $\prod_i$ is the total score of the operator at iteration i, and θ_i represents the total number of times the operator has been used up to iteration i.

To prevent certain operators from being overlooked, a special method is used. High-weight operators are often repeatedly selected. This can reduce the search capabilities of the algorithm in its later stages. When operators with a weight of zero exist, they are preferentially selected. This selection is done using the roulette wheel method. In this research, the algorithm terminates when the maximum number of iterations is reached.

4.3.5. Two-Stage Heuristic Algorithm Basic Solving Process

Step 1: Construct initial route solutions using a space-time distance-based k-center algorithm and greedy algorithm.

Step 2: Set the initial solutions as the current and best solutions, and establish initial parameters, including initial scores and weights for the operators and the initial temperature for simulated annealing.

Step 3: Check if the termination conditions are met. If so, output the optimal solution; otherwise, proceed to Step 4.

Step 4: Update the weights of the destruction, repair, and swap operators.

Step 5: Select destruction, repair, and swap operators using the roulette wheel method to process the current solution and generate a new feasible solution, and proceed to Step 6.

Step 6: Compare the objective function value of the new feasible solution with that of the current solution. If the new solution improves the objective function, set this new solution as the current solution, then continue to Step 7; otherwise, proceed to Step 8.

Step 7: Compare the current solution with the best solution. If the current solution improves upon the best solution, set the current solution as the best solution, then proceed to Step 9.

Step 8: If a non-improving solution is generated, determine whether to accept this solution based on the simulated annealing criteria. If accepted, update the current solution; otherwise, maintain the original current solution, then proceed to Step 9.

Step 9: Update the scores of the destruction, repair, and swap operators in the operator score ledger.

Step 10: Check if the iteration number *i* modulo 30 equals zero; if so, consider using the Ortools solver's linear programming approach to solve for the optimal sub-route combination from a set of iteratively generated sub-route solutions and randomly generated sub-route solutions, set the newly obtained solution as the current solution and initialize the set of sub-route solutions; otherwise, return to Step 3.

The pseudocode for the Two-Stage Heuristic Algorithm is as follows:

IALNS Based on Linear Programming

```
Input： demand data, model parameters
Output: optimal solution
Cluster demand nodes using a spae-time distance-based k-center clustering
Construct initial solution S using a greedy algorithm
Set Sbest = S, Scur = S
Repeat
    Select a set of destruction, repair, and exchange operators (d, r, c) using
    the roulette wheel method
    Snew = c(r(d(s)))
    if accept (Snew , Scur) then
        Scur = Snew
        if f(Snew) < f(Sbest) then
            Sbest = Snew
        End if
    End if
    Store Snew in sub-loop combinations
    Update operator weights
    i = i +1
    if i % 30 = 0
        Use integer programming to solve the optimal sub-loop combinations and
        update neighborhood search Scur
        Initialize sub-loop combinations
    End if
Until the termination condition is met
Return Sbest
```

5. Case Study and Analysis of Computational Results

In this section, a case study of a dairy product cold chain distribution in Shanghai is presented to verify the feasibility and effectiveness of the proposed model and algorithms. A sensitivity analysis was conducted to explore the effects of average satisfaction changes at demand nodes with soft time windows and the ratio changes between demand nodes with soft and hard time windows on total costs. All experiments were carried out on a PC equipped with an Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz and 2.11GHz, with 16.0GB of RAM, using Python 3.9.

5.1. Parameter Settings

The case study is based on the distribution of dairy products in Shanghai. The geographical distribution of nodes, demand, and time window information are shown in Table 3, while vehicle-related parameters are presented in Tables 4 and 5. Other goal-related parameters are listed in Table 6.

Table 3. Node Information.

Node	Coordinates	Demand	Preferred Time Window	Acceptable Time Window	Service Time (min)
D1	121.519079,31.178838	0	—	—	0
S1	121.336189,31.318649	0	—	—	0
S2	121.361485,31.180489	0	—	—	0
S3	121.351671,31.01652	0	—	—	0
C1	121.375689,31.366831	0.6t	7:30-7:35	7:10-9:00	15
C2	121.455022,31.323576	0.4t	8:00-8:05	7:40-9:20	10
C3	121.427371,31.282269	0.4t	8:30-9:00	8:30-9:00	10
C4	121.232525,31.274027	0.8t	7:30-7:35	7:10-8:50	20
C5	121.464384,31.26076	0.4t	8:30-8:35	8:00-10:00	10
C6	121.351335,31.262449	0.8t	8:40-8:45	8:20-10:20	20
C7	121.391808,31.248767	0.8t	8:00-9:00	8:00-9:00	20
C8	121.422921,31.240309	0.6t	7:30-7:35	7:10-8:50	15
C9	121.33835,31.228173	0.6t	7:50-7:55	7:40-9:00	15
C10	121.371073,31.22393	0.4t	7:50-8:00	7:40-9:00	10
C11	121.201626,31.202598	0.6t	8:00-8:05	7:40-9:00	15
C12	121.437737,31.143082	0.4t	6:20-7:30	6:30-7:30	10
C13	121.391792,31.149528	0.6t	6:00-7:00	6:00-7:00	15
C14	121.447908,31.14241	0.6t	6:30-7:10	6:20-7:10	15
C15	121.318492,31.142219	0.6t	7:30-7:35	7:20-9:20	15
C16	121.28026,31.117884	0.4t	7:30-8:30	7:30-8:30	10
C17	121.320235,31.114561	0.6t	7:30-8:30	7:30-8:30	15

doi:10.20944/preprints202411.1470.v1

C18	121.413897,31.10436	0.6t	8:20-9:20	8:20-9:20	15
C19	121.510914,31.100481	0.6t	6:00-7:10	6:00-7:10	15
C20	121.589159,31.090836	0.8t	7:00-8:00	7:00-8:00	20
C21	121.411191,31.084089	0.4t	8:20-8:25	8:00-9:40	10
C22	121.347315,31.094539	0.8t	8:30-8:35	8:10-9:50	20
C23	121.270105,31.109855	0.6t	7:30-7:35	7:10-8:50	15
C24	121.228476,31.068138	0.6t	7:40-7:45	7:20-9:00	15
C25	121.456738,31.05216	0.4t	8:00-8:05	7:40-9:20	10
C26	121.24119,31.011107	0.8t	7:50-8:00	7:30-8:50	20
C27	121.420271,31.03132	0.4t	8:00-8:05	7:40-9:20	10
C28	121.518847,31.032595	0.4t	6:30-7:30	6:30-7:30	10
C29	121.440531,30.96813	0.6t	7:50-8:00	7:30-8:50	15
C30	121.492671,31.297665	0.6t	8:20-9:30	8:20-9:30	15

Table 4. Type 1 Vehicle Parameters.

Parameter	**Value**
fk_1	600 Yuan
v_1	40km/h
Q_1	8t
Quantity	4

Table 5. Type 2 Vehicle Parameters.

Parameter	**Value**
fk_2	250 Yuan
v_2	30km/h
Q_2	2.5t
Quantity	8

Table 6. Parameters Related to Objectives.

Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
θ_1	0.002	C_r	40 Yuan/h	ε_2	1 Yuan/h	a_2	1.3 Yuan/km
θ_2	0.003	β_1	15 Yuan/h	P	20000 Yuan/t	A	0.8
α_1	40 Yuan/h	β_2	20 Yuan/h	T	220km		
α_2	50 Yuan/h	ε_1	10 Yuan/h	a_1	3 Yuan/km		

5.2. Initial Solution Comparison

To verify the effectiveness of the first-stage algorithm of the proposed two-stage approach, the initial solutions obtained by our designed first-stage algorithm were compared with those from the 2-opt greedy algorithm and a basic greedy algorithm. The cost components of the distribution optimization scheme are shown in Table 7.

Table 7. Cost Comparison of First-Stage Algorithms.

Algorithm	Fixed	Transport	Refrigeration	Cargo Loss	Penalty	Total Cost	Satisfaction
Space-time k-center greedy	5350	743.80	311.97	2314.95	308.34	9029.06	96.21%
2-opt greedy	5600	734.76	308.36	2237.11	326.21	9206.44	93.06%
Greedy	5600	756.31	316.96	2354.53	457.29	9485.09	94.65%

From Table 7 that the initial total cost and satisfaction obtained by the spatial-temporal distance based K-center greed algorithm are superior to the other two algorithms, and the fixed cost related to the number of refrigerated trucks used is significantly reduced compared with the other two algorithms, which verifies the effectiveness of the spatial-temporal distance based K-center greed algorithm in planning the initial vehicle path.

5.3. Comparative Analysis of Second-Stage Algorithm Results

To validate the performance of the improved ALNS algorithm based on linear programming proposed in this paper, the solution performance of the improved ALNS algorithm was compared under three different first-stage algorithms. Among these algorithms, the improved ALNS algorithm based on linear programming is included. Table 8 displays these comparisons, where satisfaction refers to the average satisfaction at demand nodes with soft time windows. The convergence curves comparing the cost optimization of the improved ALNS algo-rithm and the linear programming-based improved ALNS algorithm on these initial solutions are depicted in Figures 4–6

Table 8. Comparative Results of Second-Stage Algorithms.

	Greedy		2-opt Greedy		Space-time Distance-based k-center Greedy	
Second-Stage Algorithms	**IALNS**	**Ortools-IALNS**	**IALNS**	**OrtoolsIALNS**	**IALNS**	**Ortools-IALNS**
Fixed	5600	5350	5600	5600	5350	5350
Transportation	729.99	723.55	719.41	730.37	717.03	712.11
Refrigeration	306.45	303.87	302.22	306.60	301.26	299.29
Cargo Loss	2318.19	2331.62	2243.49	2242.34	2296.72	2279.75
Penalty	429.09	370.67	259.45	236.44	327.19	324.13
Total	9383.72	9080.01	9124.57	9115.75	8992.20	8965.28
Satisfaction (%)	95.1128%	95.7367%	95.4544%	95.4551%	92.0028%	92.7767%

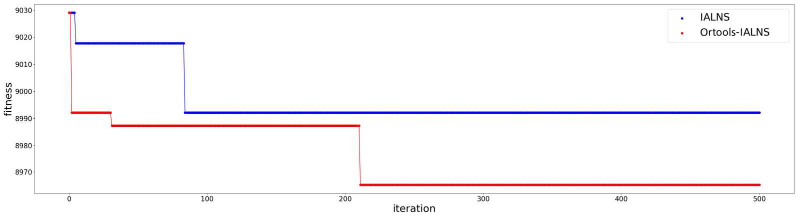

Figure 4. Space-time Distance-based k-center Greedy Algorithm.

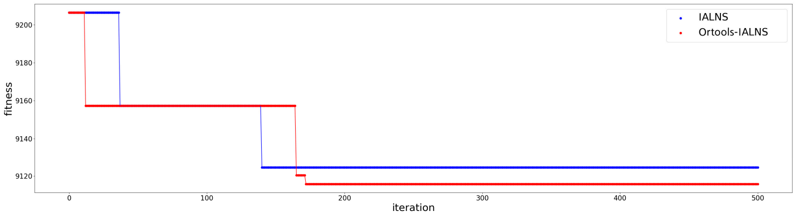

Figure 5. 2-opt Greedy Algorithm.

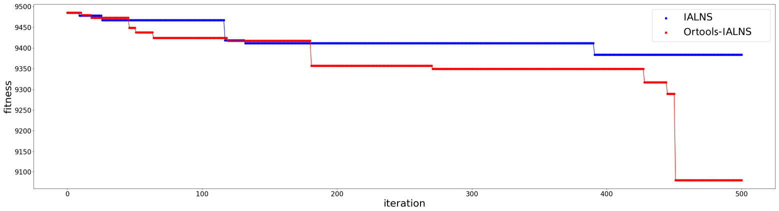

Figure 6. Greedy Algorithm.

The results from Table 8 illustrate that among the three two-stage heuristic algorithms, the improved ALNS algorithm based on linear programming consistently outperformed the standard improved ALNS algorithm in terms of cost reduction and satisfaction enhancement. This indicates a significant performance superiority of the linear programming-based approach. Moreover, it can be seen from the cost comparison convergence curve of the initial solution optimization of the improved ALNS algorithm and the improved ALNS algorithm based on linear programming that the solution result of the two-stage heuristic algorithm proposed in this paper is significantly better than the other two two-stage heuristic algorithms. Moreover, when the second-stage algorithm optimizes the path on the initial solution obtained by the K-center greedy algorithm based on space-time distance, Faster convergence.

5.4. Sensitivity Analysis

5.4.1. Impact of Changes in Average Satisfaction at Demand Nodes with Soft Time Windows

To examine the impact of changes in average satisfaction at demand nodes with soft time windows on costs, the proposed Two-Stage Heuristic Algorithm was employed to compute the costs under four levels of average satisfaction, as shown in Table 9.

Table 9. Results Under Different Satisfaction Levels.

Satisfaction Threshold	Fixed	Transportation	Refrigeration	Cargo Loss	Penalty	Total	Actual Satisfaction
≥60%	5350	810.08	338.48	2331.38	436.13	9266.07	87.55%

≥70%	5350	718.44	301.83	2295.18	484.11	9149.56	90.26%
≥80%	5350	712.11	299.29	2279.75	324.13	8965.28	92.78%
≥90%	5350	693.01	291.66	2229.40	374.18	8938.25	97.42%

The analysis reveals that increasing satisfaction generally leads to reductions in most cost categories and the overall costs. However, penalty costs may sometimes increase; this likely occurs because optimizing routes to enhance average satisfaction at soft time window demand nodes can inadvertently increase penalty costs at hard time window demand nodes, resulting in a net increase in overall penalty costs.

5.4.2. Effect of Varying Proportions of Demand Nodes with Soft and Hard Time Windows Across Different Data Scales

In this section, the impact of different ratios of demand nodes with soft and hard time windows on routing solutions is explored. This section aims to further validate the efficacy of the improved ALNS algorithm based on linear programming introduced in this pape. We consider three data scales—30, 40, and 50 total demand nodes—and two scenarios of hard time window demand nodes that can be directly served in a one-echelon distribution system, with quantities of 3 and 6. The cost comparisons in the tables are defined by the total number of demand nodes minus the ratio of hard time window to soft time window demand nodes, minus the number of directly served hard time window demand nodes. In these tables, 'C' represents the total cost objective, and 'GAP1/%' indicates the percentage reduction in cost achieved by the improved ALNS based on linear programming relative to the standard improved ALNS. The term "GAP2%" refers to the relative error in the optimal solutions obtained from ten computations with using Ortools-IALNS , under different proportions of demand nodes with soft and hard time windows, compared to the optimal solution when the ratio of hard to soft time window demand nodes is 1:1.The algorithms were executed ten times to obtain the optimal solution results, as shown in Tables 10–12.

Table 10. Cost Comparison for a Data Scale of 30 Demand Nodes.

Data Scale	IALNS	Ortools-IALNS	GAP1/%	GAP2/%
	C	C		
30-1:1-6	9612.22	9232.40	3.95%	—
30-0.3:0.7-6	8949.70	8910.91	0.43%	-3.48%
30-0.4:0.6-6	8992.20	8965.28	0.30%	-2.89%
30-0.6:0.4-6	9285.10	9269.87	0.16%	0.41%
30-0.7:0.3-6	10026.69	9655.68	3.7%	4.58%
30-1:1-3	8395.13	8329.34	0.78%	—
30-0.3:0.7-3	8350.61	8310.43	0.48%	-0.23%
30-0.4:0.6-3	8342.02	8319.95	0.26%	-0.11%
30-0.6:0.4-3	8380.33	8361.29	0.22%	0.38%
30-0.7:0.3-3	9754.01	8910.84	8.64%	6.98%

Table 11. Cost Comparison for a Data Scale of 40 Demand Nodes.

Data Scale	IALNS	Ortools-IALNS	GAP1/%	GAP2/%
	C	C		
40-1:1-6	11442.61	11206.45	2.06%	—
40-0.3:0.7-6	11168.27	10979.78	1.69%	-2.02%
40-0.4:0.6-6	11327.20	11106.72	1.95%	-0.89%

40-0.6:0.4-6	11561.03	11237.28	2.80%	0.28%
40-0.7:0.3-6	11495.71	11288.33	1.80%	0.73%
40-1:1-3	10120.82	10039.85	0.80%	—
40-0.3:0.7-3	10036.88	9920.34	1.16%	-1.19%
40-0.4:0.6-3	10037.74	10026.90	0.11%	-0.13%
40-0.6:0.4-3	10218.74	10139.85	0.77%	0.99%
40-0.7:0.3-3	11073.03	10973.10	0.90%	9.30%

Table 12. Cost Comparison for a Data Scale of 50 Demand Nodes.

Data Scale	IALNS	Ortools-IALNS	GAP1/%	GAP2/%
	C	C		
50-1:1-6	13479.45	13135.93	2.55%	—
50-0.3:0.7-6	13118.99	12658.86	3.51%	-3.63%
50-0.4:0.6-6	13259.92	12698.76	4.23%	-3.33%
50-0.6:0.4-6	13776.39	13150.06	4.55%	0.11%
50-0.7:0.3-6	13218.68	13165.01	0.41%	0.22%
50-1:1-3	13290.33	12221.38	8.04%	—
50-0.3:0.7-3	11577.62	11338.74	2.06%	-7.22%
50-0.4:0.6-3	11849.68	11659.99	1.60%	-4.59%
50-0.6:0.4-3	14195.33	12843.36	9.52%	5.09%
50-0.7:0.3-3	13617.69	13045.75	4.20%	6.75%

The results from Tables 10, 11, and 12 demonstrate that the improved ALNS algorithm based on linear programming consistently delivers superior solution quality across data scales of 30, 40, and 50 demand nodes when compared to the standard improved ALNS algorithm. Furthermore, as the data scale increases, the optimization effectiveness enhances, corroborating the robust performance of the algorithm proposed in this paper. Additionally, within the same data scale, an increase in the number of hard time window demand nodes directly served in a one-echelon system correlates with higher costs. Similarly, under the same ratio of soft to hard time window demand nodes, greater numbers of directly served hard time window demand nodes result in elevated costs.

6. Conclusions

The demand for fresh food is expanding, making the timely distribution of perishable goods a significant societal issue. This paper explores the vehicle routing problem for cold chain joint distribution from a single supply point through multiple distribution centers to various demand nodes, categorized into two types, within one-echelon and two-echelon networks. An optimization model aimed at minimizing the total cost while considering the time window satisfaction of these two types of demand nodes was proposed. For this model, a two-stage heuristic algorithm was designed and solved. Case study analysis has demonstrated that the algorithm proposed in this paper converges more quickly and with greater precision. Additionally, sensitivity analyses were conducted on the impact of changes in the average satisfaction of demand nodes within soft time windows on costs, and the effect of varying the ratio of soft and hard time window demand nodes on the routing scheme. The results indicate that increasing the average satisfaction of soft time window demand nodes generally leads to a reduction in overall costs, although penalty costs may increase. Furthermore, an increase in the proportion of hard time window demand nodes tends to raise the total costs, especially when the number of direct deliveries to hard time window nodes increases. The findings of this paper provide practical insights for cold chain logistics enterprises. Considering appropriate proportions of direct deliveries in one-echelon networks and the balance of soft and hard time window ratios in two-echelon networks can potentially reduce overall distribution costs. Although effective, the model presented has limitations; it only considers a single type of good, whereas actual logistics involve multiple types of goods and potential cooperation among distribution centers. Future work will aim to improve the model by addressing these limitations, creating a more realistic model for perishable goods distribution in one-echelon and two-echelon networks, and designing more efficient algorithms to solve it.

Author Contributions: Conceptualization, Manqiong Sun and Yang Xu; Data curation, Feng Xiao, Bing Su and Fei Bu; Formal analysis, Hao Ji; Investigation, Feng Xiao, Hao Ji, Bing Su and Fei Bu; Methodology, Manqiong Sun, Feng Xiao and Bing Su; Supervision, Yang Xu; Validation, Yang Xu; Visualization, Fei Bu; Writing – original draft, Manqiong Sun and Yang Xu; Writing – review & editing, Manqiong Sun, Yang Xu and Feng Xiao.

Funding: This paper is supported by the Shaanxi Provincial Natural Science Basic Research Program for Youth, (2024JC-YBQN-0737).

Data Availability Statement: Not applicable.

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

References

1. Miao, Xiaohong, Sujuan Pan, and Lei Chen. Optimization of perishable agricul tural products logistics distribution path based on IACO-time window constraint[J]. Intelligent Systems with Applications 20 (2023): 200282.
2. El Raoui H, Oudani M, Pelta D A, et al. A metaheuristic-based approach for the customer-centric perishable food distribution problem[J]. Electronics, 2021, 10(16): 2018.
3. Shi Hao, He Junsheng, and Ma Chang. (2013). An improved genetic algorithm for solving vehicle routing problems with both soft and hard time windows. Journal of Shandong Jiaotong University, 21(2), 14-18.
4. Dantzig G B, Ramser J H. The truck dispatching problem[J]. Management science,1959, 6(1): 80-91.
5. Utama D M, Dewi S K, Wahid A, et al. The vehicle routing problem for perishable goods: A systematic review[J]. Cogent Engineering, 2020, 7(1): 1816148.
6. Tarantilis C D, Kiranoudis C T. A meta-heuristic algorithm for the efficient distribution of perishable foods[J]. Journal of food Engineering, 2001, 50(1): 1-9.
7. Qin Gaoyuan, Tao Fengming, Li Lixia. A vehicle routing optimization problem for cold chain logistics considering customer satisfaction and carbon emissions[J]. International journal of environmental research and publichealth, 2019, 16(4): 576.
8. Agrawal A K, Yadav S, Gupta A A, et al. A genetic algorithm model for optimizing vehicle routing problems with perishable products under time-window and quality requirements[J]. Decision Analytics Journal, 2022, 5: 100139.
9. Zulvia F E, Kuo R J, Nugroho D Y. A many-objective gradient evolution algorithm for solving a green vehicle routing problem with time windows and time dependency for perishable products[J]. Journal of Cleaner Production, 2020, 242: 118428.
10. Li Qian, Jiang Li, and Liang Changyong. (2021). Multi-objective optimization of cold chain distribution based on fuzzy time windows. Computer Engineering and Applications, 57(23), 255-262.
11. Barma P S, Dutta J, Mukherjee A, et al. A multi-objective ring star vehicle routing problem for perishable items[J]. Journal of Ambient Intelligence and Humanized Computing, 2022: 1-26.
12. Huai C X, Sun G H, Qu R R, et al. Vehicle routing problem with multi-type vehicles in the cold chain logistics system[C].2019 16th International Conference on Service Systems and Service Management (ICSSSM). IEEE, 2019: 1-4.
13. Wang Ziqi, Wen Peihan. Optimization of a low-carbon two-echelon heterogeneous-fleet vehicle routing for cold chain logistics under mixed time window[J]. Sustainability, 2020, 12(5): 1967.
14. Sahraeian R, Esmaeili M. A multi-objective two-echelon capacitated vehicle routing problem for perishable products[J]. Journal of Industrial and Systems Engineering, 2018, 11(2): 62-84.
15. Jaigirdar S M, Das S, Chowdhury A R, et al. Multi-objective multi-echelon distribution planning for perishable goods supply chain: A case study[J]. International Journal of Systems Science: Operations & Logistics, 2023, 10(1): 2020367.
16. Ma Yanfang, Li Baoyu, Yang Yifu, et al. (2021). Research on two-echelon routing problems and algorithms for fresh food distribution with customer classification. Computer Engineering and Applications, 57(20), 287-298.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.