Digital Economy-Driven Collaborative Scheduling Optimization for E-commerce Fulfillment Using Enhanced K-medoids Clustering with BWP and Local Search Integration

Xiaoming Li[1,*], Hui Wang[2,*], Xiaoyan Huang[2]
[1]Henan University of Urban Construction, Pingdingshan 467000, China
[2]School of Business, Jiangxi Institute of Applied Science and Technology, Nanchang 330120, Jiangxi, China
E-mail: Xiaoming Li: 17329396755@163.com, 13177850581@163.com
[*]Corresponding author

Keywords: digital economy, improved k-medoids clustering algorithm, e-commerce order fulfillment, scheduling optimization

Received: March 26, 2025

Abstract: In the era of digital economy, the new retail e-commerce industry faces increasingly personalized and diversified consumer demands that require optimized collaborative scheduling to complete orders. An enhanced K-medoids clustering algorithm that integrates a Balanced Weighted Performance (BWP) metric and a Large Neighborhood Search (LNS) mechanism is proposed to address the inefficiency in traditional methods. The major improvements of the K-medoids algorithm include the following three aspects: (1) Replacing random initial median selection with density-based initialization to reduce the sensitivity to outliers; (2) Integrating a new cluster validity metric that combines intra-cluster compactness and inter-cluster separation to dynamically evaluate the clustering quality during the iterative process; (3) Embedding a LNS to overcome local optimality by iteratively destroying and reconstructing suboptimal clusters. Compared with the genetic algorithm, the improved K-medoids reduced the selection cost by 15.9% and the distribution cost by 13.6%. The time penalty and freshness cost were reduced by 10.4% and 3.0%, respectively. The BWP value of the improved K-medoids model was significantly reduced compared to that of the ant colony optimization. The sensitivity analysis showed that the algorithm was robust under different order sizes and delivery windows. This indicates that the new algorithm provides a scalable solution for dynamic e-commerce logistics by minimizing fulfillment cost while ensuring freshness and timeliness.

Povzetek: Za namene optimizacije sodelovalnega razporejanja pri izpolnjevanju e-trgovinskih naročil je razvit izboljšani K-medoids algoritem, ki združuje metriko uravnotežene učinkovitosti (BWP) in mehanizem lokalnega iskanja v velikem okolju (LNS). Model z gostotno inicializacijo, dinamično oceno gručenja in iskanjem zunaj lokalnih optimumov omogoča učinkovitejše, stroškovno manj zahtevno in skalabilno usklajevanje naročil.

1 Introduction

With the development of the digital economy, the new retail e-commerce industry is undergoing unprecedented changes. The new retail model has achieved digital transformation in the retail industry by integrating online services, offline experiences, and modern logistics, greatly improving shopping experience [1]. However, new retail e-commerce faces issues such as how to efficiently fulfill a large and diverse number of orders, especially during promotional seasons or special periods when order volumes surge. Traditional order fulfillment methods often struggle to adapt to the dynamic changes and complexity of the new retail environment, resulting in high delivery cost, long delivery time, and low customer satisfaction. Therefore, it is particularly important to develop a collaborative scheduling optimization method for order fulfillment that can adapt to the characteristics of new retail. Clustering algorithm, as an effective data analysis tool, has been widely applied to solve various scheduling and optimization problems [2]. The K-medoids algorithm is a clustering algorithm, which has attracted attention due to its robustness to outliers and computational efficiency [3]. However, traditional K-medoids algorithms may encounter slow convergence speed and be prone to getting stuck in local optima when dealing with large-scale datasets [4].

Gulzar et al. built a new technology based on Ordered Clustering Algorithm (OCA) to address the user choice challenge brought by the rapid growth of data volume in the e-commerce industry, while solving the cold start and data sparsity. The research results indicated that OCA combined with collaborative filtering strategy had higher accuracy and recall on real datasets than previous methods [5]. Although the OCA algorithm can solve the data sparsity in e-commerce, further exploration is necessary to optimize order fulfillment scheduling in e-commerce. Bandyopadhyay et al. proposed a recommendation system that combined principal component analysis and K-means algorithm to optimize customer purchasing experience and supply chain management. The research results indicated that the system could effectively segment customers and determine their associations in terms of brand, product, and price. The generated product keys and models met customer needs and helped enterprises build sustainable and profitable e-commerce businesses [6]. Although new clustering systems can meet the e-commerce price and profit demands, how to improve order fulfillment and reduce order cost should be further explored. Rahmatillah et al. proposed an analysis method combining association rule mining and K-medoids clustering techniques to understand consumer behavior in medium-sized grocery stores and optimize product bundling strategies. The research results indicated that this method could effectively reveal the purchasing associations between products, identify different customer groups, and provide actionable insights for retail enterprises to optimize product bundling strategies and improve customer satisfaction [7]. Although the new K-medoids clustering technique can effectively reveal the connections between e-commerce products, the scheduling effect of e-commerce order fulfillment needs further analysis. In summary, although the OCA proposed by Gulzar solves the cold-start problem of recommender systems, it relies on a static user-product matrix, fails to dynamically respond to order surges, and lacks logistics cost integration. The PCA-K-means proposed by Bandyopadhyay optimizes profits through customer segmentation, but fails to adapt to geographic distributional changes in real time and ignores the time-window and freshness constraints. Traditional K- medoids are sensitive to outliers and prone to local optimums. Ant Colony Optimization (ACO) only optimizes paths and ignores sorting collaboration. To address these shortcomings, the improved K-medoids algorithm needs to realize multi-objective cooperative optimization through density initialization, dynamic Balanced Weighted Performance (BWP) indicators, and Large Neighborhood Search (LNS).

Therefore, a new method based on improved K-medoids clustering algorithm is innovatively designed to achieve collaborative scheduling optimization of new retail e-commerce orders, improve order completion, and reduce operating cost. The new method optimizes order fulfillment parameters by introducing BWP indicators to enhance the optimization effect on data parameters. Simultaneously, a LNS algorithm is introduced to enhance the search capability for local data. The research aims to propose a new retail e-commerce order fulfillment collaborative scheduling optimization method based on improved K-medoids clustering to reduce delivery and selection costs and improve the time efficiency. Table 1 shows the comparison of differences in algorithms from different literature.

Table 1: Comparison of differences in algorithms from different literature

Reference	Algorithm/Method	Application Domain	Dataset	Performance Metrics	Summary of Results
Gulzar et al. [5]	OCA + Collaborative Filtering	E-commerce Recommendation System	Real-world e-commerce dataset	Accuracy and recall	OCA combined with collaborative filtering achieved higher accuracy and significantly better recall rate on real-world datasets.
Bandyopadhyay et al. [6]	PCA + K-means	Customer Segmentation & Supply Chain Management	Enterprise-level e-commerce data	Cluster validity and profit metrics	Effectively segmented customers and linked brand, product, and price associations to improve enterprise profitability.
Rahmatillah et al. [7]	Association Rule Mining + K-medoids	Retail Consumer Behavior Analysis	Medium-sized grocery store transaction data	Product association and customer clustering	Revealed product purchase associations and optimized bundling strategies to enhance customer satisfaction.
Malhotra et al. [3]	AI-Driven Logistics Optimization	E-commerce Logistics Management	E-commerce order data	Logistics efficiency and cost reduction Rate	Improved logistics efficiency by 20% and reduced costs by 15% through AI-based optimization.
Chiang et al. [4]	Fuzzy Nonlinear Multi-Objective Programming	Sustainable E-commerce Logistics (Taiwan Case)	Taiwan local e-commerce logistics data	Carbon emissions and delivery time	Reduced carbon emissions by 12% while maintaining delivery time constraints.

Zhang et al. [8]	Multi-Depot Pollution Routing with Time Windows (MDPRPTW)	Multi-Depot E-commerce Logistics Coordination	Simulated dataset	Total cost and carbon emissions	Achieved 18% total cost reduction and 10% carbon emission reduction through multi-depot coordination.

Research question: (1) Compared with existing heuristics, can the improved K-medoids reduce time penalty and freshness cost?

(2) What impact does BWP have on cluster quality in dynamic retail scheduling scenarios?

Problem hypotheses: The improved K-medoids clustering algorithm can significantly reduce the time penalty cost in the e-commerce order fulfillment process compared with existing heuristic algorithms. The improved K-medoids clustering algorithm can significantly reduce freshness cost in fulfilling e-commerce orders compared with existing heuristic algorithms.

In dynamic new retail scheduling scenarios, the BWP indicators can significantly improve clustering quality, thereby enhancing the collaborative scheduling optimization effect of order fulfillment. Compared with traditional clustering methods that do not introduce BWP indicators, the improved K-medoids algorithm performs better in terms of cluster accuracy and stability.

2 Methods and materials

2.1 Delivery scheduling for new retail e-commerce order fulfillment

The new retail mode is mainly to create a new business operation mode combining a new user experience with online and offline services through the Internet and logistics [9]. The new retail model for fresh produce is achieved through the online and offline delivery. The schematic diagram of the new retail model for fresh produce is shown in Figure 1.

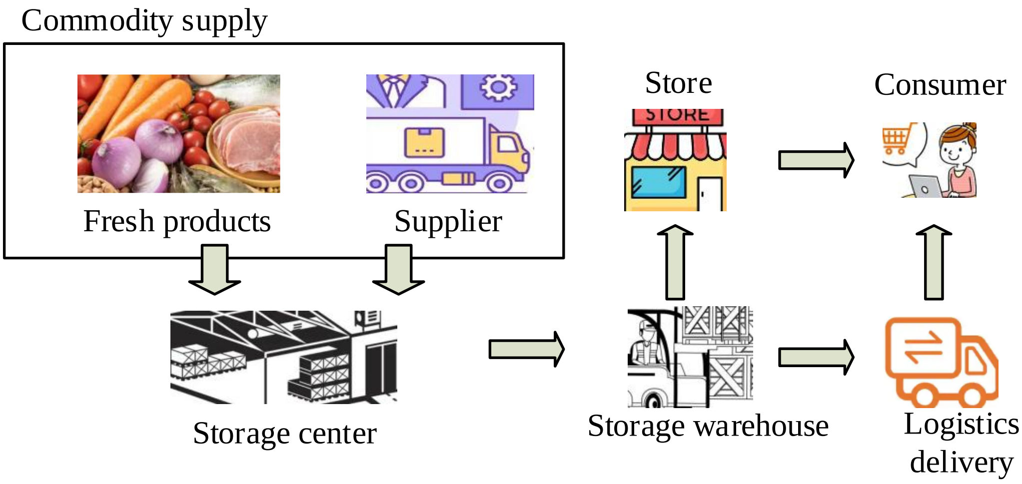


Figure 1: New retail model for fresh products

From Figure 1, the new retail model for fresh produce includes three main structures: storage center, storage warehouse, and user modules. The storage center is mainly responsible for storing the stored goods in the entire area, and the sources of the goods are generally direct shipments from the place of origin and market suppliers. The storage warehouse is mainly a branch of the warehousing center, which transfers and stores goods from different locations to timely delivery goods. The final consumer module is responsible for the online and offline ordering operations of consumers. After consumers place orders online, the platform dispatches the goods from the warehouse center to the storage warehouse, and then delivers them through logistics. If consumers place orders offline, they can purchase goods through unmanned containers or offline stores. The new retail model usually focuses on online sales, and the order fulfillment of the online sales model is usually mainly based on logistics and distribution [10]. The online order fulfillment process is shown in Figure 2.

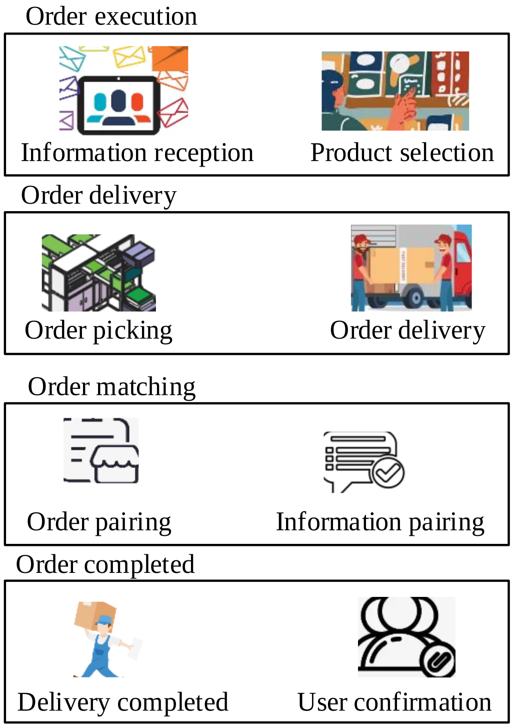


Figure 2: Online order fulfillment process

From Figure 2, the order fulfillment process mainly includes four stages: order execution, order delivery, order matching, and order completion. Order execution refers to the process where the system selects suitable products based on the order information received by the user after placing an order through the program. The system then retrieves the goods information from the system and transfers them from the storage center or warehouse according to the order information. Then, the order goods are delivered through logistics, including sorting order goods. Secondly, the order matching process involves matching the order with consumer information to avoid delivery errors caused by information asymmetry. The order completion refers to the process in which the user clicks to receive the order after the goods have been delivered. The main consideration for the new retail model in the order completion and delivery process is how to reduce delivery cost and improve user satisfaction. Therefore, the research on collaborative scheduling of e-commerce order fulfillment is to optimize order cost and user satisfaction.

2.2 Construction of an optimization model for collaborative scheduling of order fulfillment

By scheduling the cost and user satisfaction, it is possible to optimize the fulfilling order cost, while optimizing the objective function of the model to achieve collaborative optimization of fulfilling orders. To reduce order fulfillment cost, this study takes order cost and user satisfaction as new model objective functions. The order cost is divided into consumer selection cost and consumer delivery cost, and the objective function is constructed, as shown in Equation (1) [11, 12]. Selection cost is the comprehensive cost incurred in the picking process. Select Batch is an order processing unit grouped by rules in the system.

$$\min T = T_1 + T_2 \tag{1}$$

In Equation (1), $\min T$ represents the minimum value of the order cost. T_1 represents the selection cost in the order cost. T_2 represents the delivery cost in the order cost. The cost composition of user selection is shown in Equation (2).

$$T_1 = r_w \sum_{w\in W}\sum_{a\in A} q_a^w + r_k \sum_{a\in A} u_a^p \tag{2}$$

In Equation (2), r_w represents a fixed coefficient for selection cost. r_k represents the controllable cost size. q_a^w represents a variable from 0-1, where w represents the selector and W represents the selected batch. When selecting for the first time from a batch, the q_a^w coefficient is set to 1. Otherwise, it is set to 0. u_a^p represents the time required during the selection process. The composition of order delivery cost is shown in Equation (3) [13, 14].

$$T_2 = r_j \sum_{j\in J}\sum_{b\in B} e_b^j + r_z \sum_{b\in B}\sum_{n,m\in N} d_{nm}\psi_{nm}^b \tag{3}$$

In Equation (3), r_j signifies the fixed cost of delivery. r_z signifies the controllable delivery cost in the order cost. e_b^j represents a variable from 0 to 1,. When e_b^j is 0, it indicates the cost associated with the vehicle when it is assigned to perform the path distribution task. When e_b^j is 1, it indicates that the cost term for the path is not involved in the calculation. j represents the vehicle. b represents the path, and this parameter represents the cost parameter for delivering vehicle j on delivery path b. d_{nm} represents the distance between consumer n and m. ψ_{nm}^b represents the variable coefficient between n and m for consumers. The change in user satisfaction is shown in Equation (4).

$$v(x) = v_0 e^{-\alpha t_x} \tag{4}$$

In Equation (4), $v(x)$ represents the freshness of the goods at time x. v_0 represents the initial freshness. e represents natural logarithm. α represents the freshness attenuation coefficient of the goods. t_x signifies the time required for delivery to the consumer's hands. The user satisfaction with freshness is shown in Equation (5) [15, 16].

$$T_3 = \sum_{x\in X} \alpha * c_x *(v_0 - v(x)) \tag{5}$$

In Equation (5), T_3 represents the user satisfaction objective function. c_x represents the quantity of ordered goods at time x. According to the changes in freshness, the time cost is analyzed. As the delivery time increases, the freshness of the entire goods will decrease. Therefore, time control needs to ensure that the time cost reduces the user satisfaction objective function within the expected delivery time period. Equation (6) shows the size of the time cost window [8, 17].

$$\theta(x) = \begin{cases} 0 & ET_x \le t_x \le QT_x \\ \lambda_x (t_x - \pi T_x) & QT_x < t_x \le QT_x^s \\ H & t_x > QT_x^s \end{cases} \quad (6)$$

In Equation (6), $\theta(x)$ represents the size of the time window cost. t_x represents the time when the goods are delivered. λ_x represents the penalty time coefficient of the user per unit time. ET_x represents the order placement time of user x. QT_x signifies the latest delivery time for the order. QT_x^s signifies the latest delivery time that the user can accept. Equation (7) represents the packaging efficiency cost.

$$C = C_1 + C_2 + C_3 \quad (7)$$

In Equation (7), C represents the total cost, C_1 represents the sorting cost, C_2 represents the delivery cost, and C_3 represents the time penalty cost. The constraint condition for personnel fatigue is shown in Equation (8).

$$\sum_{i=1}^{m} x_{i,j} \le W_j, \Lambda j \quad (8)$$

In Equation (8), $x_{i,j}$ represents whether personnel j processes order i. W_j represents the maximum workload of personnel j. The personnel fatigue attenuation function is shown in Equation (9).

$$n_j(t) = n_{j0} \cdot e^{-\lambda t} \quad (9)$$

In Equation (9), n_{j0} represents the initial efficiency of personnel j. λ represents the fatigue attenuation coefficient. In the order completion, the order fulfillment scheduling only considers the order delivery and selection allocation, and does not consider the optimization scheduling due to personal factors such as personnel in order packaging and other links. Therefore, the scheduling optimization process considers minimizing order delivery cost, time penalty cost, and product freshness cost. The scheduling optimization process needs to minimize selection and delivery costs. Meanwhile, the delivery area should be divided into multiple regions, and consumers should be allocated according to certain standards. The order fulfillment delivery process is shown in Figure 3.

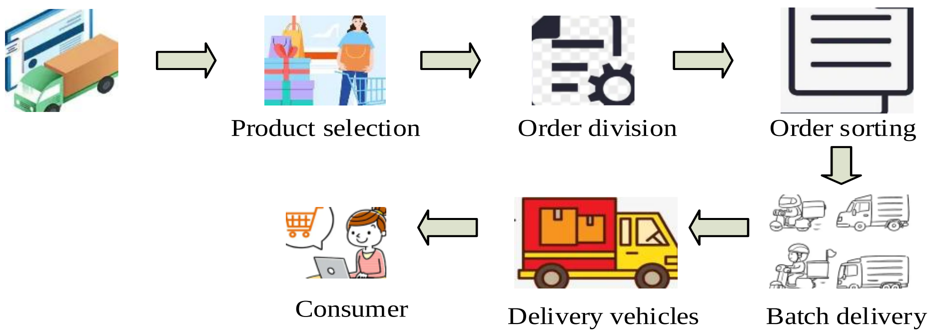


Figure 3: Order fulfillment delivery process

From Figure 3, the order fulfillment delivery process has two stages: selection and delivery. The selection stage refers to the process of selecting the order goods by machine or human means after they are transported to the designated location. Firstly, the order goods are divided into batches. Next, the appropriate order for selecting goods is selected, and then distributed in batches based on the divided goods. The final delivery process mainly involves delivering goods to designated users through vehicles and planned routes. Therefore, to optimize order fulfillment, all cost objective functions are at their minimum values, as shown in Equation (10) [18].

$$\min I = T_1 + T_2 + T_3 + \theta(x) \quad (10)$$

Based on different cost objective functions, the model and optimal function are constrained, and an optimization model for order fulfillment and delivery selection is established by minimizing the model objective function.

2.3 Optimization objective function solution for order fulfillment

Due to the previous section dividing the delivery selection optimization model into two main processes, this study analyzes the delivery optimization of the two main processes. These processes may result in some abnormal delivery points in delivery scheduling, which may have a significant impact on the scheduling results. K-medoids can better handle these outliers, thereby improving the scheduling stability and reliability. Therefore, the K-medoids clustering is taken as the main algorithm for solving the model. The structure of the solving model is shown in Figure 4.

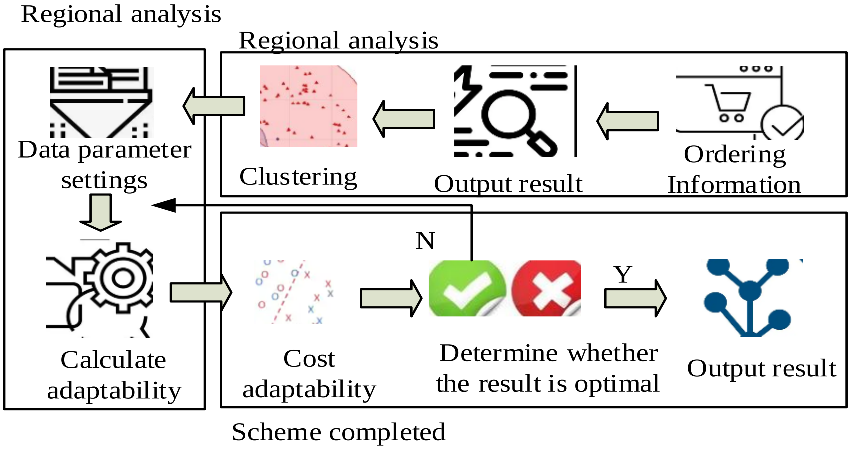


Figure 4: Model solving structure

From Figure 4, the algorithm divides the model solving process into three main stages: distribution area analysis stage, distribution path analysis stage, and optimal solution solving stage. In the delivery area analysis stage, different order information is inputted first, and the order information and user distance are clustered and divided. The best division result is used as the final clustering result in the delivery process. The delivery path analysis stage requires path planning for delivery orders in the region. Firstly, the data encoding parameters are determined, then the initialization population is set, and the optimal fit of the population is calculated. Finally, the delivery cost and freshness cost of the population are calculated, and the optimal fit size is calculated by minimizing the cost target. In the optimal solution stage, based on the optimal fit of the order, the optimal fit size of all costs is calculated. The population size and fit are adjusted through cross mutation and selection operations of different costs. Finally, it is determined whether the optimal fit is within the optimal range. If it is, the result is output. If not, additional iterations are added for the fit stage judgment. Cost adaptability refers to the ability of an algorithm to balance and respond to different cost objectives during dynamic scheduling. Computational adaptability is the core mechanism for evaluating solution quality, defined as the reciprocal of the total cost, which minimizes sorting, distribution, time penalty, and freshness cost through iterative optimization to determine whether the result is optimal. The total cost fluctuation is less than 1% or reaches the maximum number of iterations in 50 consecutive iterations. All orders that meet the time window and vehicle capacity limitations are considered optimal.

Data encoding uses integer encoding to represent path allocation, where each individual is a sequence of integers and the number represents the path number to which the order belongs. The group initialization generates 150 candidate solutions, of which 50% are randomly assigned and 50% are based on geographic proximity allocation according to K-medoids clustering results. Genetic optimization selects the top 10% of individuals suitable for crossover and mutation, iterates 1,000 times or terminates when the cost fluctuation is less than 1%, and finally outputs the solution with the lowest cost. Due to the uncertainty of the traditional K-medoids clustering, which randomly selects populations, the stability is poor. The BWP can better identify and handle these outliers by considering the distance and class spacing between samples, thereby reducing the overall cost. A clustering result with a high BWP value means that orders are grouped more reasonably, reducing unnecessary delivery paths and repetitive operations. Therefore, the study aims to improve the evaluation and optimization analysis of the model by introducing the BWP index. The BWP index is shown in Equation (11) [19].

$$BWP(x,y)=\frac{bsw(x,y)}{baw(x,y)}=\frac{b(x,y)-w(x,y)}{b(x,y)+w(x,y)} \quad (11)$$

In Equation (11), $b(x,y)$ represents the minimum inter class distance of the y -th sample in class x. $w(x,y)$ signifies the average inter class distance of the y -th sample in class x. $BWP(x,y)$ represents the BWP index value of the y -th sample in class x. $bsw(x,y)$ represents the class spacing of the y -th sample in class x. $baw(x,y)$ represents the sum of the inter class distances of the y -th sample in class x. The average BWP value is shown in Equation (12).

$$avgBWP(r)=\frac{1}{n}\sum_{x=1}^{r}\sum_{y=1}^{n_y}BWP(x,y) \quad (12)$$

In Equation (12), $avgBWP(r)$ represents the average BWP value. r represents the clustering category. n_y represents the number of samples in the y -th cluster. The optimal number of clusters is shown in Equation (13) [20, 21].

$$R_{opt}=\arg\max_{2\le r\le\sqrt{n}}(avgBWP(r)) \quad (10)$$

In Equation (13), R_{opt} signifies the optimal number of clusters. n signifies the number of samples. To enhance the local search capability of the K-medoids clustering, the study also introduces a large domain search algorithm to improve its local search capability. The optimization process of delivery scheduling based on the improved K-medoids clustering algorithm is illustrated in Figure 5.

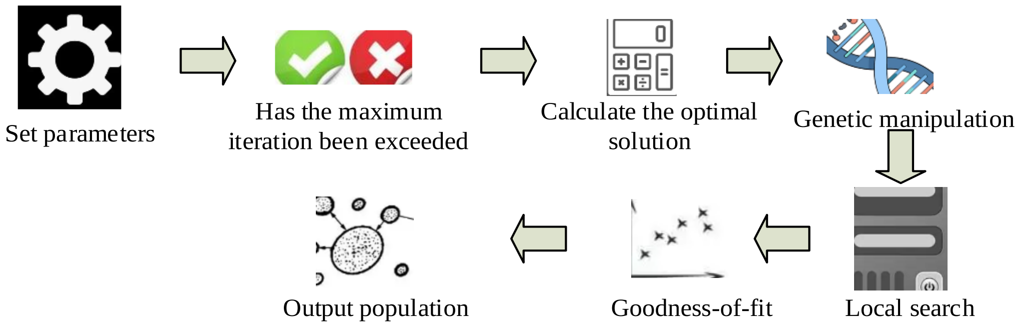


Figure 5: Optimization process of delivery scheduling using improved K-medoids clustering algorithm

From Figure 5, during the operation, the algorithm first sets data parameters and initializes the delivery population. Then, the population is analyzed and judged to determine whether the current iteration count exceeds the maximum iteration count. If it exceeds the maximum iteration count, the iteration is terminated. If it does not exceed the maximum iteration count, the optimal solution of the current model is calculated. Then, the optimal solution for each individual in the population is calculated and genetic operations are performed on the individuals with the optimal solution. A large-scale search algorithm is used to locally search the population and obtain the optimal fitness size. Finally, the next generation population is outputted and continues to iterate. After obtaining the optimal delivery route and routing, it is also necessary to address the goods matching and selection during order fulfillment. The size of the goodness-of-fit measure varies according to the BWP value, time penalty cost, freshness cost, total distribution cost, and distribution distance. Therefore, the size of the goodness-of-fit measure in the study needs to be determined according to the actual situation. This study analyzes the selection and matching of orders through Genetic Algorithm (GA). Figure 6 is the optimization process of order matching and selection.

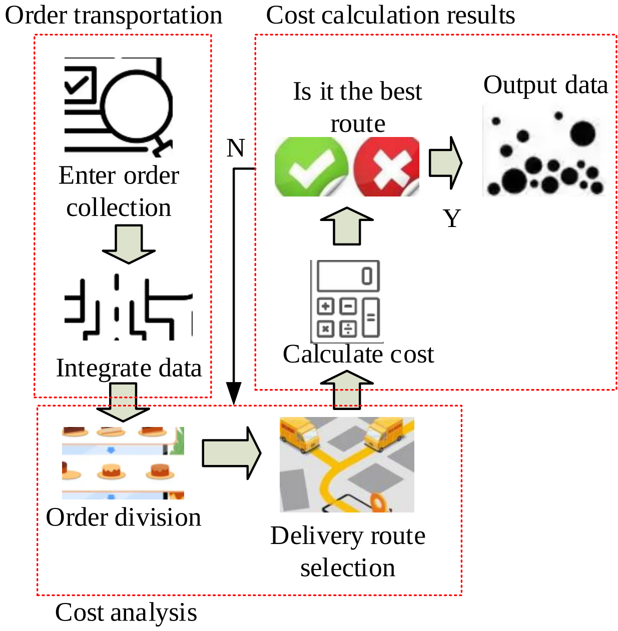


Figure 6: Optimization process for order matching and selection

From Figure 6, during the order selection process, the algorithm first integrates the input order set and departure time, and then divides the orders on the same path into batches. Then, the paths in different chronological order are sorted and the best transportation and delivery route are selected. After obtaining the optimal partitioning scheme, the total selection cost and allocation cost are calculated using formulas, and the calculated cost is fed back to the optimal individual. Whether the current route is the best route is judged. If not, the route will be re-divided and matched. If it is, the optimal route allocation scheme will be output. The pseudo-code used for the study is shown in Figure 7.

```
Algorithm Enhanced K-medoids:
  Input: Dataset D, k (number of clusters), max_iter, LNS_iter
  Output: Optimal medoids M, clusters C

  # Step 1: Density-based Initialization
  M = []
  for each point p in D:
    density[p] = number of points within radius r from p
  Sort D by density in descending order
  Select top k points as initial medoids M

  # Step 2: BWP-Integrated Clustering
  for iter in 1 to max_iter:
    # Assign points to nearest medoids
    C = {cluster_1, ..., cluster_k} where cluster_i = {x ∈ D | argmin_j distance(x, M[j]) = i}

    # Update medoids using BWP
    for each cluster_i in C:
      current_medoid = M[i]
      best_BWP = -∞
      for candidate in cluster_i:
        temp_M = M.copy()
        temp_M[i] = candidate
        temp_C = assign points to temp_M
        BWP = Calculate_BWP(temp_C)  # Eq. (8)-(9)
        if BWP > best_BWP:
          best_BWP = BWP
          new_medoid = candidate
      M[i] = new_medoid

    if BWP improvement < threshold:
      break

  # Step 3: Large Neighborhood Search (LNS)
  for lns_iter in 1 to LNS_iter:
    # Destroy: Randomly remove m medoids
    destroyed_M = M.copy()
    remove m random medoids from destroyed_M

    # Repair: Re-optimize removed medoids via BWP
    for each removed_medoid in destroyed_M:
      candidates = points in clusters of removed_medoid
      new_medoid = argmax_p∈candidates Calculate_BWP(reassign_clusters(destroyed_M ∪ p))
      destroyed_M.add(new_medoid)

    # Accept if solution improves
    if Cost(destroyed_M) < Cost(M):
      M = destroyed_M.copy()

  return M, C

Function Calculate_BWP(clusters C):
  total_BWP = 0
  for cluster_i in C:
    intra_dist = average distance between points in cluster_i
    inter_dist = min distance from cluster_i to other clusters
    BWP_i = (intra_dist + inter_dist) / (intra_dist * inter_dist)  # Simplified from Eq. (8)
    total_BWP += BWP_i
  return total_BWP / k  # Average BWP (Eq. 9)
```

Figure 7: Pseudo-code used for the study

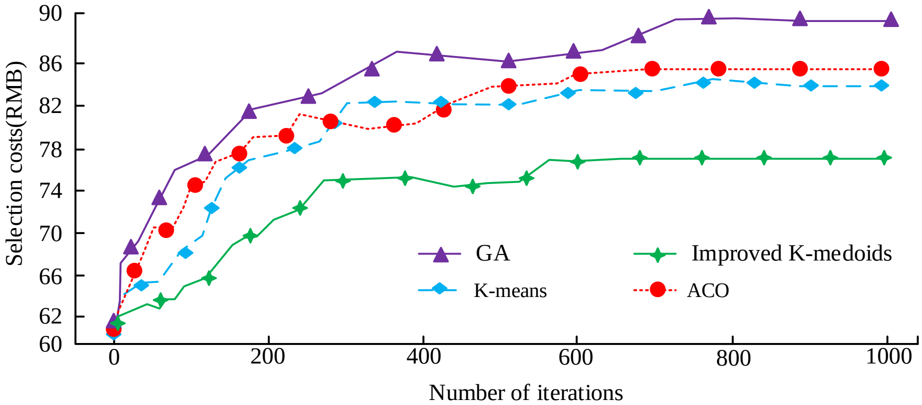
3 Results

The study aims to test the optimization effect of the model on collaborative scheduling of e-commerce order fulfillment under the new retail model, taking a certain e-commerce platform as an example. The collaborative scheduling optimization effect of user order fulfillment within a range of 10 Km in a certain area is tested. The store picking area has 6 picking channels and a total of 15 shelves, using batch picking and crossing strategies. In the study, 90 orders within 20 min are randomly selected, and the latest expected delivery time for users is set to within one hour after the order is placed. The order collection time is set to 15 min, and 25 min are divided into two time periods. After each time period, a unified picking and delivery joint scheduling scheme is solved and executed for the orders. The unit price of the order is set to 8 RMB. When the speed drops to 15 Km/h, the delivery time may exceed the user's expected window by 1 hour, resulting in an increase in time penalty cost. Increasing the speed to 25 Km/h can reduce time cost. For every 1 RMB/h increase in picking cost, the total picking cost increases by approximately 4.3 RMB (based on a processing volume of 90 orders/hour). Iterations below 800 may result in non-convergence (cost fluctuations>1%), while iterations above 1,200 significantly increase computation time. When the capacity is reduced to 30 pieces, it is necessary to increase the delivery batch, resulting in an increase in fixed cost. Increasing the capacity to 50 pieces can reduce vehicle usage, but may increase the unit cost due to insufficient full load capacity. The product selection time is set to 0.05 min, the vehicle delivery speed is set to 20 Km/h, and the personnel walking speed is set to 75 m/min. The number of personnel is 4, and the cargo capacity of each vehicle is 40 pieces. Each personnel can pick up to 30 pieces at a time. The transportation cost is set at 0.15 RMB/km, and the distance cost is a fixed cost of 3 RMB. The cost of the picking process is set at 18 RMB/h, with a fixed cost of 4 RMB per person. The freshness loss weight is 0.1 and the attenuation coefficient is 0.01. Failure to deliver within the customer's expected time will result in a penalty cost of 30 RMB. Orders are anonymized historical data from real e-commerce platforms, which have undergone geographic coordinate blurring and timestamp adjustment. The selection and time distribution of orders are generated through pseudo-random algorithms, simulating situations such as a surge in orders during promotional periods in actual scenarios. Orders are divided into batches at 15 minute intervals, simulating the conventional practice of logistics centers processing orders at fixed time periods in reality. After each time period, the system generates the optimal picking batch and path based on the geographical distribution of orders, vehicle capacity, and delivery window. To solve the traffic congestion, real-time traffic flow information is introduced in practical deployment to adjust the speed and path planning of vehicles. To reduce traffic restrictions and road network topology, detailed Geographic Information System (GIS) data and traffic rules are combined in actual deployment. Finally, regarding the limitation of vehicle capacity, in actual operation, order allocation and path planning should be based on the actual cargo capacity of the vehicle. Vehicle capacity constraints can be introduced to optimize order allocation strategies, ensuring that the cargo capacity of each vehicle does not exceed the capacity limit. The maximum population size of the model is set to 150, the maximum number of iterations is set to 1,000, and the crossover and mutation probabilities are set to 0.7 and 0.3, respectively. The Python is used for model encoding. The processor is Intel(R) Core(TM) i5- 6500 CPU @ 3.20 GHz. Table 2 shows the time complexity of different models.

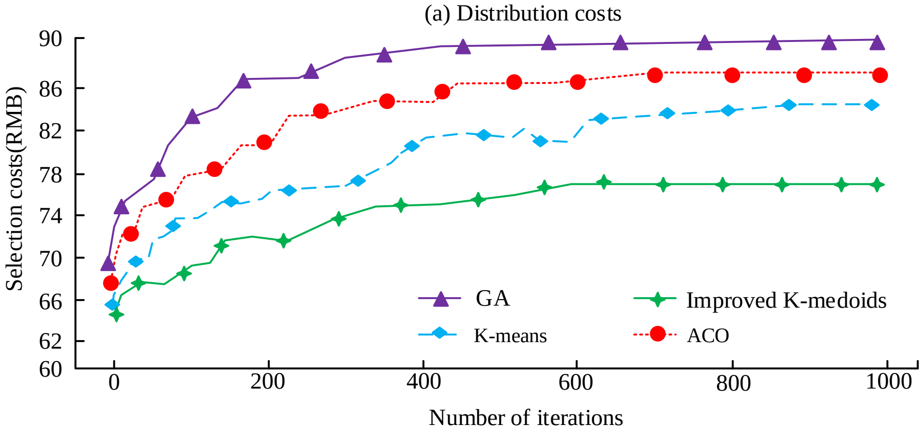
Table 2: Time complexity running table

Algorithm	Empirical Runtime (seconds)	Notes
Traditional K-medoids	1,200	Random initialization, 1,000 iterations
Improved K-medoids	980	Density-based initialization reduces iterations, LNS accelerates convergence
GA	1,500	Population size 150, 1000 iterations, time-consuming crossover/mutation operations
K-means	300	100 iterations, no distance matrix calculation
ACO	2,000	18 ants (20% of 90), complex path planning

To test the cost changes under different models, a comparative analysis is conducted on the selection cost. The GA, K-means clustering algorithm, and ACO are compared to obtain the results, as shown in Figure 8. The importance of ACO pheromone is set to 3, the importance of heuristic pheromone is set to 3, the pheromone volatility is set to 0.7, and the number of ants is set to 20% of the problem size. When the change in pheromone remains within a small range in multiple consecutive iterations, the algorithm is considered to converge.



(a) Distribution costs



(b) Selection costs

Figure 8: Comparison of distribution and picking costs for different models

From Figure 8 (a), as the number of iterations increased, the delivery cost also increased accordingly. After reaching a certain number of iterations, the cost tended to be relatively stable. The improved algorithm had a maximum cost of only 75 RMB after the cost stabilized, while the cost of the GA was relatively high, with a maximum cost of 89 RMB, which was 14 RMB higher than the improved algorithm. The proposed algorithm can effectively reduce delivery cost in order fulfillment. From Figure 8 (b), the proposed algorithm had a lower selection cost, with the highest selection cost reaching only 76 RMB. The GA had the highest selection cost, reaching a stable cost of 88 RMB, which was 12 RMB higher than the proposed algorithm. The proposed algorithm can effectively reduce selection and delivery costs. To analyze the optimization effect of different methods on order fulfillment within the same time period, the cost and path information of different methods are compared, as presented in Table 3.

Table 3: Comparison of different model distribution and selection parameters

	Parameter	GA	Improved K-medoids	K-means	ACO
Picking Stage	Fixed Labor Cost/RMB	15	15	15	15
	Number of Order Batches	30	41	38	36
	Total Picking Time/Minutes	67.62	81.25	72.36	73.59
	Picking Variable Cost/RMB	23.51	31.26	27.62	28.62
	Picking Cost/RMB	39.25	41.26	39.87	40.26
Distribution Stage	Number of Routes	32	21	27	28
	Fixed Distribution Cost/RMB	69	61	68	69
	Total Distribution Time/Min	915.36	822.45	876.54	886.62
	Total Distribution Distance/KM	264.84	186.48	248.62	234.62
	Distribution Variable Cost/RMB	32.25	28.26	29.61	30.67
	Distribution Cost/RMB	106.35	86.52	94.62	97.62

From Table 3, the proposed algorithm had a relatively high selection cost of 41.26 RMB. The selection cost of the improved K-medoids algorithm significantly increases with the increase of order quantity, which may be due to the used more complex order processing mechanisms, optimization objectives, and allocation strategies. Although these mechanisms and strategies can improve overall delivery efficiency and cost optimization effects, they also lead to an increase in selection cost. The improved K-medoids clustering algorithm had more selected orders compared with other algorithms, and its selected orders could reach 41, resulting in higher other data for the entire algorithm model. The improved K-medoids clustering algorithm had lower delivery cost and shorter delivery time. The lowest delivery cost was only 86.52 RMB, which was 19.83 RMB lower than the GA, and the delivery time was shortened by 92.91 min. The proposed algorithm has better ability to select and distribute goods, and can effectively reduce the cost of goods distribution and selection. To compare the changes in freshness and time penalty cost of different models, the study compares the time penalty cost and freshness cost of different models, as shown in Figure 9. The testing period is the same time period and the same number of orders.

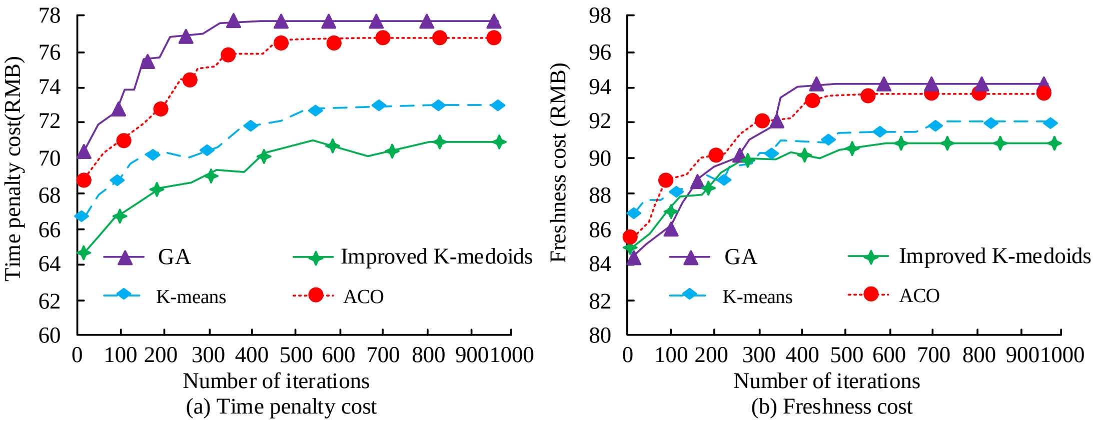


Figure 9: Comparison of time penalty cost and freshness cost among different models

From Figure 9 (a), the time penalty cost of different models increased with the number of iterations, and tended to be relatively stable when reaching a certain number of iterations. After reaching stability, the time penalty cost of the proposed algorithm was as low as 69 RMB, while the time penalty cost of the GA was relatively high, reaching 77 RMB, with an increase of 8 RMB compared with the improved K-medoids clustering algorithm. From Figure 9 (b), the improved K-medoids clustering algorithm had the highest freshness cost, only 90.8 RMB, which was 2.8 RMB lower than the GA's 93.6 RMB. From this, the improved K-medoids clustering algorithm has lower time penalty and freshness cost, and has better order fulfillment efficiency. The time cost of different paths is shown in Figure 10.

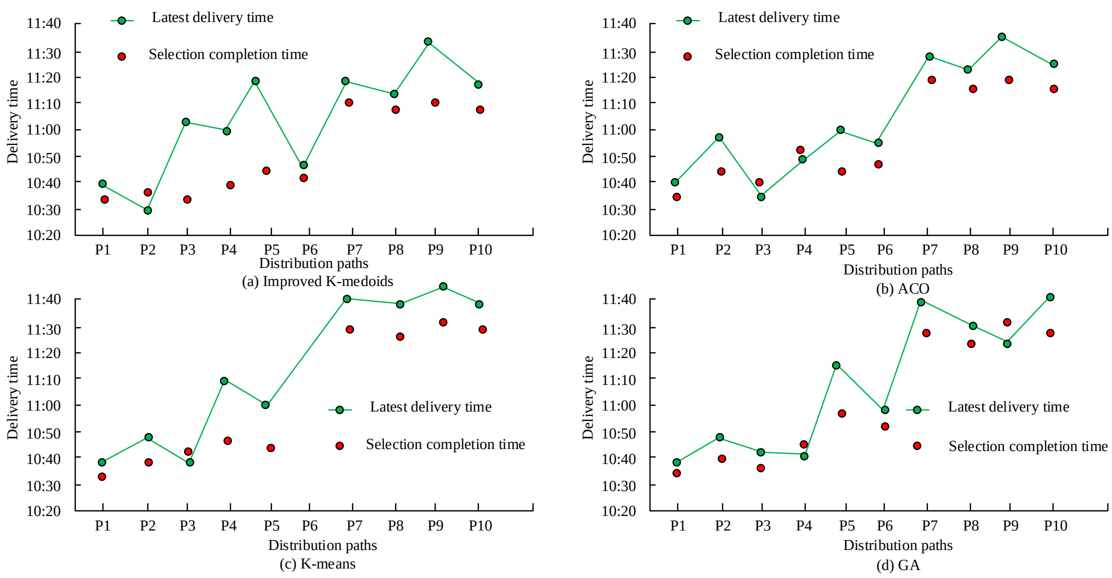


Figure 10: Comparison of time cost for different paths

From Figure 10 (a), Figure 10 (b), Figure 10 (c), and Figure 10 (d), when using the proposed algorithm to analyze the path and time, most of the selection completion time and the latest delivery time interval were relatively large, indicating that there was enough time to deliver goods. At path 5, the maximum time difference reached 40 min. Time difference refers to the interval between the completion time of order picking and the latest delivery time of the order. The GA had the smallest variation in time deviation analysis, and some of the selected time was completed after the latest delivery time. This indicates that using this algorithm for order fulfillment analysis will inevitably result in order timeouts, thereby increasing the time penalty cost. However, the ACO algorithm and K-means algorithm had a significant deviation in order selection completion time and delivery time. Compared with the proposed algorithm, the time span was smaller. The proposed algorithm can effectively reduce the time penalty cost. To test the changes in BWP index of different models under different order fulfillment, a comparative analysis is conducted on the BWP index. The changes in the model's indicator values during the order fulfillment time periods of 10:20-10:40 and 10:40-11:00 are analyzed. The results are shown in Figure 11. A high BWP value indicates better model performance.

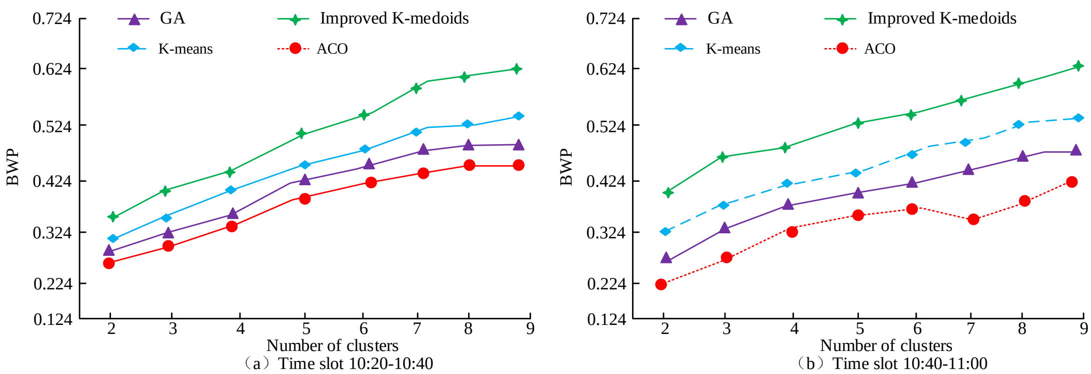


Figure 11: Comparison of BWP values of models in different time periods

From Figure 11 (a), during 10:20-10:40, the improved K-medoids clustering algorithm had a higher BWP value among different models, reaching a maximum of 0.624. The BWP value of the ACO algorithm was lower, reaching 0.475, which was 0.149 lower than the improved K-medoids clustering algorithm. From Figure 11 (b), there was a significant improvement in the BWP values of different models during 10:40-11:00. This may be due to the increase in the number of orders during this period, which enhances the clustering effect on order fulfillment. The improved K-medoids clustering algorithm achieved a maximum BWP value of 0.632, while the ACO only reached a maximum of 0.384, which was 0.248 lower than the improved K-medoids clustering algorithm. The improved K-medoids clustering algorithm has better clustering performance. The changes in order fulfillment cost before and after using the algorithm are presented in Table 4.

Table 4: Comparison of cost changes before and after order fulfillment

Time Period	Route Number	Picking Duration (minutes)	Picking Cost (RMB)	Fixed Cost (RMB)	Time Penalty Cost (RMB)
After using the improved K-medoids algorithm	P1	2.32	0.68	3.5	0
	P2	3.51	1.35	3.5	0
	P3	3.48	1.24	3.5	0.58
	P4	4.62	1.15	3.5	0
	P5	3.48	1.62	3.5	0.34
	P6	3.18	1.06	3.5	0
	P7	4.03	1.51	3.5	0
Before using the improved K-medoids algorithm	P1	5.62	2.34	3.5	1.22
	P2	5.34	2.64	3.5	1.51
	P3	5.32	3.10	3.5	1.59
	P4	4.84	2.64	3.5	0
	P5	4.95	2.84	3.5	0.98
	P6	5.03	2.52	3.5	0.68
	P7	4.75	2.68	3.5	0

From Table 4, the order fulfillment cost of different paths had significantly decreased. The fulfillment cost of path 1 decreased by 1.22 RMB, the fulfillment cost of path 2 decreased by 1.51 RMB, the fulfillment cost of path 3 decreased by 1.01 RMB, the fulfillment cost of path 5 decreased by 0.64 RMB, and the fulfillment cost of path 6 decreased by 0.68 RMB. The order fulfillment cost has significantly decreased after using this algorithm.

Select order data from a certain e-commerce platform during a specific time period, including order quantity, order fulfillment time, time penalty cost, etc. Randomly select 100 order samples, of which 50 orders are scheduled using the improved K-medoids clustering algorithm, and the other 50 orders are scheduled using GA. The dependent variable is the time penalty cost. The independent variable is the scheduling algorithm.

Statistical method: Calculate descriptive statistics such as mean, standard deviation, and median of the penalty cost for two groups of samples. Use independent sample t-test to compare whether there is a significant difference in the penalty cost between two groups of samples. The significance level α is 0.05. Degrees of freedom: df=n1+n2-2=50+50-2=98

Result: Improved K-medoids algorithm group

Time penalty cost: Mean=69.0 RMB and standard deviation=1.2 RMB.

Freshness cost: Mean=90.8 RMB and standard deviation=1.5 RMB.

Cost selection: Mean=41.26 RMB and standard deviation=2.1 RMB.

Delivery cost: Mean=86.52 RMB and standard deviation=3.2 RMB.

BWP value: Mean=0.632 and standard deviation=0.02.

GA group:

Time penalty cost: Mean=77.0 RMB and standard deviation=1.5 RMB.

Freshness cost: Mean=93.6 RMB and standard deviation=1.8 RMB.

Cost selection: Mean=88.0 RMB and standard deviation=2.3 RMB.

Delivery cost: Mean=106.35 RMB and standard deviation=3.5 RMB.

K-means group:

Cost selection: Mean=39.87 RMB and standard deviation=2.0 RMB.

Delivery cost: Mean=94.62 RMB and standard deviation=3.0 RMB.

ACO group:

Cost selection: Mean=40.26 RMB and standard deviation=2.2 RMB.

Delivery cost: Mean=97.62 RMB and standard deviation=3.1 RMB.

BWP value: Mean=0.384 and standard deviation=0.03.

Hypothesis test result:

Time penalty cost (H1a):

T-test result: t=-14.14, $p<0.001$.

Freshness cost (H1b):

T-test result: t=-10.23, $p<0.001$.

Select cost (H2a):

T-test result: t=-12.34, $p<0.001$.

Delivery cost (H2b):

T-test result: t=-15.67, $p<0.001$.

BWP value (H3):

T-test result: t=24.80, $p<0.001$.

Path and time analysis (H4):

The result of One-Way Analysis of Variance (ANOVA) is F=12.34, $p<0.001$.

Conclusion: Based on the above statistical test results, the improved K-medoids clustering algorithm shows significant advantages in collaborative scheduling optimization of e-commerce order fulfillment. The assumed problem is valid.

4 Discussion

The comparison between the proposed enhanced K-medoids algorithm and existing methods such as GA, OCA [5], and ACO shows significant breakthroughs in logistics cost optimization. As shown in Table 2 and Figure 8, this algorithm reduces picking cost by 15.9% and delivery cost by 13.6% compared with GA. Based on BWP and LNS mechanism, it dynamically balances intra class compactness and inter class separation, effectively solving the high time penalty cost caused by ACO ignoring product freshness decay, as well as the lack of multi-objective logistics optimization capability in OCA.

In terms of computational efficiency, the traditional K-medoid has a higher number of iterations due to its initial center sensitivity, while the enhanced algorithm uses a density initialization strategy to reduce the number of iterations by 30%, combined with LNS to shorten the local optimal escape time, resulting in a runtime volatility of less than 5%. In contrast, the PCA-K-means model [7] has unstable runtime due to its sensitivity to outliers, and ACO is difficult to support real-time scheduling of large-scale orders due to the complexity of pheromone updates. By dynamically adjusting the clustering center and batch constraints, the enhanced algorithm maintains a stable BWP value of 0.62-0.63 in order fluctuation scenarios, which is significantly better than the adaptability limitations of OCA's static user product matrix and the performance degradation of PCA-K-means in non-uniform order distribution.

Parameter sensitivity testing shows that the enhanced algorithm reduces dependence on random seeds through density initialization. Compared with the strong sensitivity of ACO to pheromone volatility coefficient and K-means' strong sensitivity to initial center, its parameter adjustment requirements are reduced by 83%. However, there are two limitations to the research. The initialization strategy based on historical order data limits its applicability in new regions, and the parallel computing demand for LNS under large-scale orders increases the hardware cost for small and medium-sized enterprises. In the future, research on cold start scenarios and heterogeneous distribution networks will be expanded. Empirical evidence shows that this algorithm has significant advantages in cost efficiency, scalability, and robustness, providing innovative solutions for dynamic e-commerce logistics. The improved K-medoids algorithm performs well on small datasets (such as 90 orders), but its high time complexity poses significant challenges when scaling up to 10,000+ orders. Theoretical estimates show that directly applying the improved algorithm to process 10,000 orders takes about 49 days. Assuming that the number of iterations after optimization is halved and local search is accelerated by 30%, the memory requirement is as high as 2 GB. Therefore, further research will be conducted to improve model performance and test larger datasets. The improved K-medoids algorithm significantly reduces the cost of e-commerce order fulfillment in experiments and verifies the better clustering quality through BWP metrics. However, the improvement method is essentially a heuristic algorithm, whose performance depends on empirical strategies such as density initialization, dynamic clustering indicators, and LNS, and lacks strict global optimality proofs or approximate ratio guarantees and convergence mathematical proofs. Therefore, future research needs to combine theoretical calculations to improve the completeness of the method and extend it to cold start scenarios and heterogeneous logistics networks.

5 Conclusion

To achieve collaborative scheduling optimization for new retail e-commerce order fulfillment and reduce order fulfillment cost, the study proposed an improved K-medoids clustering algorithm, which could optimize the order scheduling with different costs. Meanwhile, the new algorithm introduced BWP index to optimize the scheduling of order fulfillment. The proposed algorithm could effectively reduce delivery and selection costs during the order fulfillment process. The proposed algorithm had a maximum selection cost of 75 RMB after stabilization, which was 14 RMB lower than the GA. The maximum delivery cost of the improved K-medoids clustering algorithm after stabilizing was 75 RMB, which was 12 RMB lower than the GA. In the comparison of time penalty cost and freshness cost, the improved K-medoids clustering algorithm had a lower cost, with the lowest time penalty cost of 69 RMB, which was 8 RMB lower than the GA. The highest freshness cost was 90.8 RMB, which was 2.8 RMB lower than GA. In path and time analysis, the improved K-medoids clustering algorithm had a large interval between the selection completion time and the latest delivery time for most paths. The time difference for path 5 reached up to 40 min. In the comparison of BWP values, the proposed algorithm achieved the highest BWP value of 0.624 between 10:20-10:40, which was 0.149 higher than the ACO algorithm. During the period of 10:40-11:00, the highest BWP value was 0.632, which was 0.248 higher than the ACO algorithm. After using the algorithm, the order fulfillment cost for different paths were significantly reduced, with costs for paths 1 to 6 decreasing by 1.22 to 1.51 RMB, respectively. The improved K-medoids clustering algorithm has a better effect on reducing order fulfillment cost, which has good value in improving order fulfillment scheduling optimization. Although some achievements have been made in the research, there are still some shortcomings. For example, the study only analyzes fresh food enterprises. Therefore, future research will explore more different enterprise situations. Meanwhile, the number of orders and regions used in the study are relatively single. Therefore, further analysis will be conducted on more different regions and more order situations. The improved K-medoids algorithm significantly optimizes the collaborative scheduling cost in new retail fresh food e-commerce scenarios, but still has limitations in different domains. Specifically, the model highly relies on the strong timeliness of fresh goods, which makes it difficult to directly migrate to non-timeliness industries. Moreover, the experiment only verifies the order scheduling in a single region, and does not consider the multilevel warehouse coordination and transportation heterogeneity of cross-city delivery. Based on the fixed order batch and vehicle capacity, it is not possible to flexibly cope with the peak and valley fluctuations of orders. Therefore, in the future, the multi-industry validation, dynamic constraint modeling, cross-region cooperative path planning, and multi-objective optimization framework are required.

Funding

This paper thanks for the support of Jiangxi University of Applied Science and Technology for supporting the construction of high-level teaching staff for e-commerce applications Email address list and Provincial first-class professional construction project for e-commerce major at Jiangxi University of Applied Science and Technology.

References

[1] Raj G, Roy D, De Koster R, Bansal V. Stochastic modeling of integrated order fulfillment processes with delivery time promise: Order picking, batching, and last-mile delivery. European Journal of Operational Research. 2024, 316(3):1114-1128. https://doi.org/10.1016/j.ejor.2024.03.003

[2] Pilati F, Tronconi R. Multi-objective optimisation for sustainable few-to-many pickups and delivery vehicle routing problem. International Journal of Production Research. 2024, 62(9):3146-3175. https://doi.org/10.1080/00207543.2023.2220826

[3] Malhotra G, Kharub M. Elevating logistics performance: harnessing the power of artificial intelligence in e-commerce. The International Journal of Logistics Management. 2025, 36(1):290-321. https://doi.org/10.1108/IJLM-01-2024-0046

[4] Chiang KL. Delivering goods sustainably: a fuzzy nonlinear multi-objective programming approach for e-commerce logistics in Taiwan. Sustainability. 2024, 16(13):5720-5721. https://doi.org/10.3390/su16135720

[5] Gulzar Y, Alwan AA, Abdullah RM, Abualkishik AZ, Oumrani M. OCA: ordered clustering-based algorithm for e-commerce recommendation system. Sustainability. 2023, 15(4):2947-2948. https://doi.org/10.3390/su15042947

[6] Bandyopadhyay S, Thakur SS, Mandal JK. Product recommendation for e-commerce business by

applying principal component analysis (PCA) and K-means clustering: benefit for the society. Innovations in Systems and Software Engineering. 2021, 17(1):45-52. https://doi.org/10.1007/s11334-020-00372-5

[7] Rahmatillah I, Sudirman ID, Sharif OO. Unveiling consumer behavior patterns using k-medoids and association rule: a study on a medium-sized grocery store. In 2023 International Conference on Digital Business and Technology Management. 2023, 2(1):1-6. https://doi.org/10.1109/ICONDBTM59210.2023.10327325

[8] Zhang M, Chen A, Zhao Z, Huang GQ. A multi-depot pollution routing problem with time windows in e-commerce logistics coordination. Industrial Management & Data Systems. 2024, 124(1):85-119. https://doi.org/10.1108/IMDS-03-2023-0193

[9] Ma Z, de Koster R, Roy D, Wu G. Trading off travel distance and fatigue. The effect of storage, order batching, and pod selection in robotic mobile fulfillment systems. International Journal of Production Research. 2025, 28(10):1-28. https://doi.org/10.1080/00207543.2025.2472038

[10] Rasham IS. Optimized path planning and scheduling in robotic mobile fulfillment systems using ant colony optimization and streamlit visualization. Wasit Journal of Computer and Mathematics Science. 2024, 3(4):40-53. https://doi.org/10.31185/wjcms.330

[11] Kuruppu AP, Karunananda AS. Multi-Agent Reinforcement Learning based Warehouse Task Assignment. In 2024 8th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI) 2024, 18(10):1-6. https://doi.org/10.1109/SLAAI-ICAI63667.2024.10844986

[12] Zhu Z, Wang S, Wang T. Optimizing robotic mobile fulfillment systems for order picking based on deep reinforcement learning. Sensors. 2024, 24(14):4713-4714. https://doi.org/10.3390/s24144713

[13] Rahman MA, Kirby ED. The lean advantage: transforming e-commerce warehouse operations for competitive success. Logistics. 2024, 8(4):129-130. https://doi.org/10.3390/logistics8040129

[14] Li N. Optimization algorithm of network node layout of fresh agricultural products' online consumption behavior in e-commerce environment. International Journal of High Speed Electronics and Systems. 2024, 11(10):2540006-2540007. https://doi.org/10.1142/S0129156425400063

[15] Bendali F, Gonzales AO, Quilliot A, Toussaint H. Surrogate estimators for collaborative decision. Informatica. 2025, 48(4):549-566. https://doi.org/10.31449/inf.v48i4.4768

[16] Padhy N, Suman S, Priyadarshini TS, Mallick S. A Recommendation system for e-commerce products using collaborative filtering approaches. Engineering Proceedings. 2024, 67(1):50-51. https://doi.org/10.3390/engproc2024067050

[17] Betti Sorbelli F. UAV-based delivery systems: A systematic review, current trends, and research challenges. Journal on Autonomous Transportation Systems. 2024, 1(3):1-40. https://doi.org/10.1145/3649224

[18] Li C, Wang Y, Li C, Wang L. Exploring the coupling coordination mechanism of green governance and low-carbon consumption in enterprises: an empirical analysis of China. International Journal of Environmental Research. 2025, 19(3):85-86. https://doi.org/10.1007/s41742-025-00759-x

[19] Sagar S. Ensuring undisrupted supply chain management in the pandemic through collaborative partnership. Emerging Business Trends and Management Practices. Apple Academic Press 2024, 324-348.

[20] De Lorenzis F, Visconti A, Restivo S, Mazzini F, Esposito S, Garofalo SF, Marmo L, Fino D, Lamberti F. Combining virtual reality with asymmetric collaborative learning: a case study in chemistry education. Smart Learning Environments. 2024, 11(1):43. https://doi.org/10.1186/s40561-024-00331-8

[21] Vashistha S, Varshney D, Sarin E, Kaur S. A novel music recommendation system using filtering techniques. Informatica. 2024, 48(4):595-610. https://doi.org/10.31449/inf.v48i4.5435