

Research Article

Algorithms for Picking and Distribution of Online Orders in New Retail Enterprises

Weiya Zhong  and Jia Cui 

School of Management, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Weiya Zhong; wyzhong@i.shu.edu.cn

Received 7 May 2021; Revised 6 August 2021; Accepted 15 September 2021; Published 9 November 2021

Academic Editor: Yandong He

Copyright © 2021 Weiya Zhong and Jia Cui. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper studies joint algorithms of order picking and distribution in new retail enterprises. The problem will consider many factors, such as the type of goods, picking time, batch capacity of distribution, distribution time, and distribution cost. First of all, the research problems are summarized as mathematical programming problems. Then, a genetic algorithm and comparison algorithms are proposed. Finally, the rationality of the model and the effectiveness of the algorithms are verified by computational experiments, and management enlightenments are revealed.

1. Introduction

In the first quarter of 2020, communities in China were closed for management, and since then, the public has been buying fresh food and daily necessities online because of the outbreak of COVID-19. The number of customers in traditional stores has reduced, and orders from new fresh e-commerce have increased rapidly. The new mode of door-to-door delivery has become popular. “It’s not the alarm clock that wakes me up every day, but the desire for fresh vegetables.” After the outbreak of the epidemic, many people living at home chose to stay away from the grocery market temporarily and “embrace” the Internet, so “online shopping” has become the main choice for supplying daily necessities. This has led to the rise of various online fresh food retail platform merchants against the trend, with the number of users, orders, and sales reaching a new height. According to public information, Dingdong (an APP for online fresh food shopping) saw a 291.42% increase in the number of new users and an overall volume growth of approximately 80%; the order volume of Meituan (an APP for online fresh food shopping) increased by more than 200% in some sites in Shanghai; and the number for HEMA (an APP for online fresh food shopping) was 290%. While the business volume of each online retail platform is booming, the supply chain

including logistics and distribution is also under great pressure.

On January 15, 2016, HEMA opened its first store in Shanghai Jinqiao Square and quickly became an “Internet celebrity.” HEMA is a new retail format completely reconstructed by Alibaba for offline supermarkets that is different from other fresh e-commerce. It is a supermarket, a restaurant, and a vegetable market. However, such descriptions seem to be inaccurate. Consumers can buy goods in-store or place orders through HEMA APP. This feature is also known as “warehouse-store-integration,” and HEMA’s warehouse is also its store. Compared to traditional warehouse, the interior merchandise is placed differently. The traditional warehouse stores goods in the standard shelf-stacking pattern. However, each customer can see all goods in HEMA. The display and placement of commercial products are different according to the preferences of consumers in the area where the store is located.

By observing online orders’ picking process of HEMA, we found that once a customer places an order in HEMA APP, the information processing center will divide the order into several suborders according to commodity types, such as river fresh, fruit, pasta, and daily necessities. (*This is determined by the goods location and attributes of the commodity. For example, fresh river food products need to be*

dehydrated and processed. It must also be sealed and packed to prevent contamination of other goods.) The information of each suborder is first sent to the handheld device of the staff in the area where the goods are located. The staff selects goods and puts them into a bag. The bag is hung on the hook at the starting position of a conveyor belt, which transfers the bag to the packaging area. In the packaging area, items from the same order are assembled together and passed on to the deliverymen, who distribute customers' orders in batches. One of the biggest characteristics of HEMA is fast delivery: within 3 kilometers around the store, it takes only 30 minutes to deliver goods to your door.

From the above description, we can see that, in the order processing system, the flow of information drives the flow of goods: receiving information from the information processing center, sending instructions to the staff, and sending the goods to the customers by deliverymen. Each of the above steps is important in order to achieve the goal of rapid delivery. In this paper, we formulate the above order processing as a joint scheduling problem of online order picking and distribution.

Order picking is a process of picking the required products from the shelves accurately and quickly, according to the information of customers' orders. It is one of the most important operations in the distribution center. An appropriate order picking system helps improve working efficiency and customer satisfaction [1]. Many literatures have studied such problems, mainly involving design of goods location [2], picking routes [3], order batching [4, 5], and partition picking [6]. In the "online shopping" mode, logistics distribution business poses a lot of new challenges. Researchers have studied the decision-making and route optimization of e-commerce logistics distribution, such as [7, 8]. In order to deliver goods to customers quickly, it is necessary to manage order picking and distribution as a whole. This is because the starting time of vehicle distribution is directly related to the completion time of order picking. Wang et al. [9] constructed a joint scheduling model of picking and distribution to minimize orders' fulfillment time. Peng [10] studied a joint scheduling problem of order batching and distribution routing and took the shortest total fulfillment time from order picking to distribution completion as the goal of B2C e-commerce distribution center. Zhang et al. [11] studied the online integrated order picking and delivery problem with the multizone routing method to minimize the sum of the maximum delivery completion time and the total delivery cost.

The order picking in this paper is different from the traditional warehouse order picking. A HEMA offline store is also a warehouse, which increases the difficulty of order fulfillment due to the variety of commodity demand and small batch sizes. All orders submitted through HEMA app are picked by the staff in the sales area. The placement of goods in the sales area is different from that in traditional warehouses. There is no unified standard shelf. The display and the placement of goods varies from shop to shop according to the preferences of consumers in the area where the store is located. Therefore, we study the picking and distribution problem of orders generated in HEMA APP from a different

view. It is assumed that, in a certain period of time, there are some orders to be picked out and distributed to customers. These orders are regarded as jobs, the staff and conveyor belts in HEMA stores are regarded as machines, and the deliverymen are regarded as vehicles. We analyze the combined problem of order picking and distribution and establish the model with the goal of minimizing the sum of orders' total completion time and total delivery cost.

From the view of scheduling research, we study a machine scheduling problem with transportation of finished jobs. Hall and Potts [12] were the first to propose a joint machine scheduling problem with job transportation. Chen [13] gave a detailed review on the joint problems of machine scheduling and product transportation published before 2008. Zhong et al. [14] and Fu et al. [15] studied scheduling problems of outsourcing the transportation of products to third-party logistics companies. Ullrich [16] considered machine scheduling and job transportation problems with path planning. In this paper, we consider orders' picking and distribution in HEMA as a joint scheduling problem with distribution coordination. Orders are first processed in dedicated flow shops followed by assembly lines and then are distributed to the customers by deliverymen. To the best of our knowledge, this machine scheduling problem has not been covered in any previous literature.

The remaining sections of this paper are organized as follows: In Section 2, we give the problem description; in Section 3, we construct a MP model for the case with one machine at each stage and one deliveryman; in Section 4, we devise algorithms to solve the problem; in Section 5, numerical experiments are carried out to verify the rationality of the model and the effectiveness of the algorithms; in Section 6, we conclude this paper.

2. Problem Descriptions

Suppose that, during a certain period of time, there are several orders waiting to be processed in the online order system of HEMA. Each order is composed of several types of suborders and processed in four stages in sequence. The first and second stages are order processing within the shop: a subjob is first processed on a stage 1 machine (the staff picks out the products and put them into a bag) and then processed on a stage 2 machine (the conveyor belt transfers the bag to packing area); then, in the packing area, suborders belonging to the same order are assembled together for distribution; this is stage 3. In stage 4, completed orders are distributed to their customers in batches by deliverymen with electric power carts. The full process is shown in Figure 1.

We regard each order as a job, and a suborder as a subjob (order and job, and suborder and subjob will be used interchangeably in the following). The staff and conveyor belts are regarded as machines, and the deliverymen are regarded as vehicles.

Next, we describe the above process of orders' picking and distribution as a joint machine scheduling and transportation problem. At the beginning of the planning horizon, n orders $\{O_1, O_2, \dots, O_n\}$ are placed by n customers. Job O_i may contain one or more subjobs O_i^l (item type- l),

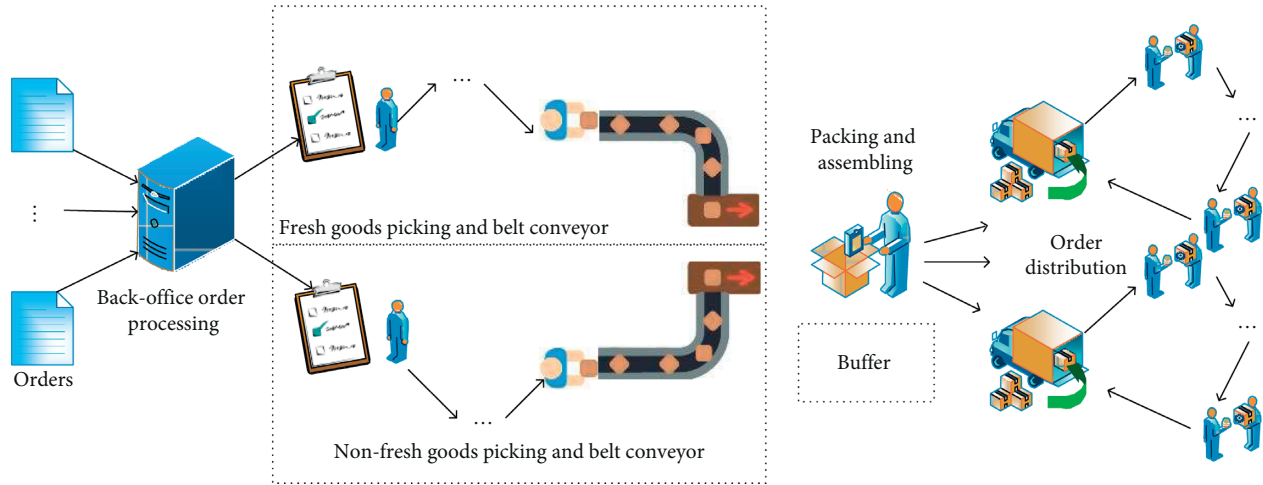


FIGURE 1: Order processing and distribution.

$i = 1, 2, \dots, n, l = 1, 2, \dots, k$, where each type of suborders is processed by a dedicated two-stage flow-shop machine $F_l(1, 1)$. For $l = 1, 2, \dots, k$, each subjob O_i^l is first processed by a stage 1 machine M_1^l with processing time p_{i1}^l (time needed for staff's picking) and then processed by a stage 2 machine M_2^l with processing time p_{i2}^l (time needed for belt conveying). When subjobs $O_i^l, i = 1, 2, \dots, n, l = 1, 2, \dots, k$ are all finished, they need to be assembled together for distribution. The processing time for job O_i 's assembling operation is p_i^3 . Each vehicle contains a batch of jobs when it departs from the store. It delivers the jobs to the customers one by one and returns to the store after it delivers the last job. Then, it is ready for the next batch's delivery. It can carry no more than Q jobs in one batch. In this paper, a complete undirected graph $G = (V, E)$ with $n + 1$ vertices is used to represent the geographical locations and distances between the HEMA store and n customers, $V = \{V_0, V_1, V_2, \dots, V_n\}$, $E = \{e_{ij}, i \neq j, i, j = 1, 2, \dots, n\}$, where V_0 is the HEMA store, and V_1, V_2, \dots, V_n represent n customers. It takes time of t_{ij} ($i, j = 1, 2, \dots, n$) for the vehicle travelling between V_i and V_j . The vehicle starts from V_0 , distributes the orders in the current batch to their customers one by one, and returns to V_0 . We assume that the triangle inequality is satisfied between any three points in the distribution network. Since a batch may contain a variety of orders (e.g., requiring refrigeration or additional packaging), each order occurs a distribution cost d_i . If several orders are contained in one batch, the distribution cost of this batch equals the arithmetic mean of the cost of all orders in the batch.

Our goal in this paper is to design a solution for the above joint machine scheduling and distribution problem, such that the total arrival time (sum of the times when the customers receive their orders) and the total distribution cost are minimized.

We need to make the following decisions: (1) subjobs' arrangement on the dedicated flow-shop machines and the assembly machines; (2) batching decision before distribution, including the number of batches and jobs contained in each batch; (3) the time when each batch leaves the machine and the routes of distributing orders in each batch.

3. MP Model for a Special Case

In this section, we study the case when each job contains two subjobs (subjob A and subjob B denote fresh goods and nonfresh goods, respectively), and there are one assembly machine and one vehicle. We construct a mathematical programming model for this problem. It seems that it is a bit special; however, we can see that it reveals the meaning for the coordinated consideration of order picking and distribution. We first list parameters and variables for this model.

Parameters

N : the set of jobs, $N = \{O_1, O_2, \dots, O_n\}$

O_i : the i th job

O_i^A : subjob A of job O_i

O_i^B : subjob B of job O_i

p_{i1}^A : processing time of subjob A of job O_i at the first stage

p_{i1}^B : processing time of subjob B of job O_i at the first stage

p_{i2}^A : processing time of subjob A of job O_i at the second stage

p_{i2}^B : processing time of subjob B of job O_i at the second stage

p_{i3} : processing time of job O_i on the assembling machine

V : the set of vertices in the distribution network

t_{ij} : transportation time for a vehicle travelling between V_i and V_j

Q : capacity constraint of a distribution batch

d_i : distribution cost for each job

$y_i, y_i = 1$ if job O_i contains subjob A; otherwise, $y_i = 0$

$z_i, z_i = 1$ if job O_i contains subjob B; otherwise, $z_i = 0$

Variables

s_{i1}^A : starting time of subjob A of job O_i at the first stage

c_{i1}^A : finishing time of subjob A of job O_i at the first stage

s_{i1}^B : starting time of subjob B of job O_i at the first stage

c_{i1}^B : finishing time of subjob B of job O_i at the first stage

s_{i2}^A : starting time of subjob A of job O_i at the second stage

c_{i2}^A : finishing time of subjob A of job O_i at the second stage

s_{i2}^B : starting time of subjob B of job O_i at the second stage

c_{i2}^B : finishing time of subjob B of job O_i at the second stage

s_{i3} : starting time of job O_i on the assembling machine

c_{i3} : finishing time of job O_i on the assembling machine

s_{i4} : departure time of job O_i (departure time of the batch that job O_i belongs to)

c_{i4} : completion time of job O_i (when it is delivered to its customer)

s_f : starting time of the distribution of batch B_f

c_f : finishing time of the distribution of batch B_f (when the last job is delivered, and the vehicle returns to the store)

F : the number of batches, $\lceil n/Q \rceil \leq F \leq n$

B : the set of distribution batches, $B = \{B_1, B_2, \dots, B_F\}$

b_f : number of jobs in batch B_f , $f = 1, 2, \dots, F$

$e_{ij}^{A1} = 1$ if O_i^A precedes O_j^A at the first stage, else = 0

$e_{ij}^{A2} = 1$ if O_i^A precedes O_j^A at the second stage, else = 0

$e_{ij}^{B1} = 1$ if O_i^B precedes O_j^B at the first stage, else = 0

$e_{ij}^{B2} = 1$ if O_i^B precedes O_j^B at the second stage, else = 0

$e_{ij}^3 = 1$ if O_i precedes O_j at the third stage, else = 0

$v_{ij}^f = 1$ if vehicle travels from V_i to V_j in batch B_f , else = 0

$x_{if} = 1$ if job O_i is allocated to batch B_f , else = 0

$v_{ij} = 1$ if vehicle travels from V_i to V_j in some batch, else = 0 (For ease of presentation, let $v_{jj} = 1$)

D_f : distribution cost for batch B_f

Mathematical programming model is as follows:

$$\min \sum_{i=1}^n c_{i4} + \sum_{f=1}^F D_f, \quad (1)$$

$$\text{s.t. } s_{i1}^A \geq 0, \quad i = 1, 2, \dots, n, \quad (2)$$

$$s_{i1}^B \geq 0, \quad i = 1, 2, \dots, n, \quad (3)$$

$$c_{i1}^A = (s_{i1}^A + p_{i1}^A) \times y_i, \quad i = 1, 2, \dots, n, \quad (4)$$

$$c_{i2}^A = (s_{i2}^A + p_{i2}^A) \times y_i, \quad i = 1, 2, \dots, n, \quad (5)$$

$$c_{i1}^A - s_{i2}^A \leq 0, \quad i = 1, 2, \dots, n, \quad (6)$$

$$c_{i1}^B - s_{i2}^B \leq 0, \quad i = 1, 2, \dots, n, \quad (7)$$

$$c_{i1}^B = (s_{i1}^B + p_{i1}^B) \times z_i, \quad i = 1, 2, \dots, n, \quad (8)$$

$$c_{i2}^B = (s_{i2}^B + p_{i2}^B) \times z_i, \quad i = 1, 2, \dots, n, \quad (9)$$

$$c_{i3} = s_{i3} + p_{i3}, \quad i = 1, 2, \dots, n, \quad (10)$$

$$\max\{c_{i2}^A, c_{i2}^B\} - s_{i3} \leq 0, \quad i = 1, 2, \dots, n, \quad (11)$$

$$s_{j1}^A \geq c_{i1}^A - (1 - e_{ij}^{A1})M, \quad i, j = 1, 2, \dots, n, i \neq j, \quad (12)$$

$$s_{j1}^B \geq c_{i1}^B - (1 - e_{ij}^{B1})M, \quad i, j = 1, 2, \dots, n, i \neq j, \quad (13)$$

$$s_{j2}^A \geq c_{i2}^A - (1 - e_{ij}^{A2})M, \quad i, j = 1, 2, \dots, n, i \neq j, \quad (14)$$

$$s_{j2}^B \geq c_{i2}^B - (1 - e_{ij}^{B2})M, \quad i, j = 1, 2, \dots, n, i \neq j, \quad (15)$$

$$s_{j3} \geq c_{i3} - (1 - e_{ij}^3)M, \quad i, j = 1, 2, \dots, n, i \neq j, \quad (16)$$

$$\sum_{f=1}^F x_{if} = 1, \quad i = 1, 2, \dots, n, \quad (17)$$

$$\sum_{O_i \in B_f} x_{if} \leq Q, \quad f = 1, 2, \dots, F, \quad (18)$$

$$\lceil \frac{n}{Q} \rceil \leq F \leq n, \quad (19)$$

$$s_f = \max_{i \in \{1, 2, \dots, n\}} \{c_{i3} - (1 - x_{if})M\}, \quad f = 1, 2, \dots, F, \quad (20)$$

$$s_{i4} = \max_{i \in \{1, 2, \dots, n\}} \{s_f - (1 - x_{if})M\}, \quad f = 1, 2, \dots, F, \quad (21)$$

$$D_f = \frac{\sum_{i=1}^n x_{if} d_i}{b_f}, \quad f = 1, 2, \dots, F, \quad (22)$$

$$s_f - c_{f-1} \geq 0, \quad f = 2, 3, \dots, F, \quad (23)$$

$$x_{if} = \sum_{j \in N \cup \{0\} \setminus \{i\}} v_{ij}^f, \quad i = 1, 2, \dots, n, f = 1, 2, \dots, F, \quad (24)$$

$$\sum_{f=1}^F \sum_{i=0}^n v_{ij}^f = 1, \quad j = 1, 2, \dots, n, \quad (25)$$

$$\sum_{f=1}^F \sum_{j=0}^n v_{ij}^f = 1, \quad i = 1, 2, \dots, n, \quad (26)$$

$$\sum_{j=1}^n v_{0j}^f = 1, \quad f = 1, 2, \dots, F, \quad (27)$$

$$\sum_{j=1}^n v_{j0}^f = 1, \quad f = 1, 2, \dots, F, \quad (28)$$

$$c_f = s_f + \sum_{i=0}^n \sum_{j=1}^{n+1} t_{ij} v_{ij}^f, \quad f = 1, 2, \dots, F, \quad (29)$$

$$c_{i4} + t_{ij} v_{ij}^f - M(1 - v_{ij}^f) \leq c_{j4}, \quad i = 1, \dots, n, j = 1, \dots, n, \\ f = 1, 2, \dots, F, \quad (30)$$

$$s_f + t_{0j} v_{0j}^f - M(1 - v_{0j}^f) \leq c_{j4}, \quad j = 1, \dots, n, f = 1, 2, \dots, F. \quad (31)$$

Objective function (1) is to minimize the sum of the total completion time and the total distribution cost of jobs. Constraints (2) and (3) ensure that each subjob starts processing after time zero. Constraints (4) and (5) denote the finishing time of type A subjobs at the first stage and second stage. Constraints (6) and (7) indicate that a subjob can be started on stage 2 machine only after it has been finished on the corresponding stage 1 machine. Constraints (8) and (9) denote the finishing time of type B sub jobs at the first stage and second stage. Constraint (10) denotes the finishing time of the jobs at the third stage. Constraint (11) ensures that a job can be assembled only after its two subjobs have been completed. Constraints (12) to (16) ensure that, at any time, only one job can be processed on a machine. Constraint (17) ensures that a job is contained in exactly one batch. Constraint (18) ensures that the number of jobs in one batch is no larger than the capacity constraint. Constraint (19) denotes the bound for the number of distribution batches. Constraint (20) ensures that the vehicle can start travelling only after all jobs in one batch are completed. Constraint (21) indicates that, for each job, the departure time is the starting time of corresponding batch distribution. Constraint (22) denotes the cost of distribution for each batch. Constraint (23) ensures that the vehicle starts a new distribution batch only if he has distributed all the jobs in the last batch and returns to the store. Constraints (24) to (26) indicate that each customer is visited exactly once. Constraints (27) and (28) indicate that the vehicle travels from/to the store exactly F times. Constraint (29) denotes the time when the vehicle returns to the store equal to its starting time of the current batch plus the total travel time in the routing. Constraints (30) to (31) denote the delivery time of job O_i in batch B_f .

4. A Genetic Algorithm

Note that the two-stage assembly scheduling problem with two parallel machines in the first stage and no delivery is strongly NP-hard [17]. Therefore, the problem in this paper with two dedicated flow shops in the first stage and delivery coordination is obviously strongly NP-hard. Only relatively small instances can be optimally solved in reasonable computational time. Consequently, in this section, we develop a genetic algorithm to find good solutions [18].

For a given sequence of $\{O_1, O_2, \dots, O_n\}$, we first devise algorithm CA-D (Algorithm 1) to arrange the jobs' processing and distribution.

Note that algorithm CA-D generates a feasible schedule for our problem; however, this schedule is not that good. This is because the processing sequence and delivery sequence are the same in one batch. Next, we propose algorithm LGP (Algorithm 2) to improve this problem. LGP does not change the batching decisions but tries to reduce the completion time on the machines as well as the total transportation time of each batch.

Next, we list the details of our genetic algorithm:

(1) Coding and encoding

A chromosome represents a sequence of orders' picking and distribution based on the above algorithm CA-D. Figure 2 shows an example of a chromosome with ten genes indicating that there are ten jobs to be processed. $\{1, 5, 4, 2, 7, 10, 9, 8, 3, 6\}$ is the input of algorithm CA-D in order to determine the processing sequence on each machine and the distribution batches with routing consideration.

(2) Fitness function

We define the fitness function of a chromosome as the inverse of the objective function: $1/(\sum_{i=1}^n c_{i4} + \sum_{f=1}^F D_f)$.

(3) Genetic operators

Genetic operators used in this paper include selection, crossover, mutation, reversal, and reinsertion. First, parent chromosomes are selected from the initial population, then the offspring chromosomes are generated by crossover operator, and then mutation operation is performed. In order to improve the local search ability, reverse operation is carried out. Finally, in order to handle the reduction of the number of chromosomes caused by the selection operator, reinsertion operation is carried out:

(3.1) Selection

In this paper, Roulette wheel selection is used to select offspring from the parent population. Roulette wheel selection is similar to the roulette wheel used in gambling. Table 1 shows the probability of selection of an individual based on the fitness functions for a population of ten individuals, where the probability of selection of an individual equals (the fitness of the individual ÷ the sum of the fitness of all individuals in the population of the individual) × 100%.

Each time, we randomly generate two numbers between $[0, 1]$, which is equivalent to turning the wheel twice to get the position of the pointer (the sector to which the pointer stops is the selected individual), and the two selected individuals are the parents. As shown in Figure 3, we plot the probability of each individual being selected on a roulette wheel for the

Step 1: process type A subjobs on each stage of the dedicated flow-shop in the sequence of $\{O_1^A, O_2^A, \dots, O_n^A\}$ and process type B subjobs on each stage of the dedicated flow-shop in the sequence of $\{O_1^B, O_2^B, \dots, O_n^B\}$. Assemble the jobs in the sequence of $\{O_1, O_2, \dots, O_n\}$.

Step 2: jobs are delivered in $F = \lceil n/Q \rceil$ batches. For $f = 2, \dots, F - 1$, batch B_f contains $\{O_{(f-1) \times Q + 1}, O_{(f-1) \times Q + 2}, \dots, O_{f \times Q}\}$, and the last batch B_F contains $\{O_{(F-1) \times Q + 1}, O_{(F-1) \times Q + 2}, \dots, O_n\}$.

Step 3: the vehicle distributes the batches in the sequence of $\{B_1, B_2, \dots, B_F\}$, and the routing inside each batch is in the same sequence as that on the assembly machine. (The jobs are delivered in the sequence of $\{O_1, O_2, \dots, O_n\}$.)

ALGORITHM 1: CA-D.

Input: a feasible solution of generated from CA-D with delivery batches of $\{B_1, B_2, \dots, B_F\}$.

Output: an improved feasible solution.

Initialization: $f = 1$.

Step 1: for batch B_f , select two jobs randomly from this batch. Swap the two jobs in the processing sequence, i.e., on each machine, corresponding subjobs are swapped. Compute the completion time of the last job in B_f on the assembly machine. Repeat such swapping operation 50 times and select the processing sequence with the minimum completion time of the last job in B_f .

Step 2: for batch B_f , select two jobs randomly from this batch. Swap the two jobs in the delivery sequence and compute the total delivery time of this new sequence. Repeat such swapping operation 50 times and select the delivery sequence with the minimum total delivery time.

Step 3: if $f < F$, $f = f + 1$, go to Step 1; otherwise, stop.

ALGORITHM 2: LGP.

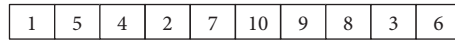


FIGURE 2: An example of a chromosome with ten genes.

TABLE 1: Calculation of selection probabilities based on individual fitness.

Individual	Chromosome	Fitness	Probability of selection	Cumulative probability
1	10, 8, 9, 5, 6, 3, 2, 1, 4, 7	5	0.060976	0.060976
2	10, 8, 9, 5, 4, 7, 6, 3, 2, 1	4	0.048780	0.109756
3	9, 5, 6, 3, 10, 8, 2, 1, 4, 7	9	0.109756	0.219512
4	4, 8, 9, 5, 6, 3, 2, 1, 10, 7	7	0.085366	0.304878
5	9, 5, 6, 10, 8, 3, 2, 1, 4, 7	8	0.097561	0.402439
6	9, 5, 6, 2, 1, 4, 7, 10, 8, 3	8	0.097561	0.500000
7	8, 3, 2, 1, 4, 7, 9, 5, 6, 10	5	0.060976	0.560976
8	3, 2, 1, 10, 8, 4, 7, 9, 5, 6	10	0.121951	0.682927
9	1, 5, 4, 2, 7, 10, 9, 8, 3, 6	15	0.182927	0.865854
10	2, 9, 7, 5, 6, 4, 3, 8, 10, 1	11	0.134146	1.000000

population in Table 1 and then select the parent individuals in turn by roulette selection.

(3.2) Crossover

In this paper, Partially Matched Crossover (PMX) method is used in the crossover. In PMX, we first select two crossover points randomly, exchange the genes between the two crossover points, and then replace the repeated genes through allele mapping. In Figure 4, we show the process of creating two offspring from a pair of parents by PMX crossover and

mutation, using individuals 9 and 10 from Table 2: first, the crossover region of the parent individual is selected, and the genes of the two chromosomes within the crossover region are crossed over. As the new chromosome contains duplicated genes, we replace the duplicated genes by mapping the alleles of the parent individuals (4 – 7 – 10, 2 – 5, 7 – 6, 9 – 3). A random number in $[0, 1]$ is generated, and if it is smaller than the crossover probability c , crossover is run.

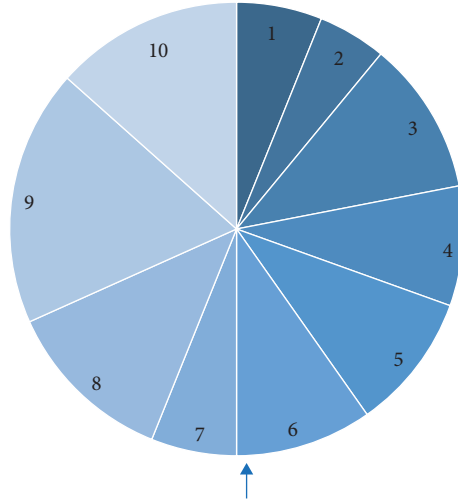


FIGURE 3: Roulette wheel selection.

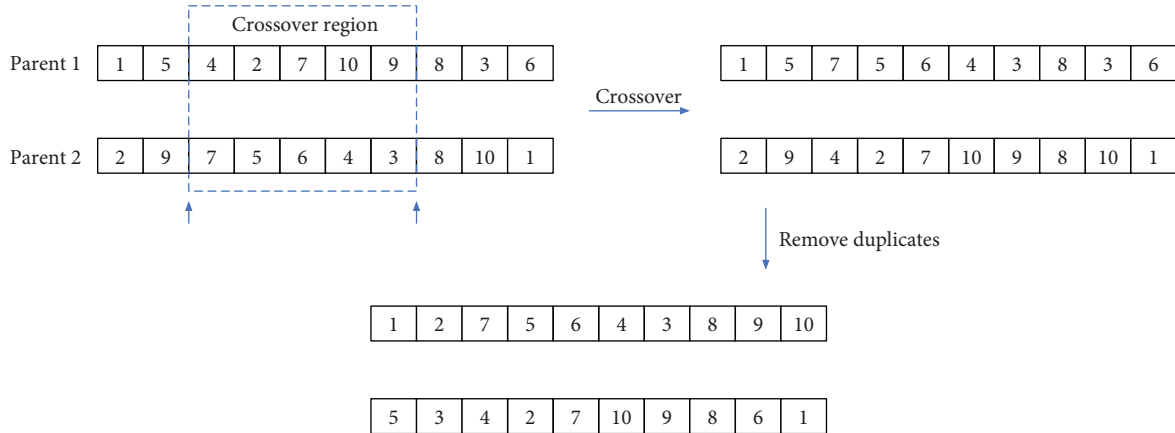


FIGURE 4: Partially matched crossover operation.

TABLE 2: Parameters for 60 instances.

Group	Instance	p_{i1}^A	p_{i2}^A	p_{i1}^B	p_{i2}^B	p_{i3}	d_i	t_{ij}
X	1–20	[5, 10]	[2, 3]	[3, 9]	[1, 2]	[2, 8]	[7, 13]	[4, 7]
Y	21–40	[0, 10]	[0, 3]	[0, 9]	[0, 2]	[2, 8]	[7, 13]	[4, 7]
Z	41–60	[0, 10]	[0, 3]	[0, 9]	[0, 2]	[2, 8]	[7, 13]	[40, 70]

(3.3) Mutation

In the mutation operation, for the two individuals after the crossover operation, we randomly select two positions in an individual to exchange genes at these two positions, as shown in Figure 5. A random number in [0, 1] is generated, and if it is smaller than the mutation probability m , mutation is run.

value of the current population is denoted as CR. If $(|LB - CR|/LB) \leq \epsilon$ (ϵ is a given positive number) or the number of iterations reaches 500, stop.

Now we are ready to devise a genetic algorithm DU-GA (Algorithm 3) for the integrated order picking and distribution problem.

(4) Iteration termination conditions

The minimal fitness function value of the last generated population is denoted as LB, where the initial value of LB is the minimum fitness function value of the initial population. The minimal fitness function

5. Numerical Experiments and Analysis

In order to verify the rationality of the above model and test the performance of algorithm DU-GA, we conduct numerical experiments and analyze the results based on Python 3.8. All numerical experiments are performed on a personal

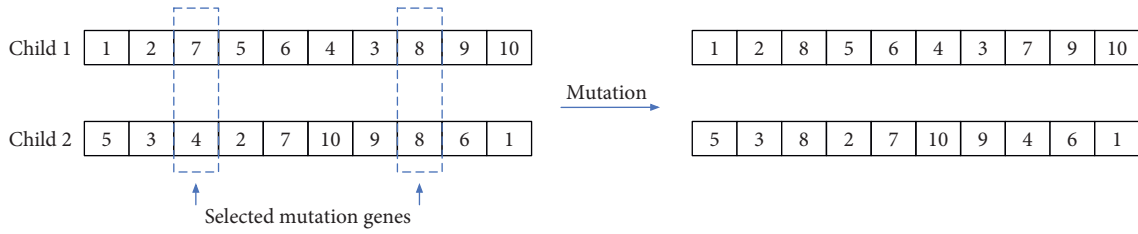


FIGURE 5: Mutation operation.

TABLE 3: Parameter settings.

Group	(50, 0.7, 0.1)	(100, 0.7, 0.1)	(150, 0.7, 0.1)	(150, 0.8, 0.1)	(150, 0.9, 0.1)	(150, 0.8, 0.2)	(150, 0.8, 0.3)
X	13455.71	12737.29	12725.33	12410.71	12411.79	12225.33	13156.88
Y	9991.83	9948.67	10067.88	10049.04	10202.08	9691.38	9739.04
Z	73821.38	73692.67	73705.42	73221.75	73667.25	72058.00	73418.88

computer with 8 GB memory and 2.70-GHz Intel Core i5 CPU.

5.1. Parameter Setting

5.1.1. Parameters for Instances. According to different situations of these parameters, we carry out three groups of numeral experiments, where each group randomly generates 20 instances. Each instance contains 50 jobs, and the vehicle capacity is $Q = 8$.

Table 2 shows the parameter intervals for 20 instances of each Group. For example, parameters for Group-X are set as $p_{i1}^A \in [5, 10]$, $p_{i2}^A \in [2, 3]$, $p_{i1}^B \in [3, 9]$, $p_{i2}^B \in [1, 2]$, $p_{i3} \in [2, 8]$ and $d_i \in [7, 13]$ ($p_{i1}^A \in [5, 10]$ means that p_{i1}^A is a random integer uniformly generated in interval of $[5, 10]$ and the other parameter settings are similar to this).

Group-X (Instances 1–20) considers the case when orders contain both fresh and nonfresh products, where the processing time is always larger than zero. The picking time for fresh products is generally longer than that for nonfresh products due to the additional steps (i.e., dewatering and weighing) during the picking process. Group-Y (Instances 21–40) and Group-Z (Instances 41–60) allow orders containing only fresh products or only nonfresh products, in recognition of the fact that customers may need to buy only fresh or nonfresh products, where the processing time may be zero, which is more realistic. Group-Z (Instances 41–60) examines cases where the delivery time of orders is significantly longer than processing time.

5.1.2. Parameters for Genetic Operators. Before the genetic algorithm is applied, the following parameters need to be determined: population size P , crossover probability c , and mutation probability m . P is set to be one of $\{50, 100, 150\}$ and maintains the same size for each iteration. The crossover probability c is set to be one of $\{0.7, 0.8, 0.9\}$. The mutation probability m is generally set to be in $[0.05, 0.1]$; however, considering the strong NP-hardness of the problem, m is set to be one of $\{0.1, 0.2, 0.3\}$ to expand the search space. Since the actual performance of genetic algorithm is closely related

to these parameters, we first test different combinations of (P, c, m) . For each group, an instance is generated. For different parameters of GA, the algorithm is run five times for each instance. Table 3 lists the average objective value for the instances (i.e., 13477.71 online 2 column 2 denotes the average objective value of the instance of Group-X with parameter $(50, 0.7, 0.1)$, and the others are similar to this).

We observe the following:

- (1) If $(c, m) \in (0.7, 0.1)$, by comparing $(P, c, m) \in \{(50, 0.7, 0.1), (100, 0.7, 0.1), (150, 0.7, 0.1)\}$, when $P = 150$, DU-GA terminates with the best objective value
- (2) If $P = 150$, by comparing $(P, c, m) \in \{(150, 0.7, 0.1), (150, 0.8, 0.1), (150, 0.9, 0.1)\}$, when $c = 0.8$, DU-GA terminates with the best objective value
- (3) Based on (1) and (2), we let $P = 150$ and $c = 0.8$. If $(P, c, m) \in \{(150, 0.8, 0.1), (150, 0.8, 0.2), (150, 0.8, 0.3)\}$, when $m = 0.8$, the objective value is best

Therefore, we set the parameters (P, c, m) for DU-GA to be $(150, 0.8, 0.1)$.

5.2. Analysis of Experimental Results

5.2.1. Stability Test. To evaluate the stability of algorithm DU-GA, for Groups X, Y and Z, we run each instance five times to assess the span between the minimal solution (denoted as S_B) and the average value (denoted as S_A), which is referred to as Relative Deviation (denoted as RD), calculating by $RD = ((S_A - S_B)/S_A) \times 100\%$.

Table 4 shows that the span between S_A and S_B is small, with an average RD of 1.46% for Group-X, 1.97% for Group-Z, and 2.05% for Group-Y. Therefore, DU-GA is stable in all three groups of instances.

5.2.2. Comparison Algorithm. To evaluate the effectiveness of the algorithm DU-GA, we design comparison algorithms DU-1 (Algorithm 4) and DU-2 (Algorithm 5). Algorithm DU-1 is a genetic algorithm that is different from DU-GA in that it calculates the fitness function based on the departure

TABLE 4: Stability test of DU-GA.

Group	Min RD (%)	Max RD (%)	Average RD (%)
X	0.35	2.48	1.46
Y	1.41	3.71	1.97
Z	0.90	4.94	2.05

Step 1: determine the population size P , the crossover probability c , the mutation probability m , and the termination condition of the algorithm, followed by initializing the population (i.e., randomly generating P individuals).
Step 2: for each individual, apply algorithm CA-D. Evaluate fitness values of the chromosome and select two individuals as parents.
Step 3: perform crossover and mutation operators on the selected parents to obtain offspring individuals.
Step 4: update the population by adding the newly generated individuals to the population. Step 2 and Step 3 are repeated until the termination condition is met, and the iteration is stopped.
Step 5: select the best solution from the current population, and let it be the input of algorithm LGP. Run algorithm LGP, and output the resulted solution.

ALGORITHM 3: DU-GA.

time of each job s_{i4} . Algorithm DU-2 processes and distributes jobs in a random order.

5.2.3. Algorithm Effectiveness Analysis. In Tables 5–7, the results for instances of Groups X, Y and Z by applying DU-GA, DU-1, and DU-2 are listed. For ease of presentation, we denote the objective value of the solution obtained from DU-GA, DU-1, and DU-2 as C_{DU-GA} , C_{DU-1} and C_{DU-2} , respectively. We run each instance five times and record the mean of the objective values. The first column shows the index of the 20 instances; the second and third columns show the results of DU-2 and DU-1; the fourth column shows the objective values of the solutions obtained from DU-GA. We record the reduction of C_{DU-GA} over C_{DU-1} and C_{DU-2} in the fifth and sixth columns and show the percentage improvement of C_{DU-GA} over C_{DU-1} and C_{DU-2} in the seventh and eighth columns.

We also observe that DU-GA takes the longest time in most instances, but the maximal does not exceed 11.83 seconds; the running time of DU-1 is shorter than that of DU-GA, while the running time of DU-2 is the shortest. Compared with DU-1, DU-GA not only considers the delivery times in fitness function but also adds further optimization of the production and distribution stages based on the basic GA algorithm:

(1) Compare DU-GA with DU-1

According to Tables 5–7, the average improvement of C_{DU-GA} over C_{DU-1} for Groups X, Y and Z instances is 1.94%, 4.00%, and 5.12%.

- (1.1) It is shown that the improvement of C_{DU-GA} over C_{DU-1} of Group-Y is better than that of Group-X. Note that the processing times of jobs in Group-Y are smaller than those of Group-X, and thus the distribution time is relatively larger in Group-Y. DU-GA differs from DU-1 in that it considers the delivery times as well as processing times in the fitness function;

therefore, the larger the delivery times, the more the improvement of C_{DU-GA} over C_{DU-1} .

- (1.2) The improvement of C_{DU-GA} over C_{DU-1} of Group-Z is better than that of Group-Y. Note that the transportation times of instances in Group-Z are obviously higher. The same reason as in (1.1) can explain this observation.

- (1.3) The results of Group-Z are significantly better than those of the other two groups, meaning that, compared to algorithm DU-1, algorithm DU-GA is highly effective for the practical case when orders may only contain fresh items or nonfresh items and take longer for distribution than processing.

(2) Compare DU-GA with DU-2

According to Tables 5–7, the average improvement of DU-GA over DU-2 for Groups X, Y and Z instances is 5.35%, 9.70%, and 10.38%.

- (2.1) It is shown that the average improvement of C_{DU-GA} over C_{DU-2} of Group-Y is better than that of Group-X. DU-2 is a FIFO strategy, and no optimization skill is applied. DU-GA is a genetic algorithm, and it considers the processing times and the delivery times in the fitness function. Note that there is only one vehicle for distribution, and multiple orders are contained in one batch; thus, the optimization of the distribution stage plays a more important role than that of the production stages. Recall that the processing times of jobs in Group-Y are smaller than those of Group-X, and thus the distribution time is relatively larger in Group-Y. Therefore, the larger the delivery times, the more the improvement of C_{DU-GA} over C_{DU-2} .
- (2.2) The improvement of C_{DU-GA} over C_{DU-2} of Group-Z is better than that of Group-Y. Note that the transportation times of instances in

Step 1: same as Step 1 of DU-GA;
 Step 2: for each individual, apply algorithm CA-D. Calculate the fitness value for individuals in the population by taking $1/\sum_{i=1}^n s_{i4} \sum_{f=1}^F D_f$ as the fitness function.
 Step 3-Step 4: same as Step 3 - Step 4 of DU-GA;
 Step 5: output the processing sequence, batching decision, and delivery routings of the best solution based on fitness function $1/\sum_{i=1}^n s_{i4} \sum_{f=1}^F D_f$. Calculate the corresponding objective function $\sum_{i=1}^n c_{i4} + \sum_{f=1}^F D_f$.

ALGORITHM 4: DU-1.

Step 1: run algorithm CA-D in a random order;
 Step 2: output the processing sequence, batching decision and delivery routings. Calculate the corresponding objective $\sum_{i=1}^n c_{i4} + \sum_{f=1}^F D_f$.

ALGORITHM 5: DU-2.

TABLE 5: Experimental results of 20 instances in Group-X.

Instance	C_{DU-2}	C_{DU-1}	C_{DU-GA}	$C_{DU-GA} - C_{DU-1}$	$C_{DU-GA} - C_{DU-2}$	$(C_{DU-GA} - C_{DU-1})/C_{DU-1}$ (%)	$(C_{DU-GA} - C_{DU-2})/C_{DU-2}$ (%)
ins1	13189.63	13025.50	12841.38	184.13	348.25	1.41	2.64
ins2	13173.00	12871.25	12680.50	190.75	492.50	1.48	3.74
ins3	13038.75	12852.13	12748.13	104.00	290.63	0.81	2.23
ins4	13380.00	12702.63	12537.75	164.88	842.25	1.30	6.29
ins5	13078.50	12928.00	12580.50	347.50	498.00	2.69	3.81
ins6	13361.75	12919.38	12837.88	81.50	523.88	0.63	3.92
ins7	13320.00	12870.25	12645.13	225.13	674.88	1.75	5.07
ins8	13145.75	12945.25	12643.50	301.75	502.25	2.33	3.82
ins9	13145.75	12945.25	12643.50	301.75	502.25	2.33	3.82
ins10	12851.88	12311.13	12168.38	142.75	683.50	1.16	5.32
ins11	12892.88	12153.13	11974.25	178.88	918.63	1.47	7.13
ins12	12898.50	12189.50	12022.50	167.00	876.00	1.37	6.79
ins13	13040.50	12281.88	11918.13	363.75	1122.38	2.96	8.61
ins14	13160.38	12254.00	12063.38	190.63	1097.00	1.56	8.34
ins15	13223.25	12294.25	11984.00	310.25	1239.25	2.52	9.37
ins16	12559.75	12365.63	12011.75	353.88	548.00	2.86	4.36
ins17	12969.50	12118.25	11956.50	161.75	1013.00	1.33	7.81
ins18	12733.00	12410.63	11903.25	507.38	829.75	4.09	6.52
ins19	13716.88	13627.00	13225.13	401.88	491.75	2.95	3.59
ins20	13561.63	13295.00	13046.88	248.13	514.75	1.87	3.80
Average	13122.06	12668.00	12421.62	246.38	700.44	1.94	5.35

Group-Z are obviously higher. The same reason as in (2.1) can explain this observation.

(2.3) The results of Group-Z are significantly better than those of the other two groups, meaning that, compared to algorithm DU-2, algorithm DU-GA is highly effective for the practical case when orders may only contain fresh items or nonfresh items, and the time for distribution is longer than processing.

(3) For each instance, $C_{DU-GA} - C_{DU-2}$ is always larger than $C_{DU-GA} - C_{DU-1}$. DU-2 is a FIFO strategy, and no optimization skill is applied, while DU-1 is a genetic algorithm, which considers jobs' processing

times in the fitness function. DU-1 is also an acceptable algorithm, but not as good as DU-GA.

5.2.4. The Enlightenment for the Management of New Retail Enterprises. The orders' processing sequence in each stage directly affects the order delivery time, which in turn has an impact on customer satisfaction. In order to improve customer satisfaction, we must consider different practical scenarios. The results of numerical experiments show that the genetic algorithm DU-GA is relatively stable for all instances. We also design comparison algorithms DU-1 and DU-2 to evaluate the effectiveness of DU-GA. The results show that algorithm DU-GA outperforms algorithm DU-1

TABLE 6: Experimental results of 20 instances in Group-Y.

Instance	C_{DU-2}	C_{DU-1}	C_{DU-GA}	$C_{DU-GA} - C_{DU-1}$	$C_{DU-GA} - C_{DU-2}$	$(C_{DU-GA} - C_{DU-1})/C_{DU-1}$ (%)	$(C_{DU-GA} - C_{DU-2})/C_{DU-2}$ (%)
ins21	11729.13	10633.63	10130.88	502.75	1598.25	4.73	13.63
ins22	10821.00	10077.50	9880.63	196.88	940.38	1.95	8.69
ins23	11544.13	10683.75	10340.50	343.25	1203.63	3.21	10.43
ins24	10883.38	10361.63	9816.38	545.25	1067.00	5.26	9.80
ins25	11455.38	11073.75	10707.88	365.88	747.50	3.30	6.53
ins26	11369.00	10325.00	9640.75	684.25	1728.25	6.63	14.45
ins27	10544.88	10399.63	9849.88	549.75	695.00	5.29	6.59
ins28	10915.63	10407.13	9877.88	529.25	1037.75	5.09	9.51
ins29	11418.13	10445.88	10103.13	342.75	1315.01	3.28	11.52
ins30	10843.63	10672.50	10224.00	448.50	619.63	4.20	5.71
ins31	10968.00	10574.00	9778.50	795.50	1189.50	6.58	9.93
ins32	11800.88	11006.00	10764.13	241.88	1036.75	2.20	8.79
ins33	11745.13	10813.63	10273.13	540.50	1472.00	5.00	12.53
ins34	11127.50	10227.13	9613.50	613.63	1514.00	6.00	13.61
ins35	11716.50	10931.00	10694.25	236.75	1022.25	2.17	8.72
ins36	11339.25	10513.88	10042.38	471.50	1296.88	4.48	11.44
ins37	11205.25	10787.50	10539.50	248.00	665.75	2.30	5.94
ins38	11774.25	10763.75	10635.75	128.00	1138.50	1.19	9.67
ins39	11000.25	10820.38	10659.00	161.38	341.25	1.49	3.10
ins40	11548.38	10616.75	10009.38	607.38	1539.00	5.72	13.33
Average	11287.48	10606.72	10179.07	427.65	1108.41	4.00	9.70

TABLE 7: Experimental results of 20 instances in Group-Z.

Instance	C_{DU-2}	C_{DU-1}	C_{DU-GA}	$C_{DU-GA} - C_{DU-1}$	$C_{DU-GA} - C_{DU-2}$	$C_{DU-GA} - C_{DU-1}/C_{DU-1}$ (%)	$C_{DU-GA} - C_{DU-2}/C_{DU-2}$ (%)
ins41	80295.00	74138.63	70463.13	3675.50	9831.88	4.96	12.24
ins42	80016.75	75866.75	71510.38	4356.38	8506.38	5.74	10.63
ins43	83567.00	78034.38	73665.00	4369.38	9902.00	5.60	11.85
ins44	80826.63	74994.88	70743.88	4251.00	10082.75	5.67	12.47
ins45	78806.63	74775.63	71372.88	3402.75	7433.75	4.55	9.43
ins46	83283.63	75297.75	71529.50	3768.25	11754.13	5.00	14.11
ins47	79538.88	76558.63	71696.88	4861.75	7842.00	6.35	9.86
ins48	81573.50	77321.38	73266.75	4054.63	8306.75	5.24	10.18
ins49	78296.25	74942.13	71648.13	3294.00	6648.13	4.40	8.49
ins50	81076.25	75760.25	72000.00	3760.25	9076.25	4.96	11.19
ins51	82371.63	78569.38	73763.38	4806.00	8608.25	6.12	10.45
ins52	79815.38	74644.25	71110.25	3534.00	8705.13	4.73	10.91
ins53	78848.38	75671.13	71391.25	4279.88	7457.13	5.66	9.46
ins54	80431.75	76728.88	73497.88	3231.00	6933.88	4.21	8.62
ins55	80261.38	75413.88	70473.25	4940.63	9788.13	6.55	12.20
ins56	79548.88	76709.63	73692.50	3017.13	5856.38	3.93	7.36
ins57	84032.75	78333.25	74515.25	3818.00	9517.50	4.87	11.33
ins58	77052.38	75897.00	71994.00	3903.00	5058.38	5.14	6.56
ins59	84053.75	76164.25	73829.63	2334.63	10224.13	3.07	12.16
ins60	78999.75	76957.00	72623.38	4333.63	6376.38	5.63	8.07
Average	80634.83	76138.95	72239.36	3899.59	8395.46	5.12	10.38

and algorithm DU-2 for all instances of three different groups.

From the above analysis, we can observe the following:

- (1) Algorithm DU-2, based on “first-arrive-first-process” method, is easy for operation. The manager of the new retail enterprise merely processes the orders according to the arrival sequence, and the distribution cost and delivery time can be relatively high.

- (2) Compared to DU-2, DU-1 takes the departure time of each job into consideration and results in a lower total order delivery time and delivery cost. This algorithm may be run in the development stage of optimizing the operation: the manager outsources the distribution to 3PL companies and focuses on the departure times and distribution costs of orders. However, when the customers receive their orders, it is out of control, and the manager has no idea of the customers’ satisfaction level.

- (3) Compared to DU-1, algorithm DU-GA takes both order delivery time and delivery cost into consideration. The manager determines orders' picking and delivery as a whole and can monitor the customers' satisfaction level and total delivery cost at the same time.

DU-2, DU-1, and DU-GA represent the degrees of attention to order picking and distribution at different stages:

- (1) DU-2 indicates the initial stage of enterprise establishment, and the variety of goods and the number of customers are relatively small. Therefore, the enterprise starts processing orders as soon as receiving them.
- (2) DU-1 represents the development stage of the enterprise. As the scale of business becomes larger, the number of orders increases. At this stage, the new retail enterprises may cooperate with 3PL companies; thus, the business managers only need to arrange the production to ensure orders' departure times.
- (3) DU-GA indicates a stable period after expansion. New retail enterprises start to focus on lean management of operation. They build their own logistics instead of resorting to 3PL companies. Hence, they need to jointly consider orders' picking and distribution.

6. Summery and Prospects

Based on the characteristics of orders' picking and delivery in new retail enterprises, this paper builds a joint scheduling and distribution model. An MP model is established for a fundamental case, and we devise a genetic algorithm as well as comparison algorithms. The results of computational experiments show the rationality of the model and the effectiveness of the algorithms. Furthermore, we simulate different practical situations to verify the applicability and performance of the algorithm. It is shown that our genetic algorithm is stable and outperforms comparison algorithms.

Our study provides management insights, showing that joint scheduling algorithms can significantly improve the efficiency of order picking and distribution in new retail enterprises. Reasonable coordination of order picking and distribution can effectively reduce business operation costs and improve customer satisfaction.

The research in this paper can be further extended, such as cases with more than three types of suborders and different types of vehicles. The research model in this paper is an offline scheduling problem, and future research can consider online problems.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant nos. 11871327 and 12071279 and the National Key Research and Development Program of China (No. 2020YFB1708200).

References

- [1] R. de Koster, T. Le-Duc, and K. J. Roodbergen, "Design and control of warehouse order picking: a literature review," *European Journal of Operational Research*, vol. 182, no. 2, pp. 481–501, 2007.
- [2] L. Ning, H. B. Zhang, and B. Zhang, "Optimizing the storage location assignment in a JIT-oriented parts distribution center," *Journal of Management Sciences in China*, vol. 17, no. 11, pp. 10–19, 2014, in Chinese.
- [3] F. Y. Chen, H. W. Wang, C. Qi, and Y. Xie, "Routing method for multiple order pickers with congestion consideration," *Journal of Systems Engineering*, vol. 28, no. 5, pp. 581–591, 2013, in Chinese.
- [4] X. P. Wang, J. Zhang, and J. Ma, "E-commerce on-line order batching model and algorithm: considering due time," *Journal of Management Science*, vol. 27, no. 6, pp. 103–113, 2014, in Chinese.
- [5] J. Zhang, F. Liu, and J. Tang, "The online integrated order picking and delivery considering Pickers' learning effects for an O2O community supermarket," *Transportation Research Part E: Logs and Transportation Review*, vol. 123, pp. 180–199, 2019.
- [6] M. Li, Y. H. Wu, N. N. Chen, and J. Zhang, "Items assignment optimization for array automated picking device based on serial order accumulation," *Computer Integrated Manufacturing Systems*, vol. 22, no. 9, pp. 2127–2134, 2014, in Chinese.
- [7] D. Q. Wang, "Research on optimizing distribution route of single-center logistics under new retail mode," *Journal of Fuzhou University (Natural Science Edition)*, vol. 47, no. 5, pp. 695–699, 2019, in Chinese.
- [8] L. H. Wu, *Research on Fresh Produce Distribution Mode Decision of Retailer's Online Channel*, Beijing Jiaotong University, Beijing, China, 2017, in Chinese.
- [9] X. P. Wang, J. Zhang, and C. Y. Yi, "Integrated scheduling of order picking and delivery under B2C E-commerce," *Chinese Journal of Management Science*, vol. 24, no. 7, pp. 101–109, 2016, in Chinese.
- [10] W. X. Peng, *Research on Fresh Produce Distribution Mode Decision of Retailer's Online Channel*, Wuhan University of Technology, Wuhan, China, 2017, in Chinese.
- [11] J. Zhang, X. Y. Zhang, and Y. F. Zhang, "A study on online scheduling problem of integrated order picking and delivery with multizone vehicle routing method for online-to-offline supermarket," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6673079, 2021.
- [12] N. G. Hall and C. N. Potts, "Supply chain scheduling: batching and delivery," *Operations Research*, vol. 51, no. 4, pp. 566–584, 2003.

- [13] Z. L. Chen, "Integrated production and outbound distribution scheduling: review and extensions," *Operations Research*, vol. 58, no. 1, pp. 130–148, 2010.
- [14] W. Y. Zhong, Z. L. Chen, and M. Chen, "Integrated production and distribution scheduling with committed delivery dates," *Operations Research Letters*, vol. 38, no. 2, pp. 133–138, 2010.
- [15] B. Fu, Y. Huo, and H. Zhao, "Coordinated scheduling of production and delivery with production window and delivery capacity constraints," *Theoretical Computer Science*, vol. 422, pp. 39–51, 2012.
- [16] C. A. Ullrich, "Integrated machine scheduling and vehicle routing with time windows," *European Journal of Operational Research*, vol. 227, no. 1, pp. 152–165, 2013.
- [17] F. S. Al-Anzi and A. Allahverdi, "An artificial immune system heuristic for two-stage multi-machine assembly scheduling problem to minimize total completion time," *Journal of Manufacturing Systems*, vol. 32, no. 4, pp. 825–830, 2013.
- [18] D. E. Goldberg, "Genetic algorithms in search," *Optimization, and Machine Learning*, Addison-Wesley Pub. Co., Boston, MA, USA, 1989.