

Research Article

Research on Dynamic Storage Location Assignment of Picker-to-Parts Picking Systems under Traversing Routing Method

Xiangbin Xu  and Chenhao Ren 

School of Transportation and Logistics, East China Jiaotong University, Nanchang 330013, China

Correspondence should be addressed to Xiangbin Xu; xuxiangbin@ecjtu.edu.cn

Received 15 July 2020; Revised 1 November 2020; Accepted 3 November 2020; Published 29 November 2020

Academic Editor: Michele Scarpiniti

Copyright © 2020 Xiangbin Xu and Chenhao Ren. 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.

Considering time-varying demand of online retail industry, the traditional static storage location assignment is converted into a multistage storage location assignment process based on the idea of gradual and small-step-forward optimization, which can respond to rapid changes in demand by adjusting the storage location of SKUs in the warehouse in real time and dynamically. First, the study formulates the framework dynamic storage location assignment. Then, the adjustment gain model of dynamic storage location assignment is built, and a genetic algorithm is designed to find the final adjustment solution. Finally, the computer program is developed to simulate the whole process. Simulation and data analysis results show that dynamic storage location assignment can effectively improve picking efficiency when the average order size is small and large demand correlation strength. Dynamic storage location assignment simplifies the warehouse operation process by combining the picking operation and storage location assignment into one without changing the picker's current walking route, which can offer some theoretical guidance for online retail enterprises implementing dynamic storage location assignment.

1. Introduction

Online retail enterprises should deliver commodities at a lower cost and a faster speed, and order picking is an important part of customer order delivery, which becomes the key to success for online retail enterprises. Picking systems can be divided into three classes: the picker-to-parts picking systems, the put systems, and the parts-to-picker systems [1]. Warehouses and distribution centers of large online retailers have begun to adopt the latter two picking systems, such as Jingdong and Amazon; however, most of small- and medium-sized retailing and logistics companies still use picker-to-parts picking systems. Order picking is in general the most laborious work of picker-to-parts picking systems, and its work labor accounts for more than 60% of the warehouse work [1, 2]. Storage location assignment is one of the important ways to improve the efficiency of picker-to-parts picking systems. Many studies research the problem of

storage location assignment based on SKU's popularity, turnover, and demand correlation [3–7], analyzing the demand pattern of SKUs by statistics, mathematical programming, or data mining-based method and optimizing the storage structure of SKUs once the SKUs are put away into shelves in the warehouse, which is a static storage location assignment problem and can be acceptable for the traditional warehouse picking systems of retail and industrial enterprises.

Online retail enterprises have the characteristics of short product life cycle, diversity of marketing strategy, and obvious seasonal demand. The SKUs' demand is high volatility, and their quantities and demand correlations change rapidly. Thus, the historical data of popularity and demand correlation of SKUs cannot reflect their dynamic demand in real time, which means the traditional storage location assignment method will lead to a sharp decrease in warehouse picking efficiency with the change of demands. In Dynamic

Storage Location Assignment (DSLAs), SKUs' demand is rapidly changing, and the location of SKUs in the warehouse should be adjusted in real time to adapt to this change [8]. DSLA transfers the traditional storage location assignment into multistage dynamic storage location assignment based on the step-by-step way, which optimizes SKU's storage location gradually instead of the traditional one-time optimization. It moves a small part of SKUs to meet the demand from one area of the warehouse to another periodically and gradually improve and optimize the storage structure of the warehouse.

Based on this, we propose DSLA framework, which considers the commonly time-varying demand in the online retail industry and transforms the traditional storage location assignment into a multistage, dynamic storage location assignment. DSLA considers both popularity and demand correlation of SKUs and makes small assignment of storage location of some SKUs to gradually improve and optimize the storage structure of the warehouse. Moreover, DSLA does not change the current picker's picking route of the picker-to-parts system, and it adjusts the storage location of SKUs while picking orders to respond to change of demands, which can reduce the picking cost and improve the customer purchasing experience.

The remainder of the paper is organized as follows. In Section 2, a review of relevant literature pertaining to DSLA is presented. We present the main description and introduce the assignment strategy of DSLA in Sections 3 and 4. The variable definition and indicators are constructed in Section 5. In Section 6, we build the dynamic adjustment plan and describe the proposed heuristic solution for DSLA. Experimental analyses and results are provided in Section 7. Finally, the main conclusions and future research are given in Section 8.

2. Literature Review

The research by Moon and Kim [9] showed that the location optimization effect was sensitive enough to the demand volatility. Under the random storage policy, the storage structure of SKU cannot match its demand when the demand volatility exceeded 60%, and the storage location must be adjusted. Kofler et al. [10] pointed out that if the storage location assignment was not carried out in real time when the demand changed rapidly, the storage location of all SKUs in the entire warehouse must be updated periodically (week/month) to meet the requirements of the change of SKUs. These research studies show that the traditional static storage location assignment cannot achieve the desired effect in a dynamic environment with rapid change. Ho [11] pointed out that DSLA was simpler and more effective than one-time long-term planning of storage location assignment in the case of frequent demand changes, which can not only improve the efficiency of picking but also improve customer response speed. Due to the high idle cost of the aisle-bound cranes of the AS/RS system, Christofides and Colloff [12] studied the DSLA problem of the AS/RS system very early, and many other studies carried out related research, but

there were few DSLA studies of picker-to-parts picking systems.

Both picker-to-parts picking and AS/RS systems optimize the storage structure of a warehouse by regularly adjusting the storage location of a small number of SKUs gradually, but the picker-to-parts picking system differs from the AS/RS system in the following aspects. First, the AS/RS system only adjusts SKUs within-aisle, while picker-to-parts picking system can adjust SKUs across-aisle even in the entire warehouse. Secondly, the AS/RS system only adjusts SKUs in idle time, but picker-to-parts picking system can adjust during the whole picking process. Finally, the AS/RS system only can reassign one pallet in one adjustment, while picker-to-parts picking system can assign multiple SKUs during the picking process. So, DSLA in the picker-to-parts picking systems could achieve better storage location optimization results.

3. DSLA Framework

Traditional warehouse slotting is performed when the SKUs are put on the shelves, and the storage location of the SKUs never changes until they are picked out. Different from that, the DSLA features the following two aspects. First, the storage location of the SKUs in the warehouse can be changed constantly and periodically, the SKUs with high popularity can be adjusted to a storage area near the depot, and the SKUs with correlation can be adjusted to the same storage location, which means the storage structure of the SKUs in the warehouse can dynamically respond to their demand variations in real time. Moreover, the slotting of DSLA is performed during the order picking process. The workflow of DSLA is described as follows.

At each picking wave time t ,

- (1) Route generation: the warehouse management system generates the wave picking list and the picker's walking route according to the preset routing method.
- (2) Candidate adjustment solution generation: find the candidate adjustment SKUs along the picker's walking route, which includes the SKUs that need to be adjusted, and the source and target storage locations of the adjustment based on the storage policy, the assignment strategy, and the demand pattern of SKUs.
- (3) Final adjustment solution generation: in picking wave time t , only a small part of the SKUs are selected to form the final adjustment solution by evaluating the candidate adjustment solutions. Final adjustment solution contains the SKUs to be adjusted and their source/target storage locations.
- (4) Perform adjustment: the final adjustment solution is merged into the picking task of the wave t and assigned to the picker. So, the task of the picker in the wave t includes order picking and storage adjustment. The picker adjusts the storage locations and picks orders simultaneously.

- (5) Information update: after the task of the wave t is completed, the warehouse management system updates the storage location and rolling popularity of the SKUs and then it enters the next picking wave time.

The whole process of DSLA is shown in Figure 1.

We can see from Figure 1 that DSLA requires picker to move SKUs between shelves during order picking, which brings additional working time to the picker (the walking distance is not changed because DSLA does not change the picker's walking route).

The guiding principle of DSLA is to obtain the maximum adjustment gain with the minimum adjustment costs. To be specific, the two key correlated issues that are involved in DSLA are "how to adjust" and "what to adjust." The former issue concerns the assignment strategy, that is, warehouse storage area planning and the way of moving SKUs. The latter issue studies how to build candidate adjustment solutions and find the final adjustment solution from them.

4. Assignment Strategy

The assignment strategy guarantees how these SKUs move in a rectangular one-block warehouse, as shown in Figure 2(a), and the strategy is closely related to the walking route of the picker. Here, we consider the most commonly used traversing (S-shaped) routing method in picker-to-parts picking system. When taking an S-shaped routing method, the picker enters from one end of the aisle where the SKUs are to be picked, picks up SKUs from the shelves on both sides of the aisle, and then leaves the aisle from the other end. Before returning to pickup/dropoff (P/D), the picker will traverse all the aisles where SKUs need to be picked. Considering that DSLA needs to move the SKUs with high popularity from areas far away from P/D to the area close to P/D, we propose a backward-traversing routing method, in which the picker first walks along the north cross to the last aisle containing SKUs to be picked and then picks SKUs from east to west under the S-shaped routing method. By taking the backward-traversing routing method, the picker can move the high popularity SKUs to the area near P/D in the warehouse during order picking, as shown in Figure 2(a).

In a popularity-based storage policy, SKUs are assigned to storage areas according to their popularity, which can reduce the picker's traveling time during order picking. Diagonal, within-aisle, across-aisle, and perimeter are the most commonly used warehouse storage policies in picker-to-parts picking systems, and the warehouse storage policy is correlated with the routing method, which was extensively studied by Petersen and Schmenner [13], and they found the within-aisle storage policy is the best storage policy together with the traversing routing method. Under the within-aisle storage policy, the SKU with highest popularity is assigned to the first storage of the first aisle, and the SKU with second highest popularity is assigned to the second storage of the first aisle, until the first aisle is completely assigned. The SKUs left will be assigned to the second aisle in the same way. The order of the storage assignment is shown in

Figure 2(c). The traversing routing method is taken by the picker in this paper, so within-aisle storage policy is the best choice for warehouse storage policy. The entire warehouse is divided into three zones: A-zone, B-zone, and C-zone according to the distance to P/D, A-zone near P/D stores the high popularity SKUs, and the low popularity SKUs are stored in the C-zone far away from P/D, as shown in Figure 2(b). The goal of DSLA is to move the high popularity SKUs from B-zone or C-zone to A-zone periodically when the SKUs' demand changes rapidly.

5. Adjustment Solution Construction

The goal of adjustment solution construction is to find the SKUs that need to be adjusted and the source/target locations of the adjustment, which includes two steps: candidate adjustment solution generation and finding the final adjustment solution from them.

5.1. Candidate Adjustment Solution Generation. In DSLA, the picker adjusts the storage locations of SKUs during order picking without changing his/her walking route, so the candidate adjustment SKUs should satisfy the following. (1) The storage locations of the candidate adjustment SKUs are on the picker's walking route; (2) the storage zones of the candidate adjustment SKUs do not match demand patterns (high popularity SKUs being stored in B-zone or C-zone).

The candidate target storage locations should meet the following three criteria. (1) Target storage locations are on the picker's walking route; (2) target storage locations are not full; and (3) target storage locations are in A-zone.

5.2. Final Adjustment Solution

5.2.1. Variable Definition. We use the following notations in this paper:

Indices:

- k : for storage location
- i, j : for SKU
- t : for picking wave
- r : for order
- s : for item set size

Parameters:

- Φ = set of candidate adjustment SKUs
- Ω = set of target storage locations
- Φ_k = set of SKUs currently stored in location k
- Curr_k = total number of SKUs currently stored in location k
- Curr_k^j = number of SKU j currently stored in location k
- Cap_k = capacity of location k
- V_t = remaining capacity of the picking cart
- $\text{Pop}_i(t)$ = popularity of SKU i in wave time t
- Rolling Pop_i = rolling popularity of SKU i
- $f(i)$ = demand frequency of SKU i

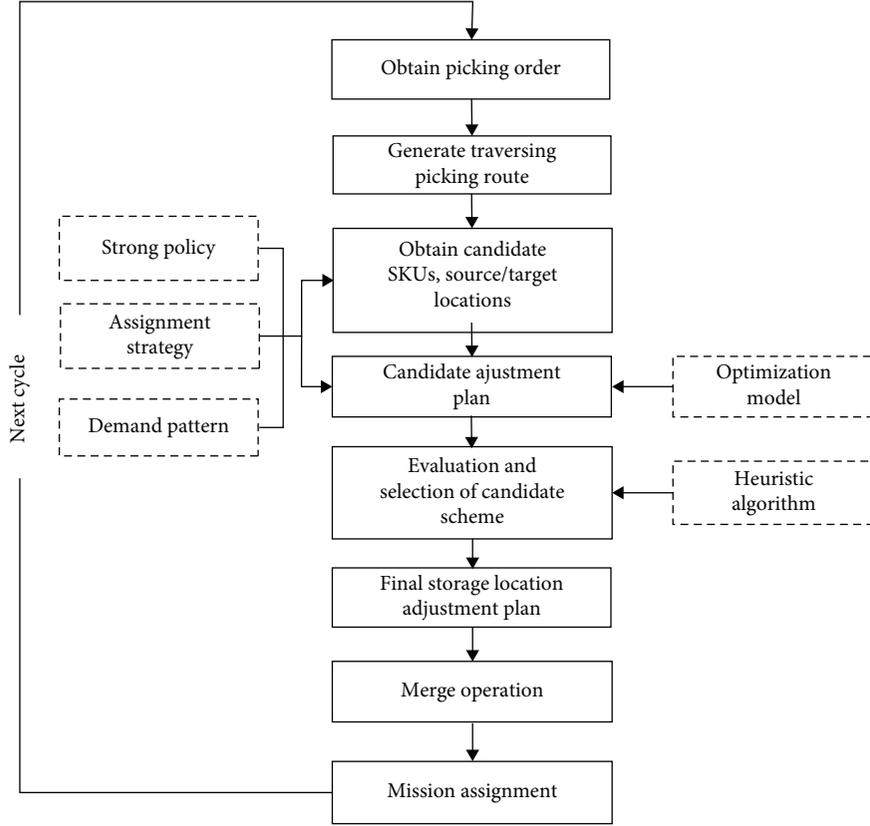


FIGURE 1: Model of DSLA.

$r(i, j)$ = strength of demand correlation between SKU i and SKU j

$d(i, k)$ = saving distance of adjusting SKU i to the target location k

N_i = adjustment quantity of SKU i

$z_{i,k}$ = adjustment gain of adjusting SKU i from its current storage location to target location k

R = set of orders, where each order r is a set of SKUs

$\sigma(i)$ = number of orders which contains SKU i

C_s = candidate item set of size s

L_s = frequent item set of size s

5.2.2. Rolling Popularity and Strength of Demand Correlation. DSLA not only adjusts the SKUs with high popularity to the storage zone near the P/D, and it also adjusts the SKUs with demand correlation to the same or adjacent storage locations. Considering the variation of SKU's popularity, we define the rolling popularity of an SKU as the moving average of its demand within a given period, which is calculated accordingly to

$$\text{Rolling Pop}_i = \sum_{t'=t-\Delta t}^{t-1} \text{Pop}_i(t'). \quad (1)$$

Considering the cold start of the DSLA, we do not adjust the storage location in the first 99 picking waves, but only calculate the popularity of SKU i ($\text{Pop}_i(t)$) in each picking wave t . We begin adjust and calculate the rolling popularity

of SKUs according to formula (1) from the 100th picking wave. After calculating the rolling popularity of all candidate SKUs, we also calculate the demand frequency of all candidate SKUs. The demand frequency of the candidate SKU i ($f(i)$) is calculated by

$$f(i) = \frac{\text{Rolling Pop}_i}{\sum_{j \in \Phi} \text{Rolling Pop}_j}. \quad (2)$$

In addition to adjusting the SKUs with high popularity, DSLA also adjusts the SKUs with demand correlation to the same or adjacent storage locations. The demand correlation between SKUs is generated by the Apriori algorithm which uses support and confidence to measure the demand correlation of SKUs [14], and SKUs with support and confidence greater than the minimum support and confidence have strong demand correlation. The workflow of Apriori is described as follows:

- (1) Scan each order r in R , count SKUs to generate the candidate item sets C_1
- (2) New frequent item set L_1 is generated by SKUs from C_1 with support greater than the minimum support
- (3) Set $s = 1$
- (4) Candidate item set C_{s+1} is obtained by L_s self-connection
- (5) Count all candidates SKUs in C_{s+1} that are contained in R , and frequent item set L_{s+1} is generated from C_{s+1} with support greater than the minimum support

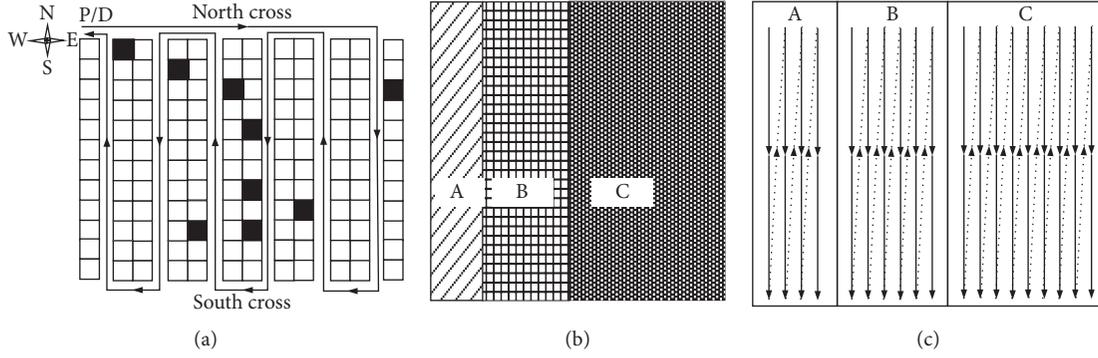


FIGURE 2: Assignment strategy of DSLA: (a) traversing routing method, (b) within-aisle storage, and (c) order of the storage assignment.

- (6) If $L_{s+1} \neq \emptyset$, $s = s + 1$, go to (4); otherwise, return frequent item set L_{s+1} .

The support represents the frequency at which both SKUs i and j appear simultaneously in one order, and the support degree $\text{Sup}(i, j)$ is calculated as

$$\text{Sup}(i, j) = \frac{\sigma(i \cup j)}{|R|}, \quad (3)$$

where $\sigma(i \cup j)$ is the number of orders that contain SKU i and j , R is the number of all orders, and the confidence $\text{Conf}(i, j)$ measures the frequency at which SKU j appears in orders containing SKU i , which is calculated as

$$\text{Conf}(i, j) = \frac{\sigma(i \cup j)}{\sigma(i)}. \quad (4)$$

Here also introduce lift to measure the strength of demand correlation between SKUs, and if lift is greater than 1, it means SKUs are positively correlated, that is, the demands between the two SKUs are mutually promoted. The greater the lift, the stronger the demand correlation between SKUs. The lift is equal to 1 means the demands of the SKUs are independent of each other, and lift is less than 1 means the demands of the SKUs are complementary. In view of this, we combine support and lift to describe the strength of demand correlation between SKUs.

Once the support degree $\text{Sup}(i, j)$ and the lift degree $\text{Lift}(i, j)$ between candidate SKU i and j are calculated by the Apriori algorithm, the strength of demand correlation $r(i, j)$ between SKU i and j is calculated by

$$r(i, j) = Lw(i, j) \times \text{Sup}(i, j), \quad \begin{cases} Lw(i, j) = \text{Lift}(i, j), & \text{if } \text{Lift}(i, j) > 1, \\ Lw(i, j) = 0, & \text{if } \text{Lift}(i, j) = 1, \\ Lw(i, j) = -\text{Lift}(i, j), & \text{if } \text{Lift}(i, j) < 1. \end{cases} \quad (5)$$

When $\text{Lift}(i, j) > 1$, the greater $r(i, j)$ is, the greater the optimization effect can be obtained if both SKU i and j are adjusted to the same or adjacent storage locations.

The random storage policy is widely used in warehouses of online retail companies, which means there is more than one category of SKUs stored in the same location. So, if we move SKU i to the target location k , all SKUs stored in target location k may have demand correlation with SKU i and can benefit from the adjustment. Considering this, we propose SKU-to-Bin Affinity (Sba) to calculate the benefit of the adjustment of moving SKU i to target location k . We define $\text{Sba}_{i,k}$ as the average strength of demand correlation of SKU i , and all SKUs are stored in target location k (Φ_k), which is calculated as

$$\text{Sba}_{i,k} = \sum_{j \in \Phi_k} \frac{r(i, j) * (\min(N_i, \text{Curr}_k^j) / N_i)}{|\Phi_k|}, \quad (6)$$

where N_i is the adjustment quantity of SKU i , Φ_k is the set of SKUs currently stored in location k , and Curr_k^j is the number of SKU j stored in location k .

5.2.3. Route-Based Saving Distance. DSLA can save picker's walking distance, since the picker in picker-to-parts warehouse systems may take different routing methods while picking, such as traversing, return, midpoint, and the largest gap [15]. The saving distance of DSLA is closely related to the routing methods, which means that even if the SKU is moved from the same source location to the same target location, the saved walking distance under different routing methods may be different. Considering this, the Route-Based Saving Distance (Rbsd) is proposed to represent the saved walking distance of DSLA, where $\text{Rbsd}(i, k)$ is proposed to represent the walking distance saved by moving SKU i from its current storage location to the target location

k . $Rbsd(i, k)$ with the traversing routing method is shown in Figure 3.

The locations marked with black in Figure 3 indicate the storage locations of the SKUs to be picked. When picker picks up these SKUs using the backward-traversing routing method, DSLA first generates the walking route of the picker and calculates the rolling popularity. If DSLA finds that SKU i (marked with the black grid line) on the current walking route is a high popularity SKU but it is stored in an area far away from P/D, then it needs to be moved to the location k (marked with the white grid line) in the storage zone near P/D. Since the picker will traverse all aisles containing the SKUs to be picked, moving SKU i can save picker's walking distance in the horizontal direction. The calculation of $Rbsd(i, k)$ is

$$Rbsd(i, k) = (u - v) * W_a, \quad (7)$$

where u is the number of the aisle where SKU i is currently stored, v is the number of the aisle of the target location k , and W_a is the distance between the center lines of adjacent aisles.

5.2.4. Adjustment Gain. The adjustment gain is calculated based on the following aspects. (1) The demand frequency of SKU: the greater it is, the greater the adjustment gain can be obtained when the SKU is adjusted from B-zone or C-zone to A-zone. (2) The strength of demand correlation between SKUs: greater adjustment gain can be obtained when highly correlated SKUs are adjusted to the same or adjacent storage locations because they can be picked up together. (3) $Rbsd$: larger $Rbsd$ means more walking distance can be saved by the adjustment, so greater adjustment gain can be obtained.

Taking the three aspects above into consideration, we define $z_{i,k}$ as the adjustment gain of moving SKU i from its current location to target location k , which is calculated as

$$z_{i,k} = \sum_{k \in \Omega} [\zeta * (1 - \partial) * Sba_{i,k} + \partial * f(i)] * N_i * Rbsd(i, k). \quad (8)$$

The first part in formula (8) $(1 - \partial) * Sba_{i,k} * N_i * Rbsd(i, k)$ represents the walking distance saved by adjusting demand-correlated SKUs together, and the second part $\partial * f(i) * N_i * Rbsd(i, k)$ represents the walking distance saved by adjusting high popularity SKUs to A-zone. The two parts are combined into $z_{i,k}$ by the weight ∂ ($0 \leq \partial \leq 1$) between $[0, 1]$. ζ is a coefficient used for normalizing the two parts according to [16], which is calculated as

$$\zeta = \frac{\sum_{i \in \Phi} \sum_{k \in \Omega} Sba_{i,k}}{\sum_{i \in \Phi} f(i)}. \quad (9)$$

5.2.5. Final Adjustment Solution Model. There are many candidate adjustment solutions in each picking wave, considering the remaining capacity of the picking cart and rapid change of online retailer's demand. DSLA only selects a small number of SKUs from the candidate adjustment solutions to adjust locations. So, the final adjustment solution can be obtained by solving the model formulated as (10)–(14):

$$Z = \sum_{i \in \Phi} \sum_{k \in \Omega} [\zeta * (1 - \partial) * Sba_{i,k} + \partial * f(i)] * X_{i,k} * Rbsd(i, k), \quad (10)$$

$$\begin{aligned} & \text{Max}(Z) \\ & \text{subject to } \sum_{k \in \Omega} X_{i,k} \leq 1, \quad \forall i, \end{aligned} \quad (11)$$

$$\sum_{i \in \Phi} X_{i,k} \leq \text{Cap}_k - \text{Curr}_k, \quad \forall k, \quad (12)$$

$$\sum_{i \in \Phi} \sum_{k \in \Omega} X_{i,k} \leq V_t, \quad (13)$$

$$X_{i,k} \in \{0, 1\}, \quad \forall i, k. \quad (14)$$

The objective of DSLA is to maximize total adjustment gain Z in formula (10) by selecting appropriate SKUs from candidate adjustment solutions. Constraint (11) represents that the SKU i can only be adjusted to one target location at most. Constraint (12) guarantees that the capacity of the target storage location k does not exceed its maximum capacity. Constraint (13) ensures that the total of SKUs to be adjusted in a picking wave does not exceed the remaining capacity of the picking cart. The $X_{i,k}$ in formula (14) is a binary variable indicating whether the SKU i is adjusted to the target location k or not. By solving the model described by formulae (10)–(14), the final storage adjustment solution of each picking wave can be found by DSLA.

6. Algorithm

The candidate adjustment solutions are combinations of the set of all candidate SKUs Φ and the set of target storage locations Ω , and the number of combinations is huge, and finding the final adjustment solution described by formulae (10)–(14) can be NP-hard. Metaheuristics algorithm can be used to find the best adjustment solution in an acceptable time, where we use the genetic algorithm to solve the model described by formulae (10)–(14).

In the proposed genetic algorithm, each candidate adjustment solution is represented by a chromosome χ and its gene is a triple (i, k, u) , which is shown in the following:

$$\chi = (2, 4, 1), (18, 6, 1), (2, 8, 1), (2, 6, 1), (34, 8, 0), (2, 12, 1), \dots, (50, 6, 1). \quad (15)$$

In each gene triple (i, k, u) , index i is SKU to be adjusted, k is the target storage location to be adjusted to, and u is

binary variable indicating whether the SKU is to be moved to storage location k or not; if u is 1, SKU i will be moved to the

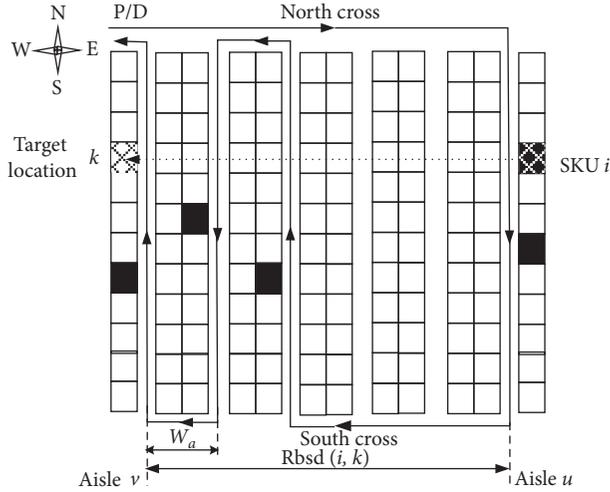


FIGURE 3: Calculation of route-based saving distance.

target storage location k ; otherwise, SKU i stays in the location where it is currently located.

The length of the chromosome is calculated as $|\chi|$, and genetic operators such as crossover and mutation are carried out following the idea of multiple bin-size bin packing problem (MBSBPP) [17]. The SKU i and the target storage location k to be adjusted are regarded as item and bin in MBSBPP, respectively, while crossover and mutation only change the target location k , which is described as follows.

Firstly, the fitness values of all chromosomes are calculated by formula (16); then, the roulette selection algorithm is used to select two parent chromosomes according to their fitness values and target storage location k in each triple (i, k, u) of the two parent chromosomes is extracted to form two target location sequences. After that, a random number is generated between $[0, 1]$ as the crossover probability P_c ; if the random probability is greater than P_c , an integer between $(0, |\chi|)$ is randomly generated as the crossover point and two new target storage locations are generated by exchanging gene section after the crossover. Mutation of two new target storage locations is performed by two-point inversion: a random number is generated between $[0, 1]$ as the mutation probability P_m ; if the random probability is greater than P_m , two integers between $(0, |\chi|)$ are randomly generated as the mutation points, and the target storage locations between these two variation points are reversed.

Finally, the two target location sequences obtained after the crossover and mutation are assigned to the original parent chromosomes, respectively, and obtaining two children chromosomes.

Considering constraints (11)–(13) in the adjustment gain model, many infeasible adjustment solutions may be generated after the crossover and mutation. For example, a chromosome containing gene triples $(2, 4, 1)$, $(2, 8, 1)$, and $(2, 12, 1)$ is not feasible because SKU 2 cannot be moved the target locations 4, 8, and 12 simultaneously, which violates constraint (11). In addition, a chromosome containing gene triples $(18, 6, 1)$, $(2, 6, 1)$, and $(50, 6, 1)$ is not feasible because moving SKUs 18, 2, and 50 to the same target storage

location 6 may exceed the remaining capacity of the storage location 6, which violates constraint (12). Moreover, the pick cart capacity limitation should be considered (the constraint condition (13) is satisfied), so we should do some extra work after crossover and mutation:

- (1) Making all chromosomes feasible: enumerate all the genes in the chromosomes including the SKU i to be adjusted. Calculate the adjustment gain according to formula (8). Set u of the gene with the largest adjustment gain as 1 and all the others as 0 so that constraint (11) is satisfied.
- (2) Making all target locations' capacity feasible: sort the adjusted chromosomes in the descending order according to the adjustment gain, and select them one by one until remaining capacity of the target location is not enough. The selected gene triples' u are set to 1, otherwise set to 0 so that constraint (12) is satisfied.
- (3) Making picking cart's capacity feasible: we use a penalty function to calculate the chromosome fitness value. The fitness value of chromosome χ is calculated by

$$F(\chi) = p(\chi) * Z(\chi), \quad (16)$$

where $Z(\chi)$ is the gain of the adjustment solution described by chromosome χ , which is calculated by formula (8), and $p(\chi)$ is the penalty function of the infeasible chromosome χ , which is calculated as

$$p(\chi) = 1 - \frac{|\sum_{i=1}^n u_i - V_t|}{\delta}. \quad (17)$$

Summation $\sum_{i=1}^n u_i$ is the total number of adjustment SKUs, V_t is the remaining capacity of the picking cart, and the coefficient δ is calculated as

$$\delta = \max\left(V_t, \sum_{i=1}^n u_i\right). \quad (18)$$

We can see from (17) and (18) that the candidate adjustment are not punished only when $|\sum_{i=1}^n u_i - V_t| = 0$ ($p(\chi) = 1$), which means whether the total number of the adjusted SKUs is greater or less than the remaining capacity of the picking cart, and the candidate adjustment will be punished. Its purpose is to make full use of the remaining space of picking cart for storage location assignment.

7. Experimental Results

This section presents the experimental analysis and results. The experimental factors and level values are shown in Table 1, and our experiment has three factors, which are average order size, strength of demand correlation, and adjustment quantity of SKU. Average order size has four levels, which are 5, 10, 15, and 20. Strength of demand correlation measures the degree of demand correlation between SKUs and has four levels, which are 2, 4, 8, and 10

TABLE 1: Summary of the factorial design of experiments.

Factors	Level values
Average order size (Aos)	5, 10, 15, 20
Strength of demand correlation (Sdc)	2, 4, 8, 10
Adjustment quantity (Aq)	2, 4, 8, 12

representing the demand correlation between 2% to 10% of SKUs, respectively. Adjustment quantity of SKU specifies maximum number of SKU allowed to be adjusted in each picking wave, including 2, 4, 8, and 12. The warehouse has 10 two-sided aisles, and the shelves on both side of these aisles consist of 3 floors, and each floor has 20 storage locations. The widths of the aisle and the storage location are 200 cm and 80 cm, respectively. The capacities of the storage location and the picking cart are 60 and 30, respectively, that is, the picking cart can accommodate up to 30 SKUs, including the SKUs both for picking out and adjustment. The system generates an order every 12 seconds and works 10 hours a day, so the total number of orders in one day is 3000. The order generation process is described as follows:

- (1) Generate an empty order $O =$ and a temporary variable $n = 0$ to record the number of SKUs in O
- (2) Generate order size: order size N is generated according to the normal distribution model with mean value of average order size and coefficient of variation of 0.1
- (3) Generate next SKU: generate an SKU i according to the demand pattern of the experimental scenario
- (4) If $n + 1 > N$, return O ; otherwise, add i to O and let $n = n + 1$
- (5) Generate correlated SKU: generate a SKU j related to SKU i according to the strength of demand correlation
- (6) If $n + 1 > N$, return O ; otherwise, add j to O and let $n = n + 1$, and go to (3)

Once order is generated, the DSLA will do the following:

- (1) Build the picker's walking route
- (2) Calculate the rolling popularity Rolling Pop
- (3) Mine demand pattern and calculate strength of demand correlation r by Apriori algorithm
- (4) Calculate demand frequency f of SKUs
- (5) Calculate SKU-to-Bin affinity S_{ba} and saving distance R_{bsd}
- (6) Obtain candidate adjustment SKUs Φ and candidate target storage locations Ω

Then, DSLA's adjustment gain optimization model is built, and final adjustment solutions are solved through the genetic algorithm. Final adjustment solutions are simulated with agent-based model implement by Java. After all orders within one day were picked out, DSLA automatically calculates average walking distance and average working time

of all pickers (which represent the working intensity and efficiency of all pickers, respectively). DSLA simulation flowchart and software are shown in Figure 4.

The total working time t_{total} of the picker, including walking time, picking time, and temporary pulling off time and putting away time of the adjustment SKUs, is calculated by

$$t_{\text{total}} = \frac{l}{v_w} + t_p * n_p + t_d * n_a + t_u * n_a, \quad (19)$$

where v_w is the walking speed of the picker, t_p , t_d , and t_u are the times taken for picking, pulling off, and putting away of a SKU, respectively, and n_p and n_a are the SKU quantities to be picked and adjusted, respectively. The walking speed of the picker (v_w) is 0.75 m/s, and the picking time of one SKU (t_p) is 18 seconds, and we let t_d and t_u be the same as the picking time t_p .

Our experiment has 64 ($4*4*4$) scenarios. In each scenario, DSLA generated 3000 orders to simulate the demand in one day, each experimental scenario was repeated 3 times to reduce random error, and a total of 192 ($64*3$) data points were obtained. SPSS21.0 is used to analyze the experimental data and obtained the ANOVA with the significance level 0.05, as shown in Table 2. Table 2 tells us the significance values of the three principal factors (average order size, strength of demand correlation, and adjustment quantity) are all less than 0.05, which means three principal factors affect the optimization effect directly. In addition, the significance values of the interaction between the three principal factors are also less than 0.05, indicating that the interactions between them also affect the optimization effect of DSLA.

Figure 5 shows the general optimization effect of DSLA, including the saved walking distance and working time. We can see from Figure 5 that average walking distance and working time of all pickers in one day decreases by 7.67% and 2.60%, respectively. It can be concluded that DSLA can not only reduce the picker's working intensity but also improve the efficiency effectively.

The ANOVA in Table 2 shows that average order size, strength of demand correlation, and adjustment quantity are the key factors affecting the optimization effect of DSLA. The three factors are taken as the main effects to further analyze the optimization effect of DSLA in the following, as shown in Figure 6.

Figure 6(a) represents the optimization effect of DSLA under different order sizes. The X-axis indicates the average order size, and Y-axis indicates the adjustment gain. We can see from Figure 6(a) that the optimization effect of DSLA decreases with the increase of the average order size, that is, the average walking distance and working time of the pickers are saved by 10.48% and 3.96%, respectively. Particularly, when the average order size is 20, the average walking distance and working time of the pickers are saved by 5.0% and 0.08%, respectively, and the effect of DSLA is not obvious. There are two main reasons: the first one is the capacity limitation of the picking cart, and the smaller order size means that more SKUs can be adjusted during order

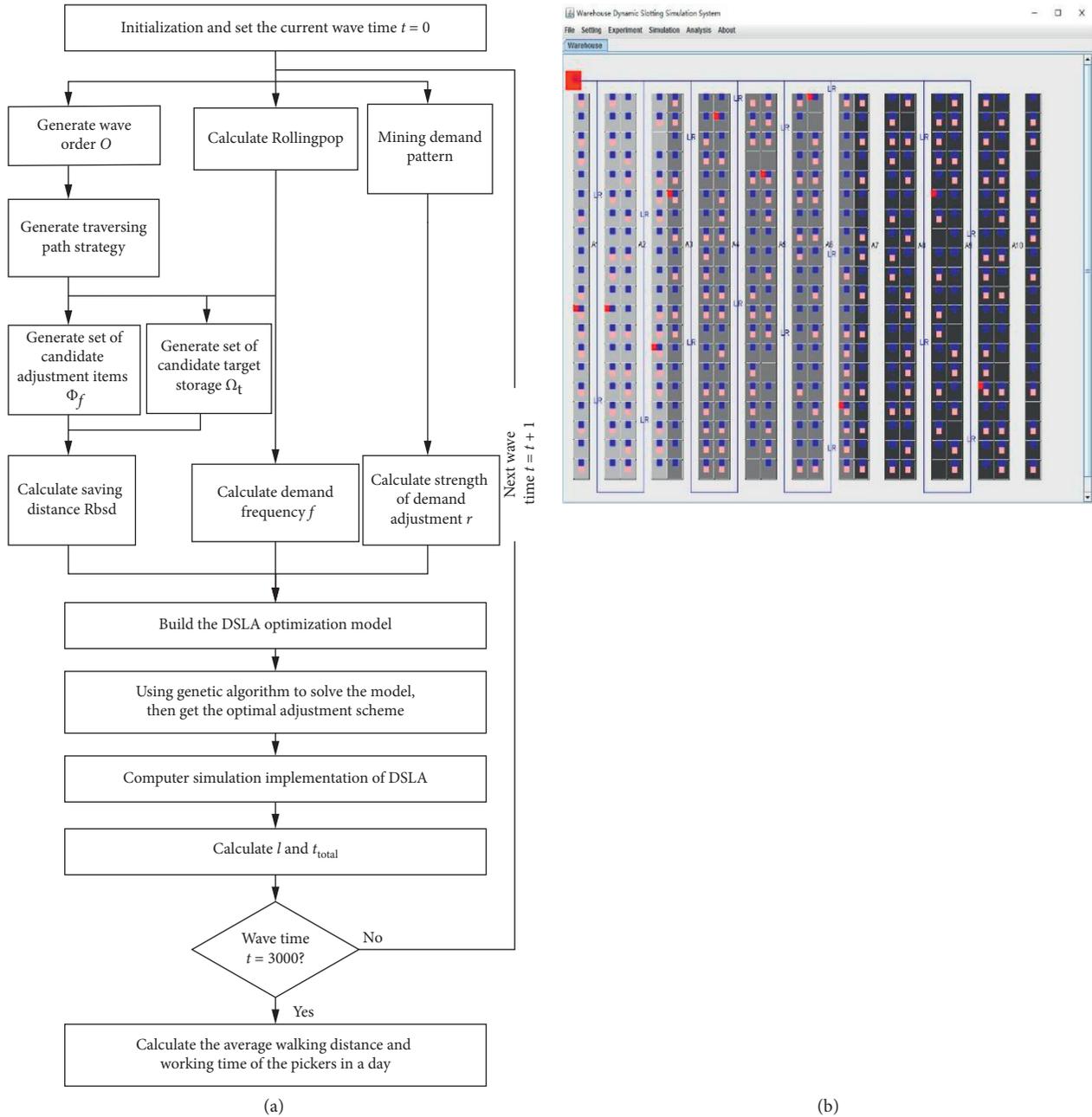


FIGURE 4: DSLA simulation flowchart and software: (a) DSLA simulation flowchart and (b) DSLA simulation software.

picking. Moreover, we find in simulation that SKUs to be picked are almost distributed in all picking aisles, and pickers need to transverse all aisles when the average order size is larger; in this case, DSLA cannot reduce the number of aisles that the picker needs to transverse when picking, DSLA's optimization is not obvious. So, it can be concluded that the optimization effect of DSLA decreases with the increase of the average order size when the pickers take the traversing routing method to pick SKUs, and the optimization effect of DSLA is best when the average order size is smaller.

Figure 6(b) describes the optimization effect of DSLA under different strengths of demand correlation. The X-axis

indicates the strength of demand correlation, and the Y-axis indicates the adjustment gain. We can see from Figure 6(b) that the optimization effect increases when the strength of demand correlation increases. When strength of demand correlation is equal to 10 (which means 10% SKUs' demand are correlated), the average walking distance and working time of the pickers are reduced up to 10.93% and 3.78%, respectively. This is because the stronger the SKUs' demand correlation is, the more likely these SKUs will be adjusted to the adjacent or same storage locations; moreover, SKUs with high popularity are also adjusted to the storage area near P/D together with their correlated SKUs, which further saves the walking distance and working time of the pickers. So, it can

TABLE 2: Significance tests of optimization effect of DSLA.

Source		III	Df	MSE	F	Sig.
Aos	Saving distance	0.112	3	0.037	365.142	0.000
	Saving time	0.026	3	0.009	372.428	0.000
Sdc	Saving distance	0.110	3	0.037	359.680	0.000
	Saving time	0.015	3	0.005	207.273	0.000
Aq	Saving distance	0.114	3	0.038	372.606	0.000
	Saving time	0.016	3	0.005	226.569	0.000
Aos*Sdc	Saving distance	0.008	9	0.001	8.785	0.000
	Saving time	0.002	9	0.000	8.427	0.000
Aos*Aq	Saving distance	0.013	9	0.001	14.112	0.000
	Saving time	0.003	9	0.000	14.239	0.000
Sdc*Aq	Saving distance	0.030	9	0.003	32.831	0.000
	Saving time	0.004	9	0.000	20.245	0.000
Error	Saving distance	0.013	128	0.000		
	Saving time	0.003	128	2.352E - 05		
Sum	Saving distance	1.532	192			
	Saving time	0.199	192			

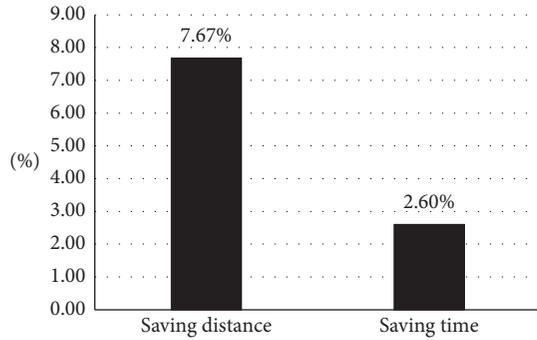


FIGURE 5: General optimization effect of DSLA.

be concluded that the optimization effect of DSLA increases with the increase of strength of demand correlation when the pickers take the traversing routing method to pick SKUs.

Figure 6(c) shows the optimization effect of DSLA under different adjustment quantities of SKUs. When the adjustment quantity of SKUs is large (12), DSLA can save the average walking distance and average working time up to 10.29% and 3.60%, respectively. On the contrary, when the adjustment quantity of SKUs is small (2), DSLA can save the average walking distance and average working time only by 3.93% and 1.21%. The reason behind that is straightforward, large adjustment quantity of SKUs can bring more candidate adjustment solution, and DSLA can select optimal adjustment solution to implement. So, it seems to be that the optimization effect of DSLA increases with the increase of the adjustment quantities of SKUs when the pickers take the traversing routing method to pick SKUs. However, we must keep in mind that the basic idea of DSLA is to dynamically optimize the storage structure of the warehouse by regularly adjusting the storage locations of a small number of SKUs, in response to the rapid changes of SKUs' demand in real time. When SKUs' demand changes rapidly, the optimization effect is not obvious when the number of adjustment SKUs is

small enough, and if the number of adjustment SKUs is too large, they will move back and forth periodically [18], so DSLA must determine the optimal number of SKUs to be adjusted under different demand patterns. The significance tests in Table 2 also show that the significance values of the interactions between strength of demand correlation, average order size, and adjustment quantity of SKUs are also less than 0.05, indicating that the interactions between them have important impact on the optimization effect of DSLA. In view of this, the interaction effects between the adjustment quantity of SKUs and the other two factors are further analyzed, as shown in Figures 7 and 8.

Figures 7 and 8 show the co-optimization effects between adjustment quantity and strength of demand correlation and adjustment quantity and average order size. In the case of high strength of demand correlation or small average order size, Figure 7 tells us that, although appropriately increasing adjustment quantity of SKUs can significantly improve the optimization effect of DSLA, the marginal utility of the adjustment gain by adjusting the quantity of SKUs is decreasing. In other words, when the quantity of adjustment SKUs increases to a certain extent, DSLA could not improve the optimization effect anymore. Moreover, we can see from Figure 7 that when the strength of demand correlation is small, increasing the quantity of adjustment SKUs reduces the optimization effect. Simulation results show that the main reason for this is that an increase in the adjustment quantity of SKUs leads to frequent movement of SKUs. So, we can draw a conclusion that DSLA is suitable for optimization under the circumstance of strong strength of demand correlation or small average order size. Due to the dynamic changes of demands, it is not advisable to increase the adjustment quantity of SKUs unreasonably when implementing DSLA, and the principle of DSLA is gradual optimization in small steps.

In summary, simulation and numerical analysis show that when the pickers perform dynamic storage location assignment without changing the current picking route,

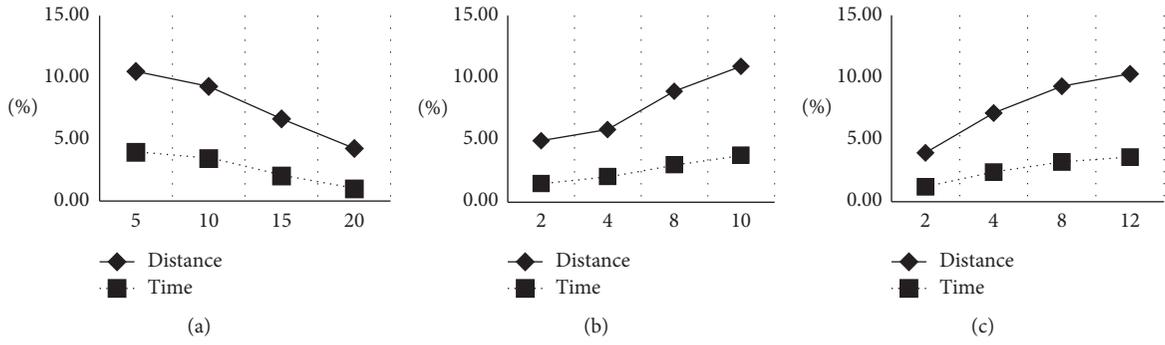


FIGURE 6: Optimization effect of DSLA: (a) average order size, (b) strength of demand correlation, and (c) adjustment quantity.

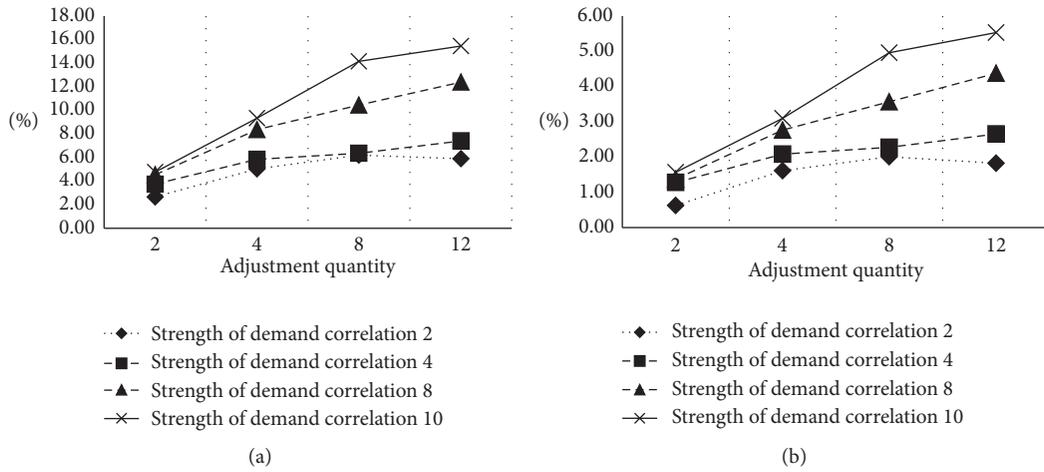


FIGURE 7: Co-optimization effect of adjustment quantity and strength of demand correlation: (a) saving distance and (b) saving time.

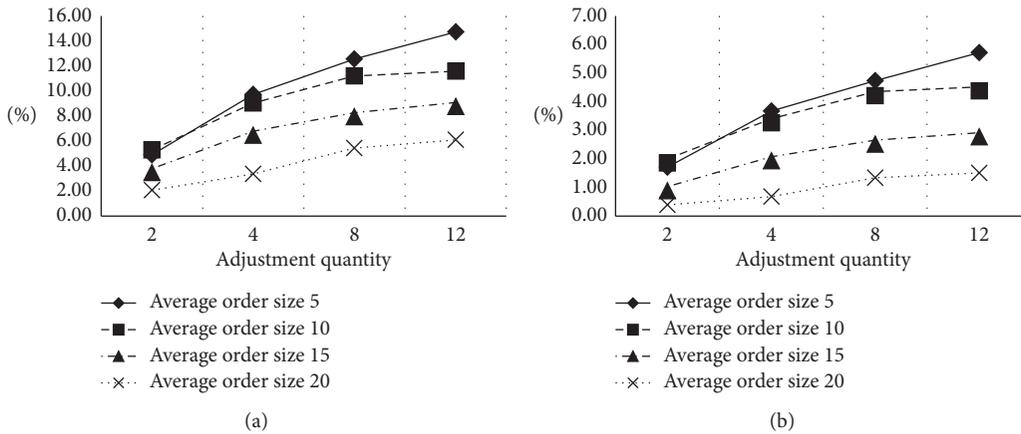


FIGURE 8: Co-optimization effect between adjustment quantity and average order size: (a) saving distance and (b) saving time.

the average walking distance and average working time of all pickers can be effectively saved, thereby greatly improving the efficiency of warehouse picking. DSLA is most suitable for the picking system with smaller average order size and stronger demand correlation; when the order size is larger and the strength of demand

correlation is relatively higher, DSLA can save walking distance and time of all pickers dramatically. If demand changes rapidly, DSLA should follow the principle of gradual optimization with multibatch and small-batch adjustment. Online retailers often carry out marketing strategies such as category promotion and cross-selling,

resulting in rapid changes in SKUs and their quantities, which means stronger strength of demand correlation and relatively smaller average order size is common phenomenon in online retail enterprises, so the conclusions we found can offer certain guidance for improving the warehouses picking efficiency of online retailers.

8. Conclusions and Future Research

DSLAs divide the traditional one-time storage location assignment into a dynamic, small-batch storage location assignment process by considering SKUs' popularity and demand correlation. Numerical experiments show that, compared with the traditional storage location assignment method, DSLA can effectively save the walking distance and working time of the pickers, and the storage location optimization effect is obvious. It is especially suitable for the optimization of the picking system of online retailers with rapid changes of SKUs and high demand uncertainty, which can bring improvement on online retailers' logistics cost and customer purchasing experience; in addition, DSLA combines order picking and storage location assignment into one working process and only adjusts the location of a small quantity of SKUs while picking, and it is feasible and very easy for online retailers' picker-to-parts picking systems to implement DSLA.

In addition to the traversing routing method, picker-to-parts picking systems also have return, midpoint, largest gap, and shortest methods. Different methods correspond to different optimal storage policies. The optimization effect of storage location assignment is closely related to the routing method, and further research can take these aspects into consideration. Finally, we do not consider the walking route changes of the pickers. Following research can study the DSLA under the variable route, which allows the pickers to change the current walking route appropriately for dynamic storage location assignment.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors wish to acknowledge the financial support of the National Natural Science Foundation of China, under Grant no. 71761013 and Natural Science Foundation of Jiangxi Province, China, under Grant no. 20181BAB201010.

References

[1] R. D. 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] F. Chen, H. Wang, Y. Xie, and C. Qi, "An ACO-based online routing method for multiple order pickers with congestion consideration in warehouse," *Journal of Intelligent Manufacturing*, vol. 27, no. 2, pp. 389–408, 2016.
- [3] X. Jian and Z. Li, "A correlated storage location assignment problem in a single-block-multi-aisles warehouse considering BOM information," *International Journal of Production Research*, vol. 48, no. 5, pp. 1321–1338, 2010.
- [4] Y. D. Li, "Model and algorithm for cartonization and slotting optimization simultaneously in wave-picking zone-based system," *Systems Engineering—Theory & Practice*, vol. 33, no. 5, pp. 1269–1276, 2013.
- [5] Y. D. Li, J. S. Lu, and G. Q. Pan, "Slotting optimization based on SKUs correlations with serpentine routing policy," *Journal of Zhejiang University (Engineering Science)*, vol. 46, no. 8, pp. 1424–1430, 2012.
- [6] Y. Zhang, "Correlated storage assignment strategy to reduce travel distance in order picking*Zhang thanks the financial support of the China scholarship council (CSC)," *IFAC-PapersOnLine*, vol. 49, no. 2, pp. 30–35, 2016.
- [7] M. E. Fontana and V. S. Nepomuceno, "Multi-criteria approach for products classification and their storage location assignment," *International Journal of Advanced Manufacturing Technology*, vol. 88, no. 9–12, pp. 1–12, 2016.
- [8] J. A. Pazour and H. J. Carlo, "Warehouse reshuffling: insights and optimization," *Transportation Research Part E: Logistics and Transportation Review*, vol. 73, pp. 207–226, 2015.
- [9] G. Moon and G. P. Kim, "Effects of relocation to AS/RS storage location policy with production quantity variation," *Computers & Industrial Engineering*, vol. 40, no. 1–2, pp. 1–13, 2001.
- [10] M. Kofler, A. Beham, S. Wagner, M. Affenzeller, and W. Achleitner, "The multi-period storage location assignment problem," in *Proceedings of the IEEE Apcast'12 Conference*, pp. 38–41, Sydney, Australien, February 2012.
- [11] S. Ho, *Intentional Fragmentation for Material Storage*, Massachusetts Institute of Technology, Cambridge, MA, USA, 2004.
- [12] N. Christofides and I. Colloff, "The rearrangement of items in a warehouse," *Operations Research*, vol. 21, no. 2, pp. 577–589, 1973.
- [13] C. G. Petersen and R. W. Schmenner, "An evaluation of routing and volume-based storage policies in an order picking operation," *Decision Sciences*, vol. 30, no. 2, pp. 481–501, 1999.
- [14] D. M.-H. Chiang, C.-P. Lin, and M.-C. Chen, "Data mining based storage assignment heuristics for travel distance reduction," *Expert Systems*, vol. 31, no. 1, pp. 81–90, 2014.
- [15] X. B. Xu and X. Li, "Research on route-based multi-staged storage location assignment," *Journal of Industrial Engineering and Management*, vol. 22, no. 5, pp. 24–31, 2017.
- [16] R. J. Mantel, P. C. Schuur, and S. S. Heragu, "Order oriented slotting: a new assignment strategy for warehouses," *European Journal of Industrial Engineering*, vol. 1, no. 3, pp. 301–316, 2010.
- [17] R. Alvarez-Valdes, J. M. F. Parreño, and J. M. Tamarit, "A GRASP/path relinking algorithm for two- and three-dimensional multiple bin-size bin packing problems," *Computers & Operations Research*, vol. 40, no. 12, pp. 3081–3090, 2013.
- [18] B. Pierre, B. Vannieuwenhuysse, D. Dominanta, and D. Van Dessel, "Dynamic ABC storage policy in erratic demand environments," *Jurnal Teknik Industri*, vol. 5, no. 1, pp. 1–12, 2003.