

Research Article

Storage Assignment Optimization in Robotic Mobile Fulfillment Systems

Ruiping Yuan ^{1,2}, Juntao Li ^{1,2}, Wei Wang ¹, Jiangtao Dou ¹, and Luke Pan ¹

¹School of Information, Beijing Wuzi University, Beijing 101149, China

²Beijing Key Laboratory of Intelligent Logistics Systems, Beijing 101149, China

Correspondence should be addressed to Juntao Li; ljtletter@126.com

Received 20 August 2021; Revised 16 October 2021; Accepted 3 November 2021; Published 25 November 2021

Academic Editor: Long Wang

Copyright © 2021 Ruiping Yuan et al. 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.

Robotic mobile fulfillment system (RMFS) is a new type of parts-to-picker order picking system, where robots carry inventory pods to stationary pickers. Because of the difference in working mode, traditional storage assignment methods are not suitable for this new kind of picking system. This paper studies the storage assignment optimization of RMFS, which is divided into products assignment stage and pods assignment stage. In the products assignment stage, a mathematical model maximizing the total correlation of products in the same pods is established to reduce the times of pod visits, and a scattered storage policy is adopted to reduce system congestion. A heuristic algorithm is designed to solve the model. In the pods assignment stage, a model is established minimizing the total picking distance of the mobile robots considering the turnover rate and the correlation of pods as well as the workload balance among picking corridors. A two-stage hybrid algorithm combining greedy algorithm and improved simulated annealing is designed to solve the model. Finally, a simulation experiment is carried out based on the historical order data of an e-commerce company. Results show that the storage assignment method proposed in the paper significantly improves the efficiency of order picking.

1. Introduction

E-commerce orders are generally large in quantity, small in batch, and unstable, so e-commerce order fulfillment can be quite challenging for warehouses. Robotic mobile fulfillment system (RMFS) is a new type of parts-to-picker order picking system, which was first brought to the market by Amazon in 2012. RMFS is particularly suited for e-commerce distribution centres that handle strong demand fluctuations and large assortments of small products, so it has become the development trend of e-commerce picking system in recent years [1]. In an RMFS, mobile robots are used to bring movable shelves, also known as inventory pods, to picking stations, where pickers take the required products off pods. After that, the robots bring the pods back to the storage area and continue to carry the next pods required. By eliminating the need for the pickers to walk and search the inventory, RMFS greatly improves the efficiency of order picking [2]. As a new picking mode, there are many optimization

problems to be studied in RMFS, such as storage assignment, order batching [3], multirobot task allocation [4], and path planning [5]. As an important optimization direction of RMFS, storage assignment has a direct impact on the total travel time/distance of robots, hence the efficiency of order picking.

In a traditional picker-to-parts picking system, the scientific storage assignment method can shorten the walking distance, reduce the search time, and improve the efficiency of order picking [6]. Hausman et al. [7] were the first to study the storage assignment strategy of traditional picking systems, and the subsequent literature conducted more in-depth research from demand correlation [8], COI (cube-per-order index) [9], turnover rate [10], and demand and structure correlation [11].

Unlike the traditional picker-to-parts picking system, storage assignment in RMFS includes not only products assignment (to decide which product to assign on which pod) but also pods assignment (to decide where the pods are

put in the storage areas) [12–14]. Because of the different ways of working, the traditional storage allocation algorithm cannot be directly applied to RMFS.

In recent years, the storage allocation problem in RMFS has attracted scholars' attention. Xi et al. [15] proposed a collaborative optimization model on the products assignment and order batching problem of RMFS considering the relevance of products. Yuan et al. [16] studied the pods assignment in RMFS, considering the random, class-based, and turnover-based assignment strategies. The research shows that class-based storage with two or three classes can achieve most of the potential benefits, and these benefits increase with greater variation in the pod velocities. Then Roy et al. [17] and Weidinger and Boysen [18] studied the pods assignment and products assignment in RMFS using random assignment policy and scattered assignment policy, respectively. Based on this, Weidinger et al. [19] further studied the reassignment of pods as a special interval scheduling problem and designed an adaptive algorithm. The results show that the proposed adaptive assignment rules are better than traditional assignment strategies. Lamballais et al. [20] developed an analytical model to optimize the inventory allocation across the pods. It is found that spreading inventory units across multiple pods is a better allocation strategy for e-commerce order fulfillment.

From the literature review, we can find that though the research on storage assignment under traditional manual picking mode is very rich, the research on that of the new picking system RMFS has just begun with the following shortcomings: (1) most of the research only focuses on the products allocation or pods allocation, but few on the joint optimization of the two stages; (2) the correlation between products is considered to reduce the pod visits in previous literature, but the correlation between pods is ignored. If we assign the pods with a strong correlation close to each other, it can effectively shorten the travel distance of robots during pods switching; and (3) the optimization objective of most existing literature is to minimize the travel distance of robots but seldom consider system congestion caused by the unbalanced workload among picking corridors, which affects system efficiency even more than travel distance of robots. Thus, reducing system congestion can effectively improve the picking efficiency of RMFS.

This paper studies the storage assignment optimization of RMFS and a joint optimization model of products assignment and pods assignment is built to improve the overall picking efficiency. The main innovations of this paper are as follows: (1) in the modelling, on the basis of the correlation and turnover rate of products, the correlation between pods is also considered to shorten the travel distance of robots during pods switching; (2) scattered storage policy and workload balance among picking corridors are adopted to reduce system congestion. Scattering products in different pods can increase the choice of pod visits and balancing the workload can reduce the congestion of some corridors; and (3) effective hybrid algorithms, including heuristic algorithm, greedy algorithm, and improved simulated annealing algorithm, are designed to solve the models.

The rest of the paper is organized as follows. In Section 2, we describe the storage assignment problem in RMFS. In Section 3, we build the mathematical models of products assignment and pods assignment. In Section 4, we design hybrid algorithms to solve the models. In Section 5, we verify the proposed methods using the real data of an e-commerce company. We finally conclude our study and discuss future research opportunities in Section 6.

2. Problem Description

The typical layout of an RMFS is shown in Figure 1, which is adopted by Amazon Kiva Systems [2] and most RMFS providers. The picking system consists of pods, picking stations, mobile robots, picking corridors, a conveyor belt, and so on. The storage area is composed of neatly distributed pods, each of which can store different products. In order to improve order picking efficiency, multiple orders arrived in a certain period of time are usually combined into one batch for picking, which is called wave picking. Orders in one wave are allocated to picking stations, and the items to be picked in the orders of each picking station are merged to generate a picking list, which consists of many picking tasks. These picking tasks are assigned to robots according to some rules, and these robots cooperate to complete the tasks.

The products picking process is as follows: A mobile robot is sent to the pod containing the required products and transports them to the picking station, where a picker picks the products from the pod, and then the robot sends the pod back to the storage areas (that is called one pod visit). After putting the previous pod back, the robot travels to carry the next pod (that is called pods switching) and so forth until all of its tasks are finished.

The optimization objective of RMFS is to minimize the total travel distance of robots or the total picking time of orders. The storage assignment in RMFS refers to assigning products to the appropriate pods (products assignment) and pods to the appropriate location in the storage areas (pods assignment), which obviously has a direct impact on the total travel time/distance of robots, hence the efficiency of order picking.

In most studies, storage assignment optimization is usually achieved by putting the related products on the same pods to reduce pod visits or by putting the pods with a high turnover rate closer to the picking stations to reduce the travel distance of robots. However, the picking corridors are generally very narrow in RMFS to save storage space. If the products with a high turnover rate are all placed in the zone close to the picking stations, it is easy to cause road congestion, and the robots have to queue and wait, which will affect the picking efficiency more than the saved travel distance. Therefore, system congestion factors should be given enough consideration in the storage allocation of RMFS.

In this paper, we study the joint optimization of products assignment and pods assignment of RMFS considering the correlation, turnover rate of products and pods, as well as system congestion factors. The main ideas of optimization in each stage are discussed below.

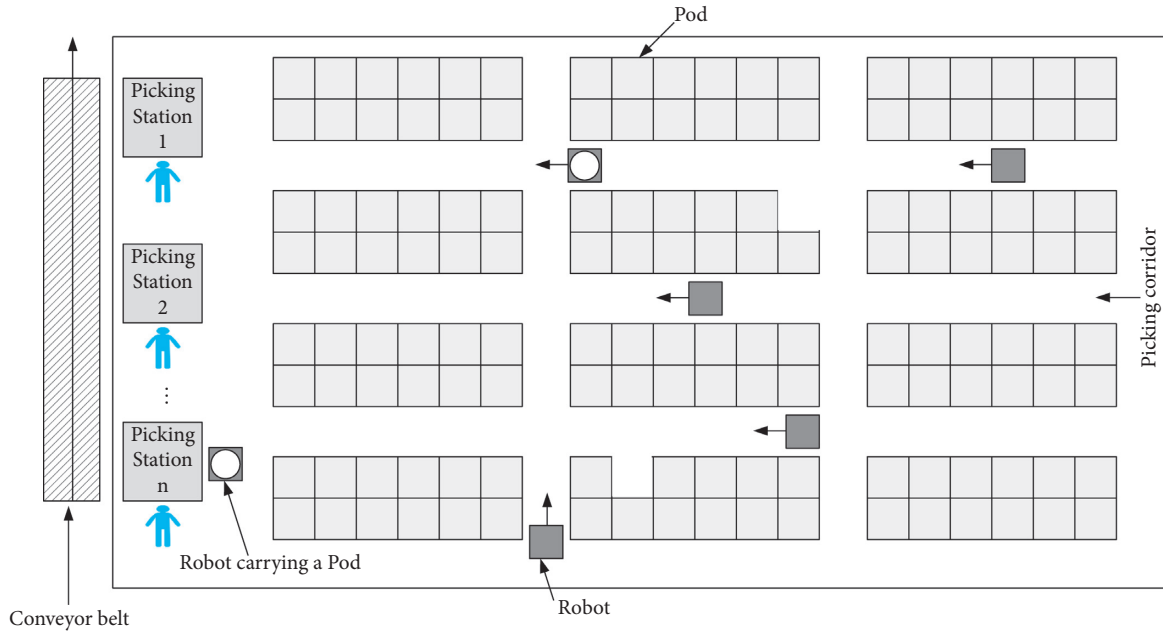


FIGURE 1: The typical layout of an RMFS.

2.1. The Products Assignment Stage. In the products assignment stage to decide which product to assign to which pod, we take the following measures to improve efficiency:

- (1) Store the products with strong correlation in the same pods to reduce the times of pod visits. The stronger the correlation of products in one pod, the more likely the products are required by the same order so that more kinds of products can be picked out by one pod visit, which obviously can reduce the total number of pod visits to fulfill the orders.
- (2) Spread the same kind of products across multiple pods other than just one pod to reduce the possibility of system congestion. When the products are scattered in different pods, it increases the choices of pod visits. When one pod is not available or it is in a congested area, there are still other pods can be chosen to fulfill the order. Thus, it can reduce system congestion and improve picking efficiency.

2.2. The Pods Assignment Stage. In the pods assignment stage, to decide where the pods to put in the storage areas, the following two aspects are considered:

- (1) In order to shorten the travel distance of robots during pods switching, the correlation between pods is considered. After putting the previous pod back, a robot has to travel to carry the next pod. If the next pod is put close to the previous pod, it will shorten the travel distance of pods switching and reduce the total travel distance of all robots.
- (2) While the pods with a high turnover rate are placed near picking stations, the workload balance among picking corridors is considered

simultaneously. The pods that contain the best-selling products have a high turnover rate and visit frequency, so putting them close to the picking stations can reduce the robots' travel distance. However, it can also cause congestion in hot areas. In order to reduce the possibility of road congestion, we balance the workload among corridors by setting a maximum workload volume for each corridor.

3. Model Formulation

3.1. Assumptions and Parameter definition. The following assumptions, which are reasonable in reality, are listed for the mathematical model formulation:

- (i) The pods are empty at the beginning.
- (ii) The inventory of each kind of product is 4 times its average demand. Yuan et al. [21] have proved that when the inventory is 4 times the average demand, there is little possibility of shortage.
- (iii) One kind of product can be stored in different pods.
- (iv) The number of pods is sufficient to store all products.
- (v) All pods have the same fixed layers (usually 6–8), and each layer can only store one kind of product in order to operate conveniently for the picker.
- (vi) The quantity of products stored on one pod can satisfy the demand of the products in one order.
- (vii) Only one pod can be allocated for each position in the storage area.
- (viii) A pod can only visit one picking station at one time and then is returned to its original position.

3.2. Mathematical Model of Products Assignment. This section will formulate an optimization model for products assignment with the objective of maximizing the total correlation of products in the same pods to reduce the number of pod visits. Meanwhile, scattered storage strategy is adopted to avoid system congestion. Before describing the model, the following parameters are given:

i, j represent product type, $i, j = 1, 2, \dots, P$, P is the total number of product types

n represents order, $n = 1, 2, \dots, N$, N is the total number of orders

m represents pod, $m = 1, 2, \dots, M$, M is the total number of pods

q represents the total number of layers on one pod

l represents the maximum number of items stored in each layer of a pod

a_i represents the average demand of product i

d_i represents the total number of storage layers needed of products i

$$d_i = \begin{cases} \frac{4a_i}{l}, & 4a_i \text{ can be divided by } l, \\ \left\lceil \frac{4a_i}{l} \right\rceil + 1, & \text{otherwise.} \end{cases} \quad (1)$$

$B_n = \{b_{1n}, b_{2n}, \dots, b_{Pn}\}^T$ represents products set included in order n . If product i is included in order n , $b_{in} = 1$; otherwise, $b_{in} = 0$.

R represents products correlation matrix:

$$R = \begin{bmatrix} r_{11} & r_{21} & \cdots & r_{P1} \\ r_{21} & r_{22} & \cdots & r_{P2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{P1} & r_{P2} & \cdots & r_{PP} \end{bmatrix}, \quad (2)$$

where r_{ij} is the correlation between products i and j . The more frequent the two kinds of products appear in the same order, the stronger their correlation is. The calculation method of the correlation between products is shown in equation (3). When $i \neq j$, the numerator is the number of orders both containing products i and j . The denominator is the number of orders either containing product i or j . When $i = j$, r_{ij} is the correlation of the same kind of product, in order to spread them across multiple pods, set $r_{ij} = 0$ in this case. r_{ij} can be obtained from historical orders data using the following equation:

$$r_{ij} = \begin{cases} \frac{\sum_{n=1}^N b_{in} b_{jn}}{\sum_{n=1}^N b_{in} + \sum_{n=1}^N b_{jn} - \sum_{n=1}^N b_{in} b_{jn}}, & i \neq j, \\ 0, & i = j. \end{cases} \quad (3)$$

x_{im} represents the decision variable and can be expressed as follows:

$$x_{im} = \begin{cases} 1, & \text{product } i \text{ is assigned to pod } m, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

y_{im} is an integer variable, which represents the number of storage layers product i occupying on pod m .

The objective is to maximize the total correlations between products on the same pods as follows:

$$\max z_1 = \sum_{i=1}^P \sum_{j>i}^P \sum_{m=1}^M \frac{r_{ij} x_{im} x_{jm}}{M}. \quad (5)$$

The constraints are presented as follows:

$$\sum_{i=1}^P y_{im} \leq q, \quad m = 1, 2, 3, \dots, M, \quad (6)$$

$$\sum_{m=1}^M y_{im} = d_i, \quad i = 1, 2, 3, \dots, P, \quad (7)$$

$$y_{im} \geq x_{im}, \quad i = 1, 2, 3, \dots, P, m = 1, 2, 3, \dots, M, \quad (8)$$

$$x_{im} = 0, 1, \quad i = 1, 2, 3, \dots, P, m = 1, 2, 3, \dots, M, \quad (9)$$

$$0 \leq y_{im} \leq q, \quad i = 1, 2, 3, \dots, P, m = 1, 2, 3, \dots, M. \quad (10)$$

Equation (6) stipulates that the number of storage layers assigned to the products is no more than the maximum layers of one pod. Equation (7) ensures that the total demand number of storage layers for each type of product is satisfied. Equations (8)–(10) are the basic constraints for the decision variables.

3.3. Mathematical Model of Pods Assignment. This section will formulate an optimization model for pods assignment with the objective of minimizing the total travel distance of all robots, considering the turnover rate of the pods, the correlation between the pods and the workload balance among picking corridors. Based on Section 3.2, some additional parameters are given.

t represents picking corridor, $t = 1, 2, \dots, T$, T is the total number of picking corridors.

Each corridor contains K storage positions. p_{tk} represents the k -th storage position of corridor t , where only one pod can be placed, $k = 1, 2, \dots, K$, $t = 1, 2, \dots, T$.

w_{tk} represents the shortest average distance between p_{tk} and the picking stations.

$d_{kk'}$ represents the distance between the current position k and the next position k' , where the next designated pod is located.

$H_m = \{h_{m1}, h_{m2}, \dots, h_{mP}\}$ represents the relationship vector between pod m and the products.

$$h_{mi} = \begin{cases} 1, & \text{if pod } m \text{ contains product } i, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

H is the pod-product relation matrix, $H = [H_1, H_2, \dots, H_M]^T$.

$C_m = \{c_{m1}, c_{m2}, \dots, c_{mN}\}$ represents the relationship vector between pod m and the orders.

$$c_{mn} = \begin{cases} 1, & \text{if pick order } n \text{ need carry pod } m, \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

C is the pod-order relation matrix,
 $C = [C_1, C_2, \dots, C_M]^T$.
 Decision variable:

$$P_{mtk} = \begin{cases} 1, & \text{if pod } m \text{ is allocated on position } k \text{ in corridor } t, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

The objective function is to minimize the total picking distance :

$$\min z_2 = \sum_{n=1}^N \sum_{k=1}^K \sum_{t=1}^T \sum_{m=1}^M c_{mn} P_{mtk} (2w_{tk} + d_{kk'}). \quad (14)$$

$2w_{tk}$ is the distance a robot travels to carry the pod located at p_{tk} to the picking station and return the pod back. $d_{kk'}$ is the distance a robot travels to fetch the next pod.

The constraints are presented as follows:

$$\sum_{t=1}^T \sum_{k=1}^K P_{mtk} = 1, \quad m = 1, 2, \dots, M, \quad (15)$$

$$\sum_{m=1}^M P_{mtk} \leq 1, \quad t = 1, 2, \dots, T; k = 1, 2, \dots, K, \quad (16)$$

$$\sum_{m=1}^M h_{mi} c_{mn} \geq b_{in}, \quad i = 1, 2, \dots, P; n = 1, 2, \dots, N, \quad (17)$$

$$\sum_{n=1}^N \sum_{m=1}^M \sum_{k=1}^K c_{mn} P_{mtk} \leq \left[\frac{\sum_{n=1}^N \sum_{m=1}^M c_{mn}}{Z} \right], \quad t = 1, 2, \dots, T; \\ Z = 1, 2, \dots, T, \quad (18)$$

$$c_{mn} = 0, 1, \quad n = 1, 2, \dots, N; m = 1, 2, \dots, M, \quad (19)$$

$$P_{mtk} = 0, 1, \quad m = 1, 2, \dots, M; t = 1, 2, \dots, \\ T; k = 1, 2, \dots, K. \quad (20)$$

Equation (15) stipulates that each pod must be placed at a position in the storage area. Equation (16) denotes that each storage position can only be placed by one pod at most. Equation (17) ensures that each kind of product that is contained in the order can be picked up from the pods that are carried to the picking station. Equation (18) represents the workload is balanced among picking corridors by setting a threshold for the total visits times of pods placed in the same corridor. Z can be any integer between 1 and T . When $Z = 1$, the constraint condition is always satisfied, that is, the threshold control does not work. When $Z = T$, the threshold constraint control is the strongest; equations (19) and (20) are the basic constraints of decision variables.

4. Algorithm Design

The integer programming models built in the previous section are nonlinear and have been proved to be NP-hard problems. A large amount of data is also needed to solve the models, so it is difficult to get the optimal solution. Therefore, according to the characteristics of the problem, this paper designs effective hybrid algorithms to solve the optimization problem quickly.

4.1. Algorithm Design for Products Assignment Model. A heuristic algorithm is designed to solve the products assignment model. The steps are as follows:

- (1) Step 1: Calculate the required storage layers of each product. According to the historical order data and equation (1), the average demand and the required storage layers of each product are obtained.
- (2) Step 2: Calculate the products information matrix RA . According to the historical order data and equation (3), the correlation between two products is calculated, and the products correlation matrix R is obtained. The products information matrix RA is formed by adding the required number of storage layers of each product to the right side of R .
- (3) Step 3: Find the maximum correlation value in the products information matrix RA and assign the two corresponding products to one pod. At the same time, reassign the correlation value to 0, update the required storage layers of the two products, and get the updated RA . In the updated RA , find the products most relevant to the above two products, assign the products to the same pod, and update RA . Repeat the process until all the layers of the pod are occupied and then update H . Notice that when the required storage layer of a product is reduced to 0, all the correlation values between the products and other products are reassigned to 0.
- (4) Step 4: Recover the correlation information of the products in RA whose required storage layers are not 0. Repeat step 3 for other pods until the required layers of all products are 0.

The pseudo code of the heuristic Algorithm 1 is as follows:

4.2. Algorithmic Design for Pods Assignment Model. The pods assignment problem is similar to the knapsack problem, which can be solved by a simulated annealing algorithm. In order to adapt to the nature of the problem and improve the search efficiency, the simulated annealing algorithm is improved from three aspects: (1) a greedy algorithm based on pods correlation is used to generate the initial solution; (2) the principle of pods with high turnover rate placed near picking stations is used to generate the new solution; and (3) at the same time, the workload among corridors is balanced to reduce system congestion.

```

Input: Historical order data
Output: The pod-product relation matrix  $H$ 
(1) #Calculate products correlation matrix  $R$ 
(2) for ( $i = 1; i \leq P; i++$ ) do
(3)   for ( $j = 1; j \leq P; j++$ ) do
(4)      $r(i, j) \leftarrow$  Calculate the correlation between products  $i$  and  $j$ 
(5)   end
(6) end
(7) #Calculate products information matrix  $RA$ 
(8) for ( $i = 1; i \leq P; i++$ ) do
(9)    $RA(i, P+1) \leftarrow$  Calculate the required storage layers for product  $i$ 
(10) end
(11) #Calculate which product to put on which pod
(12) for ( $m = 1; m \leq M; m++$ ) do
(13)    $r \leftarrow 1$ 
(14)   product  $i, j \leftarrow$  find ( $A = \max(\max(RA))$ ), then update  $RA$ 
(15)    $r \leftarrow 3$ 
(16)   while  $r \leq q$  do
(17)     Find the most relevant product  $u$  in  $RA$  and update  $RA$ 
(18)     update  $H_m$ 
(19)     if  $RA(u, P+1) = 0$  then
(20)        $RA(u, :) \leftarrow 0$ 
(21)        $RA(:, u) \leftarrow 0$ 
(22)     end
(23)     if sum( $RA(:, P+1)$ ) = 0 then
(24)       end calculation
(25)     end
(26)      $r \leftarrow r + 1$ 
(27)   end
(28) end

```

ALGORITHM 1: The heuristic algorithm of products assignment.

4.2.1. Data Preparation. Firstly, we prepare the data needed. According to the pod-product relation matrix H obtained in products assignment stage, the pod-order relation matrix C is constructed based on the principle of maximum set coverage. Then the pods correlation matrix E is constructed, and the pod turnover rate is calculated.

(1) *Pod-Order Relation Matrix C .* The maximum set coverage strategy is adopted to decide which pod to carry, that is, when one order needs to be picked, the robot carries the pods containing the most types of products in the order to fulfill it. c_{mn} is set to 0 at the beginning. For each order, repeat the following steps to update the pod-order relation matrix C .

Step 1: For the current order n , retrieve the types of unpicked products in the order.

Step 2: Search for the pod that contains the most types of the above products and choose the pod to serve the order.

Step 3: Check whether all the products in order n have been picked, if so, update $c_{1n}, c_{2n}, \dots, c_{Mn}$ in the pod-order relation matrix C and change the corresponding data from 0 to 1. For example, if pod m is selected to serve order n , c_{mn} is updated to 1; otherwise, return to step 1.

(2) *Pods Correlation Matrix E .* The correlation between two pods relates with the proportion of the common orders they can fulfill, which is measured by cosine correlation as shown in equation (21). C_m is the relationship vector between pod m and the orders, and C_g is the relationship vector between pod g and the orders. $C_m \cdot C_g$ represents the number of the common orders pod m and g can fulfill, respectively, and $|C_m|$ and $|C_g|$ is the number of the orders pod m and g can fulfill, respectively.

$$e_{mg} = \cos(C_m, C_g) = \frac{C_m \cdot C_g}{|C_m| \cdot |C_g|}. \quad (21)$$

(3) *Pod Turnover Rate Fre_m .* The pod turnover rate Fre_m is calculated using the historical order information and pod-order relation matrix C .

$$Fre_m = \frac{\sum_{n=1}^N c_{mn}}{\sum_{n=1}^N \sum_{m=1}^M c_{mn}}. \quad (22)$$

4.2.2. Two-Stage Hybrid Algorithm. A two-stage hybrid algorithm combining greedy algorithm and simulated annealing is designed to solve the pods assignment model.

Stage I: according to the pods correlation matrix, a greedy algorithm is designed to generate the initial solution of pods assignment.

Stage II: the simulated annealing algorithm is improved to optimize the solution and get the final pods assignment results.

The steps of each stage are as follows:

(1) *Stage I.* Generate initial solution: The purpose of this stage is to generate the initial solution of the subsequent simulated annealing algorithm considering the correlation of pods. The specific steps of the algorithm are as follows:

Step 1: According to the pods correlation matrix E , select a pair of pods with the maximum correlation.

Step 2: Select a pair of positions with the minimum distance in the position distance matrix D . If there are more than one pair of positions with minimum distance, choose the pair closer to the picking stations.

Step 3: Check whether the total pod turnover rate in the corridor will exceed the threshold value after the two pods in step 1 are placed in the two positions in step 2. If not, put the pods there. Then update the pods correlation matrix and distance matrix by reassigning the values corresponding to the selected pods and positions to -1 . Otherwise, select the next position pair with a smaller distance.

Step 4: Repeat steps 1 to 3 until each pod is assigned to a storage position and the initial solution is obtained.

(2) *Stage II.* Optimize the initial solution: In this stage, the simulated annealing algorithm is improved to optimize the initial solution considering pod turnover rate and the balance among corridors.

Step 1. Initialization: Generate initial solutions $P = P_0$ from the previous stage and set the initial temperature T_0 , which is big enough. Set $T = T_0$ and determine the number of iterations, which is metropolis chain length L .

Step 2: For current temperature T , switch the positions of pods to get new solution P' according to pod turnover rate and workload balance among corridors. Select some pairs of pods with a higher turnover rate and switch their positions with the pods with a lower turnover rate but closer to the picking stations. The purpose is to move the pods with a high turnover rate closer to the picking stations to reduce travel distance. Then check whether the workload balance constraint (equation (18)) is still satisfied after the pods movement to avoid congestion. If it is satisfied, a new solution P' is generated; otherwise, select other pods to switch positions to get a new solution P' .

Step 3: Compute the increment of P' : $\Delta f = f(P') - f(P)$. $f(P)$ is the cost function of P . If $\Delta f < 0$, accept P' as a new current solution, set $P = P'$, $l = l + 1$. If $\Delta f \geq 0$, calculate acceptance

probability $\exp(-\Delta f/T)$ of P' . That is, generate a random number rand with uniform distribution in $(0,1)$ intervals; if $\exp(-\Delta f/T) > \text{rand}$, P' is accepted as a new current solution and set $P = P'$ and $l = l + 1$. Check if l reach the maximum number of iteration, if so, go to step 4; otherwise, return to step 2.

Step 4: According to the attenuation function, set $T = T * \Delta T$ (ΔT is generally $0.95 \sim 0.99$) to reduce temperature. Check if the minimum temperature T_{\min} is reached, if so, output the current solution P as the best solution, and end the program; otherwise, return to step 2.

5. Simulation Analysis

The storage assignment models and algorithms proposed in this paper are verified by the data of an e-commerce company. The company sells 250 kinds of products online and uses RMFS for order picking. The products are stored on mobile pods, each of which is divided into 8 layers and the maximum capacity of each layer is 70 units. There are 10 mobile robots and 6 picking workstations in the warehouse. Suppose the travel speed of robots is 1 m/s and the picking time of one item is 5 s. We compare our proposed method with other methods in products assignment, pods assignment, and the overall picking efficiency.

5.1. Simulation Results of Products Assignment. The simulation program of the storage assignment method proposed in this paper is constructed using MATLAB. The optimal result for product assignment is obtained using the data of 2,000 history orders of the e-commerce company. The result for the products assignment is shown in Table 1.

We can see from Table 1 that one kind of product is assigned to different pods; for example, product 131 is allocated to pods 8 and 9, that is, when product 131 needs to be picked, there are two choices of pod visits, which can reduce system congestion and improve the efficiency of order picking.

The heuristic algorithm proposed in the paper is compared with the random assignment method and the Apriori algorithm. Random assignment is to store the products on the pods randomly without considering the correlation of products. Apriori algorithm was first proposed by Agrawal and Shafer [22] and has been widely used in storage assignments [23]. The basic idea is to count the frequency of multiple products in the same order and put the products with high frequency on the same pod. The methods were compared under different order size ($n = 500, 1,000, 1,500,$ and $2,000$) in the number of pod visits. The results are shown in Figure 2.

It can be found from Figure 2 that the times of pod visits of the heuristic algorithm are the smallest among the three methods under different order sizes, 32.7%~36.6% less than random algorithm and 16.9%~18.6% less than the Apriori algorithm. Besides, with the increase of order size, the growth rate of pod visits in the proposed heuristic algorithm slows down, which means that the bigger the order size is, the greater the possibility of the correlation between

TABLE 1: Products assignment result.

Pod number	Product number							
1	26	203	100	103	137	96	247	238
2	175	41	165	250	33	50	185	25
3	8	208	49	14	189	112	19	153
4	142	62	159	58	128	36	60	237
5	121	139	155	111	148	221	141	100
6	144	236	195	107	232	135	197	45
7	43	82	93	102	66	50	232	219
8	63	35	7	187	5	45	193	131
9	213	150	95	125	79	149	131	16
...								
164	202	180	187	34	1	144	99	250
165	161	190	55	208	118	7	—	—

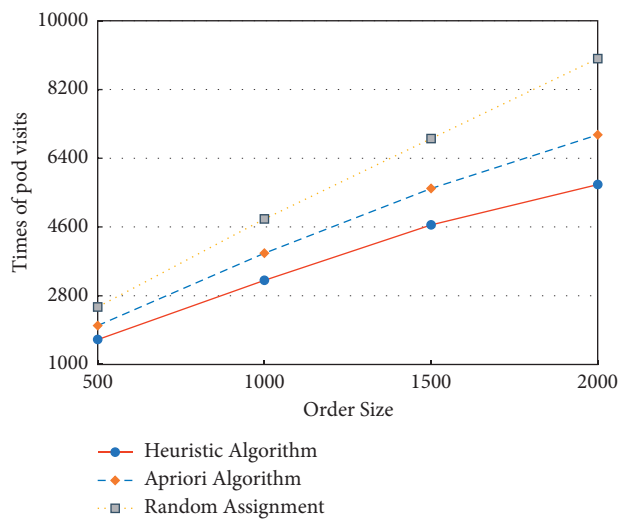


FIGURE 2: Times of pod visits using different algorithms.

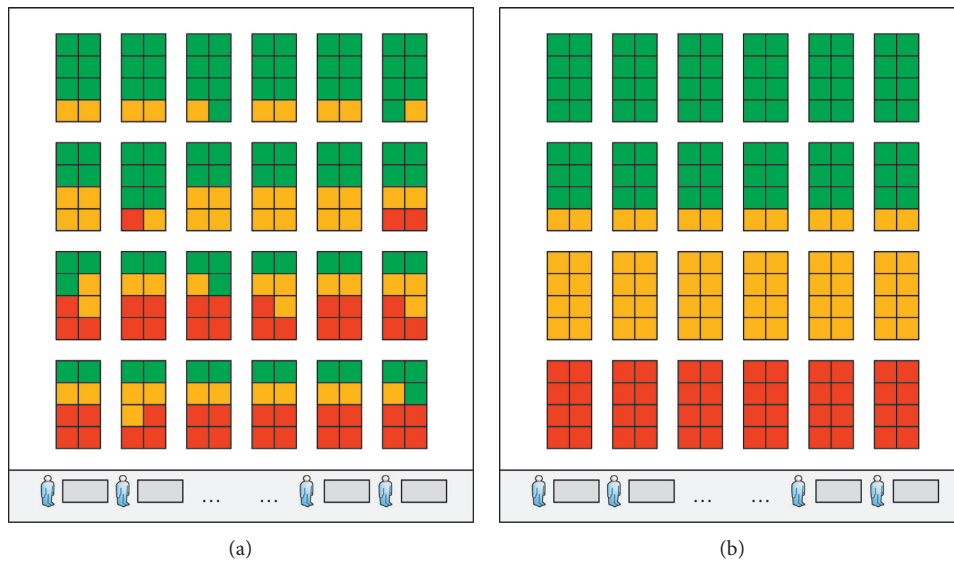


FIGURE 3: Comparison of workload balance using different pods assignment methods: (a) pods assignment using the proposed method and (b) pods assignment using ABC classification method.

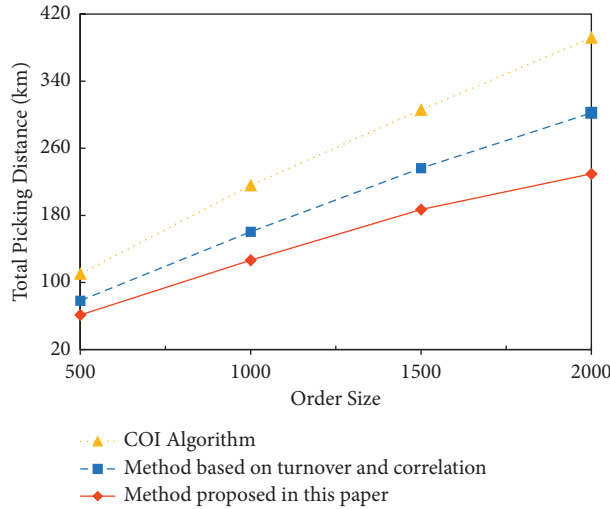


FIGURE 4: Total piking distance of different methods.

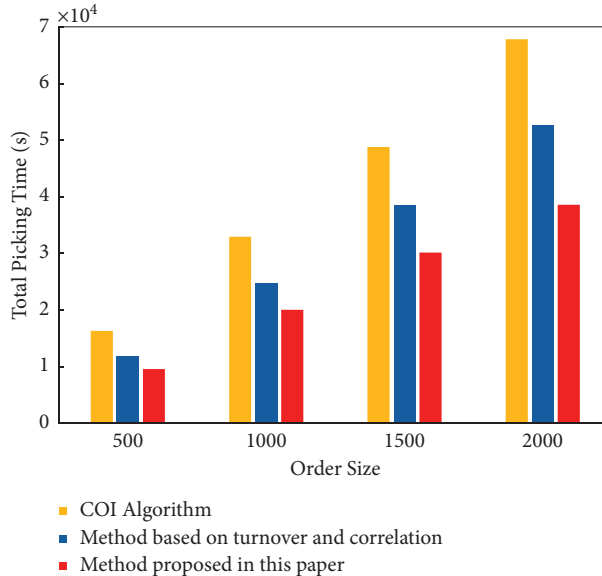


FIGURE 5: Total picking time of different methods.

products, so the advantage of the heuristic algorithm is more obvious. The result shows that maximizing the correlations of the products in one pod can effectively reduce the times of pod visits and improve the picking efficiency.

5.2. Simulation Results of Pods Assignment. In order to alleviate system congestion, the method of balancing the workload among corridors is adopted, and the pod assignment result is shown in Figure 3(a). The typical pod layout using the traditional ABC classification method [11] is shown in Figure 3(b). In the figures, the red ones represent pods with a high turnover rate, accounting for about 25% of the total number of pods; the yellow ones represent pods with a medium turnover rate, accounting for about 30% of the total number of pods; and the green ones represent pods with a low turnover rate, accounting for about 45% of the total number of pods.

We can see from Figure 3(b) that all the pods with the high turnover rate (the red ones) are assigned to the storage positions in the first row, which is closest to the picking stations, and most pods with the medium turnover rate (the yellow ones) are assigned to the storage positions in the second row. This will lead to a high total access rate of pods in this area, resulting in congestion of picking robots in the corridors nearby. In Figure 3(a), the red and yellow pods are properly scattered in different areas. We can see that the red pods account for only about 48% of the total positions in the first row. In this way, the workload of corridors can be effectively balanced, and system congestion can be alleviated.

5.3. Comparative Analysis of the Overall Picking Efficiency. In order to validate the efficiency of the storage assignment models and algorithm proposed in this paper, it is compared with other storage assignment methods commonly used

(COI algorithm in reference [24] and the algorithm just based on turnover and product correlation in reference [21]) from total picking distance and picking time.

The COI algorithm is also called the cube-per-order index algorithm, which considers the volume and the turnover rate of products [24]. The method used in reference [21] is mainly considered the correlation and turnover rate of products but does not consider the correlation between pods. The method proposed in this paper considers the correlation of products, the turnover rate, and the correlation of pods, as well as the workload balance of corridors. The experimental results of the total picking distance and picking time using the three methods are shown in Figures 4 and 5, respectively.

Figure 4 shows that the total picking distance using the method proposed in this paper is much shorter than that of the COI algorithm and the method, only considering the correlation and the turnover rate of products. And with the increase of order size, the growth rate of picking distance in our proposed method slows down, while the growth rate of other methods shows a linear growth.

Figure 5 shows that the total picking time using the storage assignment method proposed in this paper is much lower than that of the COI algorithm and methods based on correlation and turnover rate of products. And with the increase of order size, the gap of the picking time among methods increases.

Simulation results above show that the storage assignment method proposed in this paper can significantly improve the order picking efficiency by alleviating system congestion and reducing picking distance. Besides, with the increase of order size, the improvement of this method is more obvious.

6. Conclusions and Prospects

This paper studies the joint optimization of products assignment and pods assignment in RMFS. A two-stage mathematical model is established considering the correlation, turnover rate of products and pods, as well as system congestion factors. Hybrid algorithms, including heuristic algorithm, greedy algorithm, and improved simulated annealing algorithm, are designed to solve the models. Simulation experiments are carried out based on the historical data of an e-commerce company. The method proposed in this paper is verified by comparing with other commonly used storage assignment methods from times of pod visits, system congestion situation, total picking distance, and time. Experimental results show that the storage assignment method proposed in the paper significantly improves the efficiency of order picking.

The paper mainly solves the static storage assignment problem considering the characteristics of the operation process of RMFS. However, due to the complexity and dynamics of system scheduling, there are still some problems that need to be further studied. For example, in practical application, the characteristics of orders will change over time, so it is necessary to adjust the storage allocation dynamically according to the changes. Thus, the dynamic

storage assignment optimization can further improve the picking efficiency and is a very promising research direction.

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 there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The paper was funded by National Natural Science Foundation of China (72101033 and 71831001), Beijing Key Laboratory of Intelligent Logistics Systems (BZ0211), Canal Plan-Youth Top-notch Talent Project of Beijing Tongzhou District (YHQN2017014), Ningxia Science and Technology Key Research and Development Project (2018BEG03003), and Scientific Research Program of Beijing Municipal Commission of Education (KM202110037003).

References

- [1] T. Lamballais, D. Roy, and M. B. M. De Koster, "Estimating performance in a robotic mobile fulfillment system," *European Journal of Operational Research*, vol. 256, no. 3, pp. 976–990, 2017.
- [2] N. Boysen, R. De Koster, and F. Weidinger, "Warehousing in the E-commerce era: a survey," *European Journal of Operational Research*, vol. 277, no. 2, pp. 396–411, 2019.
- [3] X. M. Luo, X. Y. Xia, and J. B. Li, "Study on the batch of warehouse orders of FMCG E-commerce," *Journal of Systems Science and Mathematical Sciences*, vol. 36, no. 6, pp. 847–859, 2016.
- [4] R. P. Yuan, H. L. Wang, L. R. Sun, and J. T. Li, "Research on the task scheduling of part-to-picker order picking system based on logistics AGV," *Operations Research and Management Science*, vol. 27, no. 10, pp. 133–138, 2018.
- [5] Z. Zheng, Q. Guo, J. Chen, and P. J. Yuan, "Collision-free route planning for multiple AGVs in automated warehouse based on collision classification," *IEEE Access*, vol. 6, pp. 2602–2603, 2018.
- [6] 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.
- [7] W. H. Hausman, L. B. Schwarz, and S. C. Graves, "Optimal storage assignment in automatic warehousing systems," *Management Science*, vol. 22, no. 6, pp. 629–638, 1976.
- [8] 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.
- [9] F. Caron, G. Marchet, and A. Perego, "Routing policies and COI-based storage policies in picker-to-part systems," *International Journal of Production Research*, vol. 36, no. 3, pp. 713–732, 1998.
- [10] J. B. Li, G. Y. Yang, and F. Chen, "Retail warehouse center storage location assignment research for E-commerce,"

- Industrial Engineering and Management*, vol. 18, no. 4, pp. 102–108, 2013.
- [11] J. Li, M. Moghaddam, and S. Y. Nof, “Dynamic storage assignment with product affinity and ABC classification—a case study,” *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 9, pp. 1–16, 2016.
 - [12] X. B. Xu and Z. Q. Ma, “Robotic mobile fulfillment systems: state-of-the-art and prospects,” *Acta Automatica Sinica*, vol. 46, no. 9, pp. 1–25, 2020.
 - [13] N. Boysen, D. Briskorn, and S. Emde, “Parts-to-picker based order processing in a rack-moving mobile robots environment,” *European Journal of Operational Research*, vol. 262, no. 2, pp. 550–562, 2017.
 - [14] R. Yuan, H. Wang, and J. Li, “The pod assignment model and algorithm in robotic mobile fulfillment systems,” in *Proceedings of the 2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pp. 99–103, Zhengzhou, China, November 2019.
 - [15] X. Xi, C. Liu, and L. Miao, “Storage assignment and order batching problem in Kiva mobile fulfillment system,” *Engineering Optimization*, vol. 50, no. 11, pp. 1941–1962, 2018.
 - [16] R. Yuan, S. C. Graves, and T. Cezik, “Velocity-based storage assignment in semi-automated storage systems,” *Production and Operations Management*, vol. 28, no. 2, pp. 354–373, 2019.
 - [17] D. Roy, S. Nigam, R. De Koster, I. Adan, and J. Resing, “Robot-storage zone assignment strategies in mobile fulfillment systems,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 122, pp. 119–142, 2019.
 - [18] F. Weidinger and N. Boysen, “Scattered storage: how to distribute stock keeping units all around a mixed-shelves warehouse,” *Transportation Science*, vol. 52, no. 6, pp. 1412–1427, 2018.
 - [19] F. Weidinger, N. Boysen, and D. Briskorn, “Storage assignment with rack-moving mobile robots in KIVA warehouses,” *Transportation Science*, vol. 52, no. 6, pp. 1479–1495, 2018.
 - [20] T. Lamballais, D. Roy, and R. B. M. De Koster, “Inventory allocation in robotic mobile fulfillment systems,” *IIEE Transactions*, vol. 52, no. 1, pp. 1–17, 2020.
 - [21] R. Yuan, T. Cezik, and S. C. Graves, “Stowage decisions in multi-zone storage systems,” *International Journal of Production Research*, vol. 56, no. 1-2, pp. 333–343, 2018.
 - [22] R. Agrawal and J. C. Shafer, “Parallel mining of association rules,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 962–969, 1996.
 - [23] D. Ming-Huang 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.
 - [24] M. Li and Y. Zhang, “A study of workload distribution and COI-based storage policies,” *Industrial Engineering*, vol. 18, no. 1, pp. 37–41, 2015.