Hindawi
Complexity
Volume 2024, Article ID 9260431, 15 pages
https://doi.org/10.1155/2024/9260431

Research Article

Joint Optimization of Item and Pod Storage Assignment Problems with Picking Aisles' Workload Balance in Robotic Mobile Fulfillment Systems

Jun Zhang,[1,2] **Lingkun Tian**,[1] **and Zijuan Zhou**[1]

[1]*School of Information Management, Central China Normal University, Wuhan, China*
[2]*E-Commerce Research Center of Hubei Province, Central China Normal University, Wuhan, China*

Correspondence should be addressed to Jun Zhang; zhangj@ccnu.edu.cn

Received 23 August 2023; Revised 4 December 2023; Accepted 18 December 2023; Published 5 January 2024

Academic Editor: Dan Selişteanu

The item and pod storage assignment problems, two critical issues at the strategic level in robotic mobile fulfillment systems, have a strong correlation and should be studied together. Moreover, the workload balance in each picking aisle needs to be considered in the storage assignment problems to avoid robots' congestion within picking aisles. Motivated by these, the joint optimization of item and pod storage assignment problems (J-IPSAP) with picking aisles' workload balance is studied. The mixed integer programming model of the J-IPSAP with the workload balance constraint is formulated to minimize the robots' movement distance. The improved genetic algorithm (IGA) with the decentralized pod storage assignment strategy is designed to solve the J-IPSAP model. The experimental results show that the IGA can obtain high-quality solutions when compared with Gurobi and the two-stage heuristic algorithms. The robots' movement distance is smallest when the width-to-length ratio of the storage area is close to 1, and the robots' movement distance will increase with more stringent workload balance constraints.

1. Introduction

In recent years, intelligent order picking systems have emerged and greatly increased picking efficiency. The robotic mobile fulfillment system (RMFS), one of the intelligent order picking systems, has been successfully applied to many companies, such as Swisslog, GreyOrange, JD Logistics, and Cainiao Logistics [1, 2]. As shown in Figure 1, RMFS mainly consists of pickers, robots, picking stations, pods, and items stored on the pods [3]. The main picking process of RMFS includes the following steps: (1) the robot moves to the location of the pod containing the specified items; (2) the robot carries the pod to the pod buffer area of the designated picking station; (3) the pod queues at the station until its turn; (4) the picker stands at the picking station and waits to pick items from the pod and put items into the specified tote areas; and (5) the robot carries the pod to the warehouse.

Before the picking process, decisions are first made about where the item should be placed on the pod and where the pod should be allocated in the warehouse (storage assignment problem). After that, when orders come in, decisions are made about which picking station should the orders be assigned to (order assignment problem) and which pods should be selected to fulfill the orders (pod selection problem). The robots are then assigned to deliver the pods to the picking stations (robot task assignment and path planning problems) [4], and the pods are returned to the storage area after the picking is completed (pod reassignment problem). The replenishment operation is needed with the inventory level of items below the safety stock level. As we know, the storage assignment problem belongs to the strategic level in the RMFS decision framework [5, 6]. Other problems, such as order assignment, pod selection, robot task assignment and path planning, pod reassignment, and replenishment operation, belong to the operational level in the decision framework [5, 6].

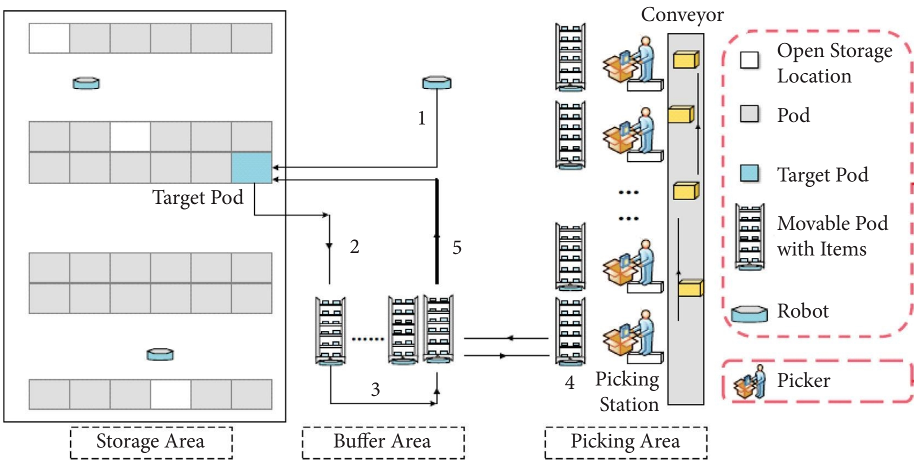


Figure 1: Picking process of the robotic mobile fulfillment system.

In this paper, we consider the storage assignment problem (SAP), which is the key decision issue at the strategic level [5]. SAP can be divided into the item storage assignment problem (ISAP) and the pod storage assignment problem (PSAP). The ISAP determines which pods the items should be allocated to [7]. The PSAP determines which locations in the warehouse the pods need to be allocated to. There is a strong correlation between ISAP and PSAP, and ISAP and PSAP should be studied together. In the current study, the ISAP and PSAP are solved following the idea of sequential decision-making, i.e., the PSAP is solved after the solutions of the ISAP are obtained [8, 9]. However, sequential decision-making cannot get a global optimization solution of SAP. Thus, it is necessary to further study the joint optimization of ISAP and PSAP to reduce the system operation cost. Moreover, the SAP solution with no consideration of the workload balance in each picking aisle will increase the robots' congestion within the picking aisles and decrease the picking efficiency [8–10]. Motivated by this, this article studies the joint optimization of item and pod storage allocation problems (J-IPSAP) with picking aisles' workload balance.

The J-IPSAP determines the items' allocation on the pods and the pods' allocation in the warehouse simultaneously. The picking aisles' workload balance is also considered to avoid congestion. The mathematical model of the J-IPSAP is formulated to reduce the robot's movement distance. The constraint of workload balance within the picking aisles is added to the model. To solve the J-IPSAP model, the improved genetic algorithm (IGA) with the elite preservation mechanism is designed. To verify the effectiveness of the J-IPSAP model and IGA, the computational experiments are conducted by comparing with the Gurobi optimization solver and the related literature [9] under different warehouse scales. The sensitivity analysis is also performed to further explore the effect of the layout of the storage area and the workload balance constraint on the picking efficiency. Although the joint optimization would increase the complexity of the model, the storage assignment decision has a relatively low requirement on CPU times, and the assignment operation is usually performed during nonworking hours. The improved genetic algorithm provided is very efficient in solving the joint optimization in a limited time, and the robots' movement distance of the joint optimization model is about 30% less than the sequent optimization models.

This article has the following main contributions: (1) the previous SAP only considered sequential decision-making, i.e., the PSAP is solved after the solutions of the ISAP are obtained. This article extends the SAP study by jointly optimizing the ISAP and PSAP to increase the RMFS's picking efficiency; (2) the picking aisles' workload balance is considered to avoid aisle congestion. The new decentralized pod storage assignment strategy is proposed to solve the picking aisles' workload balance problem; and (3) the IGA is designed to solve the J-IPSAP model, and the IGA is proven to outperform other algorithms in the related literature.

The structure of the rest of the study is as follows. In Section 2, the relevant literature is described. In Section 3, the J-IPSAP optimization model is formulated. In Section 4, the IGA is designed to solve the J-IPSAP model. In Section 5, the computational experiments are presented. In Section 6, the conclusions and future works are given.

2. Literature Review

Merschformann et al. [5] and Xie et al. [6] classified the RMFS decision problems into three levels: strategic level, tactical level, and operational level. SAP belongs to the strategic level. Other decision problems, such as order assignment [11–13], robot scheduling [14–16], pod repositioning, and path planning [17–19], belong to the operational level. This section aims to provide a comprehensive overview of previous work related to SAP in RMFS. More optimization studies on other decision levels can be seen in the recent literature reviews [20–22].

2.1. The Study of ISAP in RMFS.

The ISAP determines which pods the items should be allocated to [7]. In the research on ISAP, Xiang et al. [7] used historical orders to establish an objective function to maximize the relevance of items on the pods to solve ISAP, reducing the number of pod visits effectively. Kim et al. [23] addressed the ISAP to maximize the relevance of items on the pods, and the movement distance of the pods is reduced by the re-optimization algorithm. Guan and Li [24] studied the problem of decentralized storage allocation of items based on association rules in RMFS, aiming to determine the pods on which each item is placed to minimize the number of pods that need to be moved when picking a batch of orders. They proposed an integer programming model and developed a genetic algorithm to solve the problem. Mirzaei et al. [25] proposed an integrated cluster allocation (ICA) policy to minimize the retrieval time of parts-to-picker systems based on both product turnover and affinity. They formulated a mathematical model that can solve small instances and developed a greedy construction heuristic for solving large instances. Zhang et al. [3] addressed the ISAP by incorporating the energy consumption of pickers. By establishing a multiobjective mixed integer programming model, the objective is to maximize the items' similarities in the pods and minimize the pickers' energy expenditure.

2.2. The Study of PSAP in RMFS. The PSAP exists at both the strategic and operational levels in RMFS. At the strategic level, the PSAP is to determine which locations in the storage area the pods should be assigned to. At the operational level, the PSAP is called the pod storage reposition problem (PSRP), which is to determine which locations the pods should return to after visiting the picking stations.

This article focuses on the PSAP at the strategic level. As the PSAP is usually solved with the ISAP, fewer scholars studied the PSAP alone. Wang et al. [10] studied the PSAP in the single-deep layout of fishbone robotic mobile fulfillment systems (FRMFS). They proposed the pod allocation optimization model to maximize the picking efficiency and balance the workload of each picking aisle. Keung et al. [26] studied the zone clustering and data-driven order classification methods to solve the PSAP in both single-deep and multideep layouts.

In some studies of PSRP, it is assumed that the pods will return to their original storage locations [9, 10]. Other scholars studied the pods' optimal return locations in the PSRP. Weidinger et al. [27] transformed the PSRP into a special interval scheduling problem. After that, they constructed the optimization model and designed the adaptive large metaheuristic search. Merschformann et al. [5] proposed different strategies for the PSRP, such as random storage and nearest storage. Ji et al. [28] used the three-class-based method and the Kuhn–Munkres algorithm to solve the PSRP. Cai et al. [29] proposed a pod repositioning strategy of item clustering and pod turnover to reduce the robots' total movement time. Zhuang et al. [16] solved the PSRP and the pods' task allocation problem simultaneously to minimize the maximum completion time or the robots' movement distance.

2.3. The Study of ISAP and PSAP in RMFS. To the authors' best knowledge, only two articles studied the ISAP and PSAP in RMFS simultaneously, and the two-stage optimization approach is used to solve the ISAP and PSAP [9, 30].

Li et al. [30] solved the ISAP by clustering the items into several types through an association rule algorithm. The model was constructed to reduce the distance between two item types. After that, they proposed the PSAP mathematical model to maximize the distance between two high-turnover pods. The objective function is to avoid the congestion of robots. Yuan et al. [9] established the ISAP mathematical model that maximizes items' relevance on pods. The model was solved by designing a heuristic algorithm. After that, they developed the PSAP model, intending to reduce the robots' movement distance by considering pod association, pod turnover, and workload between picking aisles.

2.4. Summary of Research Status. Table 1 summarizes the literature related to SAP at the strategic level of RMFS. It can be seen that the objective functions include the items' relevances and the robots' movement distances. Zhang et al. [3] also considered minimizing pickers' energy expenditure from the human aspect. Li et al. [30] and Yuan et al. [9] solved the ISAP and PSAP by using a two-stage optimization approach. Wang et al. [10], Li et al. [30], and Yuan et al. [9] also considered picking aisles' workload balance in the PSAP mathematical model.

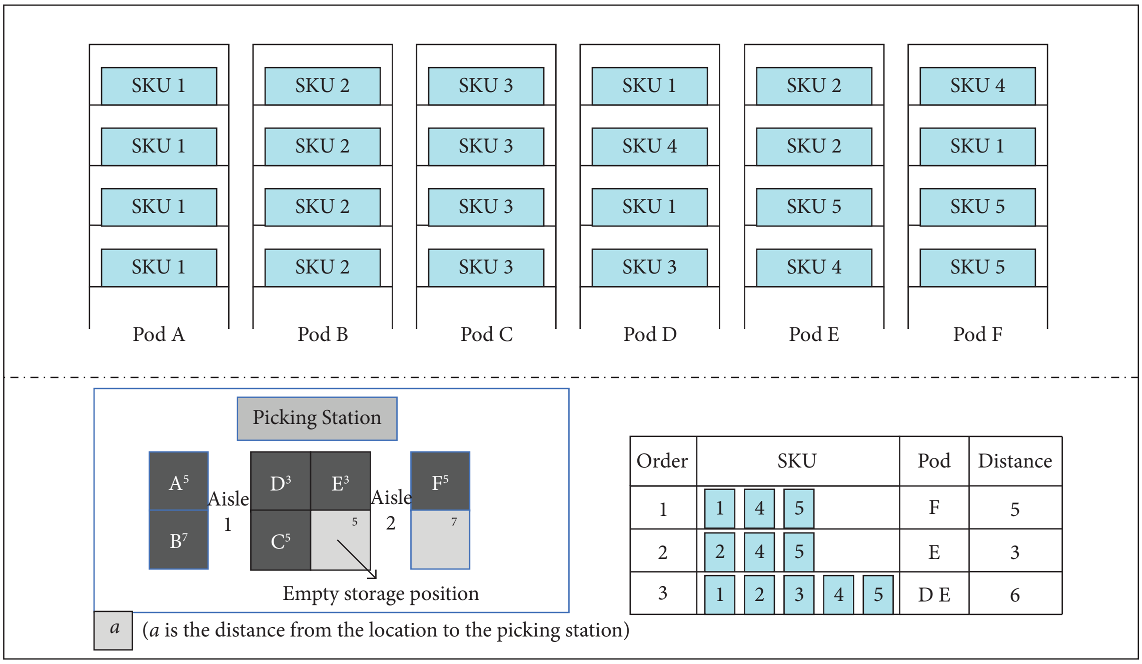
From the literature review presented above, it can be inferred that the current literature does not consider the joint optimization approach to solve the ISAP and PSAP simultaneously. Moreover, the picking aisles' workload balance is the key issue that needs to be considered in the PSAP. Therefore, this article studies the joint optimization of ISAP and PSAP (J-IPSAP) with picking aisles' workload balance. The optimization model is formulated to reduce the robots' movement distance. The constraint of the picking aisles' workload balance is added to the model. The IGA is designed to solve the J-IPSAP model. The item storage allocation strategy, pod selection strategy, and pod-decentralized storage strategy are designed in the algorithm.

3. Mathematical Model

3.1. Problem Description. Assume that there are $|T|$ picking aisles and $|L|$ locations in the RMFS. To satisfy customer orders, it is proposed to assign $|K|$ items on $|R|$ pods and to allocate $|R|$ pods to $|L|$ locations in the storage area. Since the demand for items in customer orders will change over time, the storage locations of the items and pods are also typically updated periodically to accommodate order picking tasks in further periods. This article assumes that the customer orders in further periods can be predicted and the number of storage locations of each item is known. Based on the information of the customer orders, the joint optimization of ISAP and PSAP (J-IPSAP) needs to decide: (1) which pods the items should allocate to and (2) which storage locations the pods should allocate to. The J-IPSAP optimization goal is to reduce the robot's movement distance. Meanwhile, to balance the picking aisles' workload, this article needs to obtain pod selection solutions for customer orders (see details in Section 4.3).

Table 1: Summary of the literature on SAP at the strategic level of RMFS.

Research problem	Objective function	Picking aisles' workload balance	Literature
ISAP	Maximize the items' relevances	×	Xiang et al. [7]
	Maximize the items' relevances	×	Guan and Li [24]
	Maximize the items' relevances	×	Kim et al. [23]
	Minimize robots' movement distance	×	Mirzaei et al. [25]
	Maximize the items' relevances and minimize pickers' energy expenditure	×	Zhang et al. [3]
PSAP	Minimize robots' movement distance	√	Wang et al. [10]
	Minimize robots' movement cost	×	Keung et al. [26]
Two-stage optimization of ISAP and PSAP	ISAP: Maximize the items' relevances PSAP: Minimize robots' movement distance	√	Li et al. [30]
	ISAP: Maximize the items' relevances PSAP: Minimize robots' movement distance	√	Yuan et al. [9]
Joint optimization of ISAP and PSAP	Minimize robots' movement distance	√	This article



Order	SKU	Pod	Distance
1	1 4 5	F	5
2	2 4 5	E	3
3	1 2 3 4 5	D E	6

Figure 2: Illustration of the J-IPSAP with workload balance.

An illustration of the J-IPSAP is shown in Figure 2. There are five SKUs and six pods, and the storage locations for SKUs 1~5 are 7, 6, 5, 3, and 3, respectively. The solutions of ISAP and PSAP can be seen in Figure 2. To pick order 1, pod *F* is selected, and the robot's movement distance is 5. To pick order 2, pod *E* is selected, and the distance is 3. To pick order 3, pods *D* and *E* are selected, and the distance is 6. The total movement distance is 14. Meanwhile, the workloads of picking aisles 1 and 2 are 1 and 3, respectively, which is an unbalanced workload distribution. Therefore, the suitable solutions of SAP and pod selection are both important to minimize the movement distance and balance the workloads within picking aisles.

According to the complexity of the J-IPSAP, the following assumptions are made:

(i) The system is a single-deep layout and has only one picking station.

(ii) The pod has 8 layers, each layer can store only one type of item, and each item can be stored in multiple pods.

(iii) No empty storage location is allowed on the pods.

(iv) Robots can only carry one pod at a time.

(v) Sufficient robots are available in the warehouse.

(vi) As shown in Figure 2, the distances between the storage locations and the picking stations are known. Thus, the path planning problem is not considered.

(vii) The selected pods will return to their original storage positions, which is the same as other SAP research [3, 7, 9, 23, 30].

(viii) Robot energy consumption and path conflict problems are not considered.

3.2. *Mathematical Model of J-IPSAP.*

The notations are defined in Table 2.

The decision variables are defined in Table 3.

The J-IPSAP can be formulated as follows:

$$\min \quad \sum_{i\in R}\sum_{j\in B}\sum_{l\in L} y_{ij} * z_{il} * C_l, \tag{1}$$

s.t.

$$\sum_{i\in R} x_{\mathrm{ijk}} = D_{\mathrm{jk}} \quad \forall j \in B, k \in K, \tag{2}$$

$$x_{\mathrm{ijk}} \leq q_{\mathrm{ik}} \quad \forall i \in R, j \in B, k \in K, \tag{3}$$

$$\mathrm{M} * y_{\mathrm{ij}} \geq \sum_{k\in K} x_{\mathrm{ijk}} \quad \forall i \in R, j \in B, \tag{4}$$

$$\sum_{l\in L} z_{\mathrm{il}} = 1 \quad \forall i \in R, \tag{5}$$

$$\sum_{i\in R} z_{\mathrm{il}} \leq 1 \quad \forall l \in L, \tag{6}$$

$$\sum_{k\in K} q_{\mathrm{ik}} = Q \quad \forall i \in R, \tag{7}$$

$$\sum_{i\in R} q_{\mathrm{ik}} = p_k \quad \forall k \in K, \tag{8}$$

Table 2: The notations used in the J-IPSAP.

Parameters	Meaning	Value
K	Set of items	$k \in K$
R	Set of pods	$i \in R$
Q	Set of pod layers	$q \in Q$
B	Set of orders	$j \in B$
L	Set of pod locations	$l \in L$
T	Set of picking aisles	$t \in T$ $(\lvert T\rvert \geq 3)$
σ	The value of the threshold weight coefficient	$\sigma \geq 0$
B_k	The number of orders to purchase item k	$B_k \geq 0$
$O_{\text{kk}'}$	The number of orders to purchase both items k and k'	$O_{\text{kk}'} \geq 0$
r_k	The purchase frequency of item k	$r_k = B_k/\lvert B\rvert$
D_{jk}	If order j contains item k, it equals 1, otherwise, it equals 0	$D_{jk} \in \{0, 1\}$
N_{lt}	If location l is in aisle t, it equals 1, otherwise, it equals 0	$N_{lt} \in \{0, 1\}$
p_k	The number of storage locations of item k	$p_k = \lvert R\rvert * \lvert Q\rvert * r_k/\sum_{k\epsilon K} r_k$
$S_{\text{kk}'}$	The relevance coefficient of items k and k', when $k = k'$, $S_{\text{kk}'} = 0$, otherwise $S_{\text{kk}'} = O_{\text{kk}'}/B$	$S_{\text{kk}'} \in [0, 1]$
C_l	The distance from location l to the picking station	$C_l > 0, \forall l \in L$

Table 3: The decision variables used in the J-IPSAP.

Decision variables	Meaning	Value
M_{ikq}	If item k is on layer q of pod i, it equals 1; otherwise, it equals 0	$M_{\text{ikq}} \in \{0, 1\}$
x_{ijk}	If order j is fulfilled by pod i, and item k is on pod i, it equals 1; otherwise, it equals 0	$x_{\text{ijk}} \in \{0, 1\}$
y_{ij}	If order j is fulfilled by pod i, it equals 1; otherwise, it equals 0	$y_{\text{ij}} \in \{0, 1\}$
z_{il}	If pod i is assigned to location l, it equals 1, otherwise, it equals 0	$z_{\text{il}} \in \{0, 1\}$
q_{ik}	The number of layers of pod i with item k	$q_{\text{ik}} \in Z$
X_{ijl}	If the order j is fulfilled by pod i, and pod i is allocated to location l, it equals 1; otherwise, it equals 0	$X_{\text{ijl}} \in \{0, 1\}$

$$\sum_{q \in Q} M_{\text{ikq}} = q_{\text{ik}} \quad \forall i \in R, k \in K, \tag{9}$$

$$\sum_{i \in R} \sum_{j \in B} \sum_{l \in L} y_{\text{ij}} * z_{\text{il}} * N_{\text{lt}} \leq \frac{\sum_{i \in R} \sum_{j \in B} y_{\text{ij}}}{\text{T} * \sigma} \quad \forall t \in T, \tag{10}$$

$$y_{\text{ij}}, z_{\text{il}}, x_{\text{ijk}}, M_{\text{ikq}} \in \{0, 1\} \quad \forall i \in R, j \in B, l \in L, k \in K, q \in Q, \tag{11}$$

$$q_{\text{ik}} \in Z \quad \forall i \in R, k \in K. \tag{12}$$

Equation (1) is to minimize the robots' total movement distance to carry the selected pods from their assigned locations to the picking station. Equation (2) means that customer demand for item k in order j must be satisfied. Equation (3) indicates the relationship between x_{ijk} and q_{ik}. In equation (4), M is a large value to guarantee the relationship between x_{ijk} and y_{ij}. Equation (5) indicates that the pod must be allocated to only one storage location. Equation (6) indicates that no more than one pod can be allocated to each location. Equation (7) indicates that the items in each pod are Q. Equation (8) means that the number of storage locations of item k is p_k. Equation (9) indicates that the total storage locations of item k on pod i are equal to q_{ik}. Equation (10) is the workload balance constraint, which indicates that the pods' carrying times in each aisle are less than a threshold. Equations (11) and (12) are the basic constraints.

Since the SAP is an NP-hard problem [31], this article proposes the IGA to solve the mixed integer programming model.

4. Algorithm Design

Section 4 designs the improved genetic algorithm (IGA) to solve the J-IPSAP model. The framework of the IGA is introduced in Section 4.1, and then the initial solution, pod selection mechanism, pod assignment strategy, and other improvement mechanisms are introduced in Sections 4.2–4.7, respectively.

4.1. Framework Design. The genetic algorithm (GA), proposed by Holland [32], has strong robustness and can search whole solutions in the solution space quickly. Aiming at the characteristics of the J-IPSAP, this article proposes an IGA to solve the J-IPSAP model. The main algorithm framework of the IGA is shown in Algorithm 1.

4.2. Initialization of Item Storage Location Solutions. M_{ikq} in the model is three-dimensional. To facilitate the crossover and mutation operations, this article transforms M_{ikq} into a two-dimensional real variable M_{iq}, represents the relationship among pods, layers of the pods, and items. As shown in Figure 3, the rows of the individual indicate pods and the columns indicate layers of the pods. The value in the matrix indicates the item stored in layer q of pod i. The initial population is obtained as follows:

Step 1: The purchase frequency of item k can be calculated by $r_k = B_k/|B|$. $p_k = |R| * |Q| * r_k/\sum_{k\epsilon K} r_k$.

Step 2: Find a pair of items k and k' with the largest S, randomly assign them to the bits of the same pods, and update the number of storage locations p_k and $p_{k'}$. If p_k and $p_{k'}$ are equal to 0, find the next pair of items with the largest S. If the remaining storage locations of one of the items (for example k') is zero, then the other item k is assigned separately to the idle bit.

Step 3: Continue with the previous step until all the bits have been allocated to the items.

4.3. Pod Selection Mechanism. Based on the ISAP solutions, this article needs to find the pod selection solutions to satisfy the set of orders B. To solve the pod selection problem, Xiang et al. [7] first used the greedy algorithm to obtain the initial pod set and then used the local search algorithm to improve the solution. This mechanism is shown in Algorithm 2. An initial pod selection solution W is obtained by using the greedy strategy (*Greedy_strategy* (P, N)), and the SwapPod operator (*SwapPod* (W)) and the DeletePod operator (*DeletePod* (W)) are executed if $|W| \geq 3$; otherwise, the optimal solution W is obtained. Refer to Xiang et al. [7] for the details.

4.4. Pod Assignment Strategy with Workload Balance Constraint. Different from the related literature that considers only the pod turnover rate in PSAP, the decentralized pod storage assignment strategy (DPSAS), which considers both the pod turnover rate and picking aisles' workload balance, is designed to solve the PSAP.

First, the number of pod visits α (i) is calculated according to the pod selection solutions in Section 4.3. α (i) is sorted in descending order. Second, pod i with the highest turnover rate α (i) is allocated to position l with the minimal C_l, and aisle t' of location l is recorded. After that, the closest locations in other aisles are allocated to the next pods in sequence, until all the aisles are selected once. Finally, repeat the above location selection mechanism until all the pods have been assigned to the locations. The pod storage assignment solution S is obtained. The DPSAS can balance the workload of the aisles and reduce the robots' movement distance effectively.

4.5. Fitness Function. Equation (10) is designed to balance the workload of each aisle. If the solution satisfies equation (10), the fitness function of IGA is the objective function in the J-IPSAP model. Otherwise, a penalty cost will be added to its fitness function [33]. The penalty cost is identified as the range of the workload for the aisles multiplied by the penalty coefficient δ, shown as follows:

$$\left[\left(\sum_{i\epsilon R}\sum_{j\epsilon B}\sum_{l\epsilon L} y_{\mathrm{ij}} * z_{\mathrm{il}} * N_{\mathrm{lt}}\right)_{\max} - \left(\sum_{i\epsilon R}\sum_{j\epsilon B}\sum_{l\epsilon L} y_{\mathrm{ij}} * z_{\mathrm{il}} * N_{\mathrm{lt}}\right)_{\min}\right] * \delta. \tag{13}$$

As denoted by Matl et al. [34], the range is usually been used as the measurement of equity. The penalty coefficient δ is set to be 0.25, which is the same as Zhuang et al. [16].

4.6. Variation Function

4.6.1. Crossover Operator. The crossover operator of the algorithm adopts a two-point crossover to operate on the two selected individuals. In Figure 4, Bit1 and Bit2 are randomly selected in individual S_1, and Bit1' and Bit2' are randomly selected in individual S_2. The fragments between two bits are then swapped to generate new individuals S_1' and S_2'.

Let p_k be the total amount of storage that should be allocated by item k, and then the number of bits of item k in a feasible individual is equal to p_k. The new individuals generated after the crossover may be infeasible. This article refers to the method of Jiang et al. [8] to repair infeasible individuals. The infeasible individuals are repaired by swapping the items' storage locations of the redundant item set and the items' storage locations of the absent item set. Refer to Jiang et al. [8] for the specific steps.

Figure 4 shows the process of the crossover operator. The types of items and p_k ($k = 1, \ldots, 5$) are given. Figure 5 shows the correction of two infeasible solutions into feasible ones. For example, individual S_1' is infeasible with the redundant item set {1, 1, 4} and the absent item set {2, 3, 5}. Finally, the correct individuals S_1'' and S_2'' are acquired.

4.6.2. Mutation Operator. The mutation operator randomly chooses two genes among individuals and swaps them. The number of storage locations per item is constant in this operator and the generated new individual is feasible. In Figure 6, two genes are randomly selected within individual S_1'' and swapped positions with a certain mutation probability. Finally, a new feasible individual S_1''' can be obtained.

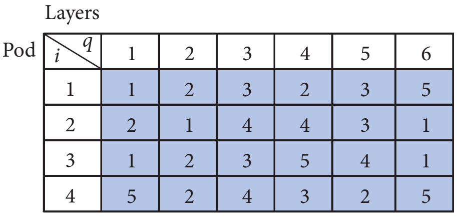
4.7. Elite Preservation Strategy. The elite preservation strategy (*Preserv_mechanism* (P, P'', N)) is performed on the parent population P and the offspring population P'', aiming to obtain the excellent new population. The specific method of the elite preservation strategy is to select the top 20% of the parent from the P and the top 80% of the updated population of the P'' to form the new population, which can be more efficient and easier to get the optimal solution.

5. Computational Experiments

The details of the experiment setting are in Section 5.1. The effectiveness of the proposed IGA is analyzed with the Gurobi, the two-stage optimization algorithm, and the related literature [9] in Section 5.2. Section 5.3 indicates the sensitivity analysis by analyzing the effect of the layout of the

```
Require: $P$ (population), $N$ (population size), $Pc$ (crossover probability), $Pm$ (mutation probability), $\theta$ (iteration number)
$P \leftarrow Initialize\ (N)$
for ($i = 1$; $i \leq \theta$; $i$++) do
    $P, N, W \leftarrow Pod\_Selection\ (P, N)$
    $P, N, S \leftarrow Pod\_Assignment\ (P, N, W)$
    $P \leftarrow Fitness\ (P, N, S)$
    $P' \leftarrow Tournament\_selection\ (P, N)$/*Offspring individuals selected based on the binary tournament strategy*/
    $P'' \leftarrow Crossover\ (P', Pc, N)$
    $P'' \leftarrow Mutation\ (P'', Pm, N)$
    $P \leftarrow Preserve\_mechanism\ (P, P'', N)$
    end for
    return $P$
```

Algorithm 1: Improved genetic algorithm (IGA).



i \ q	1	2	3	4	5	6
1	1	2	3	2	3	5
2	2	1	4	4	3	1
3	1	2	3	5	4	1
4	5	2	4	3	2	5

Figure 3: The encoding of the solution M_{iq} of the J-IPSAP.

```
Require: $P$ (population), $N$ (population size), $B$ (set of orders)
Step 1: Construct an initial pod combination $W$ by the greedy strategy
for ($j = 1$; $j \leq |B|$; $j$++) do
    while $B(j) = \varnothing$
        $W \leftarrow Greedy\_strategy\ (P, N)$
    end while
Step 2: Improve the solution
if ($|W| \geq 3$) then
        $W \leftarrow SwapPod\ (W)$
        $W \leftarrow DeletePod\ (W)$
    end if
    end for
```

Algorithm 2: Pod selection mechanism.

storage area and the effect of the workload balance constraint. Some management insights are presented in Section 5.4.

5.1. Experiment Description. Scholars have studied SAP based on different warehouse layouts. The order sizes B are 500, 1000, 1500, and 2000 in Yuan et al. [9]. Zhuang et al. [16] set the number of pod locations to be {10, 20, 40, 80, 800, 1200}. Zhang et al. [3] set four different instances with the combinations of the item (K), pod (R), and order (B): $K/R/B \in$ {10/10/1000}, {100/100/10000}, {200/200/100000}, and {500/500/1000000}. This article refers to order size in Yuan et al. [9] and pod number in Zhang et al. [3] to set three different classes of instances with the combination of $K/R/L/T$, where $K/R/L/T =$ {20/10/16/3} for the small-scale instance, $K/R/L/T =$ {250/200/240/7} for the medium-scale instance, and $K/R/L/T =$ {500/400/448/9} for the large-scale instance. The order sizes (B) are 20, 100, 500, 1000, and 1500 for different scale instances. The storage capacity of pod Q is 8, the same as Yuan et al. [9].

The order information is referred to the online retailers' dataset proposed by Guo et al. [35]. It is found that the online retailers' item requirements satisfy the ABC rule, and the items' number per order satisfies U [1, 3]. Based on these characteristics, randomly generate order sets at different order sizes {20, 100, 500, 1000}. Based on reality, every five orders are combined into one batch and sent to the picking station. σ is 0.7, and the effect of different values of σ is discussed in Section 5.3.2. The hyperparameter settings of the IGA are discussed in Section 5.2.1. This article codes with MATLAB R2022a, running on a PC with a Core i7 CPU @ 3.20 GHz, 32.0 GB RAM, and Windows 10.

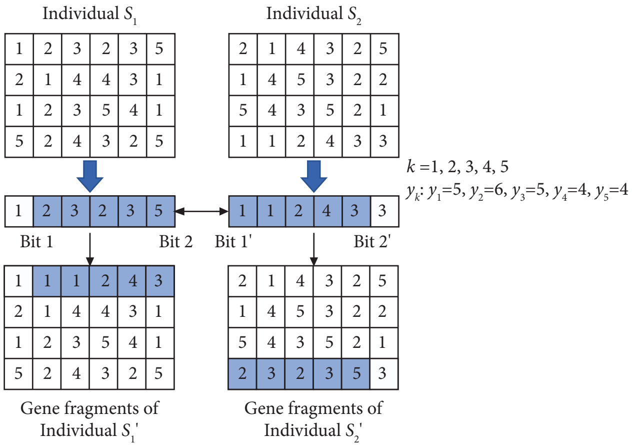


Figure 4: The crossover processes.

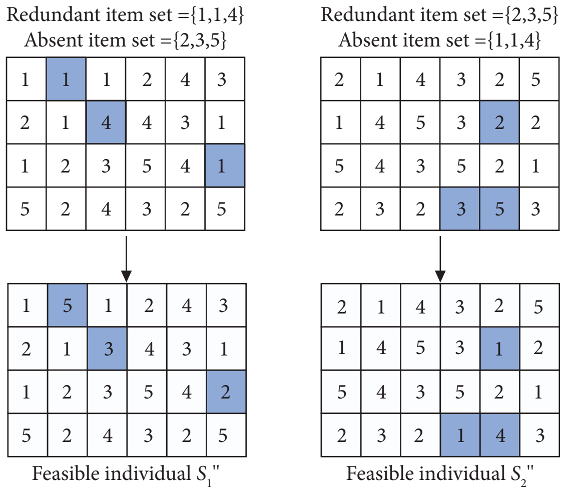


Figure 5: Correction of two infeasible individuals.

5.2. Results Analysis

5.2.1. Comparative Analysis with Gurobi Optimization Solutions in Small-Scale Problem. Before verifying the effectiveness of the IGA, we discuss the hyperparameter settings of crossover probability (Pc) and mutation probability (Pm) of the IGA. Four combinations of Pc and Pm are conducted in small-scale instances. As far as we know, if Pc is too large, the randomness of GA will increase and lose the good individuals; if Pm is too small, the diversity of the population will be less [32]. Therefore, we set $Pc/Pm = \{0.2/0.05, 0.4/0.1, 0.6/0.15, 0.8/0.2\}$. According to the problem scale, the value of the population size is 50. From the results shown in Figure 7, we find that $Pc/Pm = \{0.8/0.2\}$ gets the minimum objective value and the fastest convergence speed. Therefore, $Pc/Pm = \{0.8/0.2\}$ is applicable in this paper. The hyperparameter settings are also the same with Guan and Li [24].

The optimization solver Gurobi is used to verify the correctness of the J-IPSAP model and the effectiveness of IGA. To make Gurobi find the best solution in a limited time, the J-IPSAP model is simplified by deleting equation (10).

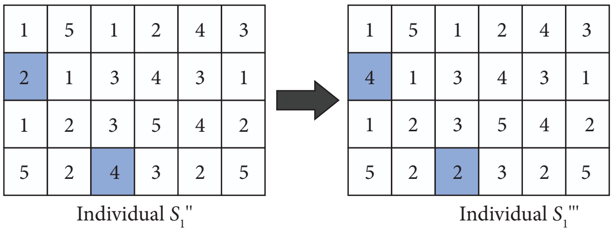


Figure 6: The mutation processes.

Since the objective function $y_{\mathrm{ij}}z_{\mathrm{il}}$ is nonlinear, let set $X_{\mathrm{ijl}} = y_{\mathrm{ij}}z_{\mathrm{il}}$, and replace it with the following equations:

$$X_{\mathrm{ijl}} \le y_{\mathrm{ij}} \quad \forall i \in R, j \in B, l \in L, \tag{14}$$

$$X_{\mathrm{ijl}} \le z_{\mathrm{il}} \quad \forall i \in R, j \in B, l \in L, \tag{15}$$

$$X_{\mathrm{ijl}} \ge y_{\mathrm{ij}} + z_{\mathrm{il}} - 1 \quad \forall i \in R, j \in B, l \in L, \tag{16}$$

$$X_{\mathrm{ijl}} \in \{0, 1\} \quad \forall i \in R, j \in B, l \in L. \tag{17}$$

Table 4 lists the results of Gurobi and IGA in small-scale instances, where the objective value (Obj) and CPU time (in seconds) are reported. The Obj is the robot's movement distance. When order sizes B are 20 and 100, both Gurobi and IGA can get the global optimal solution, and the running time of IGA is slightly less than that of Gurobi. As the order size increases, the Gap values of Obj between Gurobi and IGA become larger, but the maximum value does not exceed 1.93% (shown in the bold value in Table 4). The CPU time of Gurobi is much longer than that of IGA. When B is 1500, Gurobi is unable to get the solution even after running for many days.

5.2.2. Comparative Analysis with Two-Stage Optimization Algorithms under Different Scale Instances. This section compares the results of IGA with those of the two-stage optimization algorithms under different scale instances, shown in Table 5. Two-stage-1 refers to solving the J-IPSAP model in two stages. In the ISAP stage, the initial solution is first obtained using the greedy algorithm, and then an optimal ISAP solution is obtained using the GA. Similar to most ISAP literature, the objective considers the items' relevance. In the PSAP stage, the pod storage solution is obtained according to the pod selection mechanism and pod assignment strategy of this article. Two-stage-2 refers to the ISAP model and PSAP model established by Yuan et al. [9]. The Obj is the movement distance plus the penalty cost.

In the three instances, the Obj of IGA is smaller than that of Two-stage-1 and Two-stage-2. The majority of results do not violate the workload balancing constraint. As the order size increases, the gaps between J-IPSAP and the other two algorithms decrease. The bold values in Table 5 show the minimum Gap% in each instance. In the small-scale instances, Two-stage-1 performs slightly better than Two-stage-2. The highest Gap_1 for Two-stage-1 is 81.82%, and the lowest is 13.13%. This is because in small-scale instances, the number of pods is small, and the search space of the solution is limited. However, as the

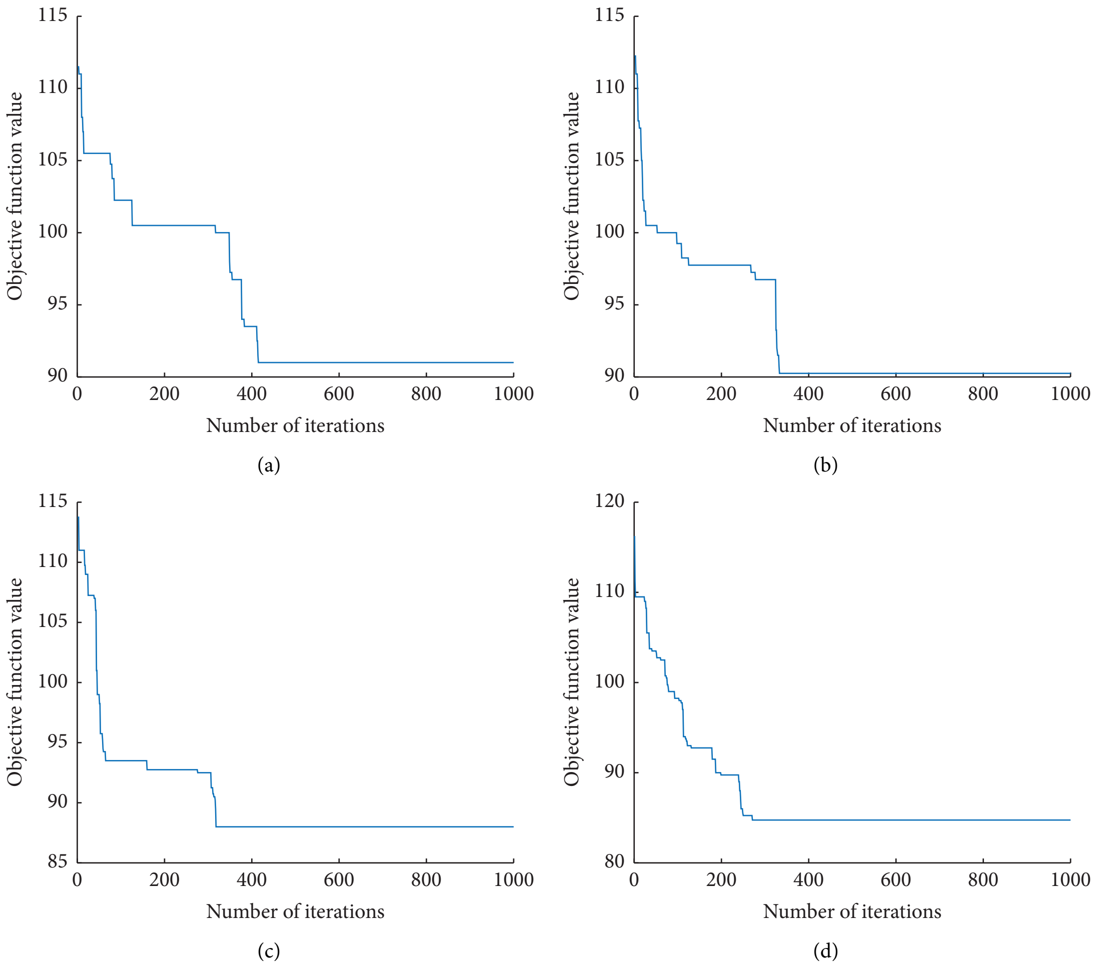


Figure 7: Sensitivity analysis of hyperparameters *Pc*/*Pm*. (a) $Pc = 0.2$, $Pm = 0.05$, (b) $Pc = 0.4$, $Pm = 0.1$, (c) $Pc = 0.6$, $Pm = 0.15$, and (d) $Pc = 0.8$, $Pm = 0.2$.

Table 4: Results of Gurobi and IGA for the small-scale instances.

K	*R*	*L*	*T*	*B*	Gurobi		IGA		
					Obj	Time (sec)	Obj	Gap (%)	Time (sec)
20	10	16	3	20	12	12.35	12	0.00	11.69
20	10	16	3	100	79	93.27	79	0.00	78.63
20	10	16	3	500	497	23913	501	0.80	2756.61
20	10	16	3	1000	1036	37163	1056	**1.93**	3573.97
20	10	16	3	1500	—	—	1632	—	4667.53

The bold value in Table 4 indicates the maximum Gap of IGA and Gurobi.

problem scale increases, the global search capability of Two-stage-2 gradually increases and Gap_2 becomes smaller. It reaches a minimum gap of 6.66% in large-scale instances.

5.3. Sensitivity Analysis

5.3.1. Effect of the Layout of the Storage Area. This section analyzes the effect of the layout of the storage area. As the number of pods in small-scale instances is too few, the experiments are conducted on medium-scale and large-scale instances.

Table 6 shows different layouts of the storage area in medium- and large-scale instances. Different numbers of picking aisles T represent different layouts of the storage area. For example, when $|L|$ is 240, $T = 3$ represents that there are 60 pods in each row and 4 pods in each column, and the number of picking aisles is 3. In Figures 8(a) and 8(b), with the increase of T, the robots' movement distance decreases first and then increases. The optimal width-to-length ratios in medium- and large-scale instances are 15 : 16 and 24 : 20, respectively.

It can be concluded that the optimal layout of the storage area is the one that the width-to-length ratio tends

Table 5: Comparative results among the IGA, Two-stage-1, and Two-stage-2.

Instances	K	R	L	T	B	IGA		Two-stage-1			Two-stage-2		
						Obj	Distance	Obj$_1$	Distance	Gap$_1$ (%)	Obj$_2$	Distance	Gap$_2$ (%)
Small-scale	20	10	16	3	20	13.75	13	25	25	81.82	30	30	118.18
	20	10	16	3	100	86	86	125	125	45.35	131	131	52.33
	20	10	16	3	500	528	528	646	646	22.35	769	769	45.64
	20	10	16	3	1000	1120	1120	1288	1288	15.00	1506	1506	34.46
	20	10	16	3	1500	1713	1713	1938	1938	**13.13**	2301	2301	**34.33**
Medium-scale	250	200	240	7	20	49.50	49	95.50	95	92.93	84	84	69.70
	250	200	240	7	100	421	421	624	624	48.22	593	593	40.86
	250	200	240	7	500	3941	3941	4839	4839	22.79	4546	4546	15.35
	250	200	240	7	1000	9105	9105	10188	10188	11.89	9962	9962	9.41
	250	200	240	7	1500	14484	14484	15729	15729	**8.60**	15482	15482	**6.89**
Large-scale	500	400	448	9	20	114.50	114	206.50	206	80.35	130	130	13.54
	500	400	448	9	100	1013	1013	1589	1589	56.86	1149	1149	13.43
	500	400	448	9	500	11155	11155	12881	12881	15.47	12269	12269	9.99
	500	400	448	9	1000	27197	27197	30207	30207	11.07	29546	29546	8.64
	500	400	448	9	1500	44105	44105	47736	47736	**8.23**	47041	47041	**6.66**

Note. Gap$_i$ = (Obj$_i$ − Obj/Obj); Distance: the fitness function value without considering the penalty cost. The bold values in Table 5 indicate the minimum Gap of two-stage-1, two-stage-2, and IGA, respectively.

Table 6: The layouts of the storage area in medium- and large-scale instances.

Instances	$\|K\|$	$\|R\|$	$\|L\|$	$\|B\|$	T (width : length)
Medium-scale	250	200	240	500	3 (60 : 4) 5 (30 : 8) 7 (20 : 12) 9 (15 : 16) 11 (12 : 20)
Large-scale	500	400	480	1500	3 (120 : 4) 5 (60 : 8) 7 (40 : 12) 9 (30 : 16) 11 (24 : 20) 13 (20 : 24) 16 (16 : 30)

to be 1. Note that this conclusion is different from that of Duan et al. [36]. Duan et al. [36] found out that the order throughput time is shortest when the width-to-length ratio is 0.5. This is because in Duan et al. [36], the aisles are unidirectional and the warehouse is asymmetric. However, in this article, aisles are bidirectional and the warehouse is symmetrical.

5.3.2. Effect of the Workload Balance Constraint. The J-IPSAP model uses σ to balance the workload of each picking aisle. This section further analyzes the influence of different σ, $\sigma = 0$, 0.5, 0.55, 0.6, 0.65, and 0.7. Note that there is no penalty cost when $\sigma = 0$~0.7, which means the model does not violate the workload balance constraint.

Table 7 shows the results in different scale instances. The variance is used to measure the workload balance of different picking aisles. It can be found that with the increase of σ, the value of the objective function increases and the variance of the workload in different picking aisles decreases. The minimum Obj and Variance in each instance are shown bolded in Table 7. The reasons are that with the increase of σ, the workload balance constraint is tighter, and the workload of each picking aisle is more balanced. The larger σ is helpful to ease the aisle congestion, but the robot's movement distance will increase. Thus, it is recommended to set σ to around 0.5~0.6 to balance both the robot's movement distance and the aisle congestion.

Figure 9 shows the work intensity of the pods when $\sigma = 0$ and 0.7 in the medium-scale instance. The red color indicates the pods with the largest movement times, and the green color indicates the pods with the smallest movement times. When $\sigma = 0$, the thermodynamic diagram shows a "*V*" shape, and when $\sigma = 0.7$, the thermodynamic diagram shows an "*M*" shape. It can be seen that when $\sigma =$ (no workload balance constraint), the high turnover pods (red pods) are scattered closer to the picking station, thus the movement distance is smallest. When $\sigma = 0.7$, the high turnover pods (red pods) are scattered in different aisles; thus the movement distance is largest, but the workload is most balanced. This also reflects the decentralized pod storage assignment strategy proposed in the study.

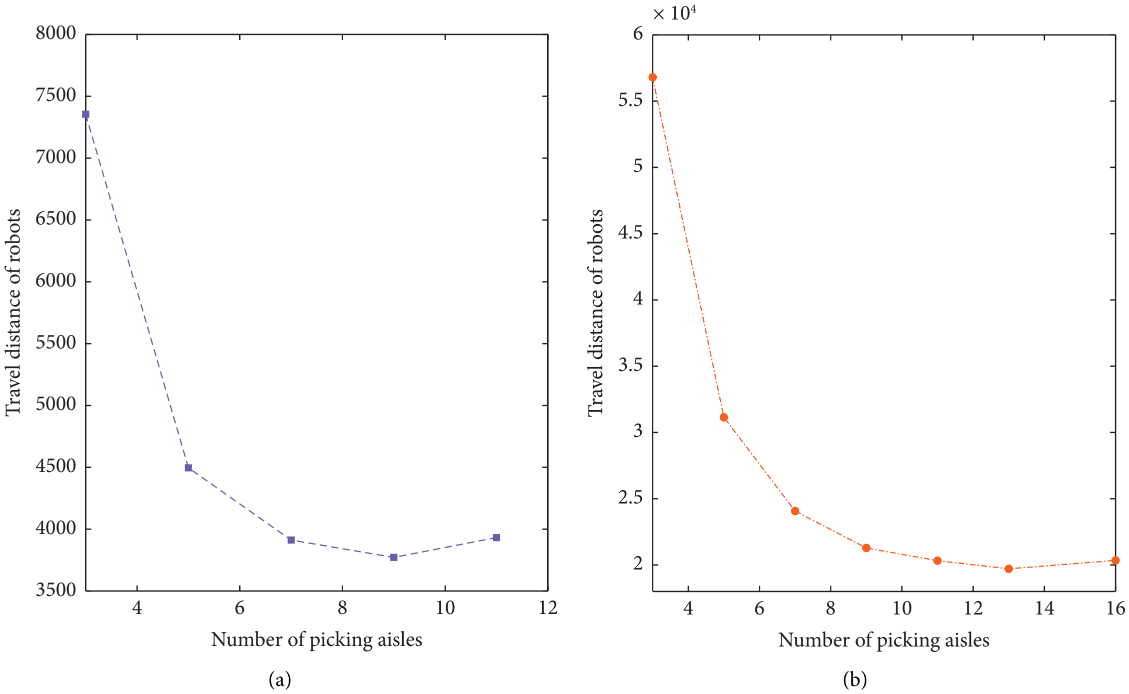


Figure 8: Robots' movement distance under different layouts in (a) medium-scale instance and (b) large-scale instance.

Table 7: Results of the influence of picking aisles' workload in different scale instances.

Instances	K	R	L	B	σ	Obj	Distance	Variance
Small-scale	20	10	16	100	0	**79**	**79**	62
					0.5	82	82	24.67
					0.55	83	83	13.56
					0.6	84	84	12.67
					0.65	85	85	8
					0.7	86	86	**6**
Medium-scale	250	200	240	500	0	**3638**	**3638**	845.84
					0.5	3896	3896	16.86
					0.55	3908	3908	12.82
					0.6	3939	3939	10.53
					0.65	3946	3946	9.96
					0.7	4089	4089	**9.43**
Large-scale	500	400	448	1500	0	**43077**	**43077**	992.91
					0.5	44105	44105	225.21
					0.55	44385	44385	204
					0.6	44538	44538	194.62
					0.65	44730	44730	185.21
					0.7	45160	45160	**165.06**

The bold values in Table 7 indicate the optimal values of obj, distance, and variance at the three instances.

5.4. Discussion. This article can provide some management insights as follows:

(1) It is necessary to jointly optimize the item and pod storage location problems, which can decrease the robots' movement distance and improve the performance of RMFS. The J-IPSAP makes the robot's movement distance smaller than one of the two-stage optimization models.

(2) The width-to-length ratio of the storage area has a great impact on the robots' movement distance. It is recommended to set the width-to-length ratio of the storage area is 1, which can further decrease the robots' movement distance.

(3) Managers should balance the robots' movement distance and the picking aisles' workload in the J-IPSAP. The more stringent the workload balance

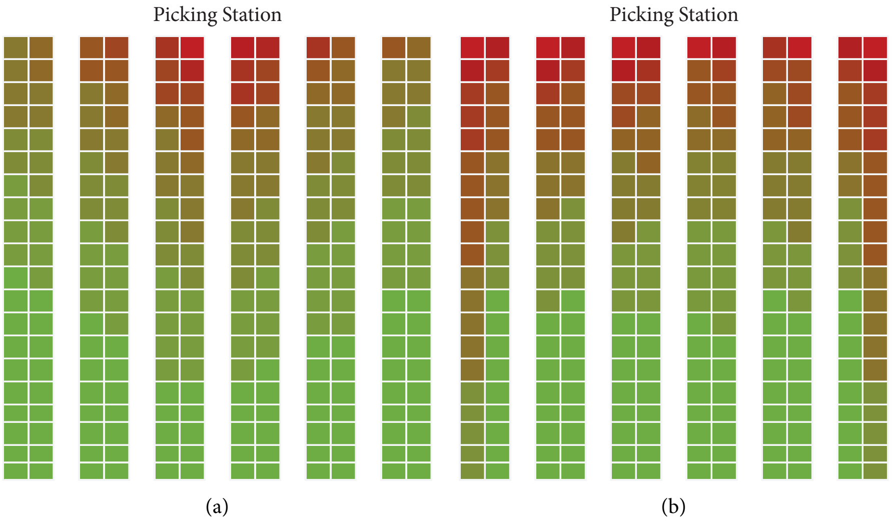


Figure 9: Thermodynamic diagrams of the working intensity of the pods. (a) $\sigma = 0$ and (b) $\sigma = 0.7$.

constraint is, the larger the robots' movement distance is, but the more balanced the workloads within picking aisles are. It is recommended to set σ to around 0.5~0.6 to balance both the robot's movement distance and the aisle congestion.

6. Conclusions and Future Work

This article studies the joint optimization of ISAP and PSAP (J-IPSAP) in RMFS by considering the workload balance within the picking aisles. The J-IPSAP mathematical model is formulated to minimize the robots' movement distance. The workload balance constraint is added to the model. By proposing the decentralized pod storage assignment strategy, this article designed the improved elite-preservation genetic algorithm to effectively solve the J-IPSAP. The results indicate that the IGA has good performance in different scale instances. Through sensitivity analysis, it can be found that (1) when the width-to-length ratio of the storage area is close to 1, the robot's movement distance is the smallest, and (2) the more stringent workload balance constraint will increase the robot's movement distance but ease the aisle congestion. The contribution of this article is to extend SAP by jointly optimizing the ISAP and PSAP to improve the picking efficiency of RMFS. In addition, the workload balance within the picking aisles is considered, and a new decentralized pod storage allocation strategy is proposed to solve the workload balance problem of robots to avoid aisle congestion. Finally, the J-IPSAP model is efficiently solved by the designed IGA, which proves that the joint optimization is better than other algorithms.

In theoretical implications, the J-IPSAP model with picking aisles' workload balance is efficient in reducing the robots' movement distance and balancing the workload in each picking aisle. In practical implications, there are several decision recommendations for managers. (1) Managers are recommended to decide the item and pod storage location problems together when planning the storage locations. (2) The picking aisles' workload balance is recommended to be considered to reduce aisles' congestion. (3) The decentralized pod storage assignment strategy designed in this paper is efficient in deciding the pod storage assignment problem with workload balance constraints. (4) The width-to-length ratio of the storage area is recommended to be set to 1, and σ is recommended to be set to around 0.5~0.6 to balance the robots' movement distance and picking aisles' workload.

The research still has some limitations and needs further improvement in the future. First, this article considers only one picking station. Besides, the replenishment process and the human factors of the pickers are not considered in the picking process. Future research can consider dynamic PSAP in a replenishment state. The human factors can be considered in the human-robot collaborative picking system.

Data Availability

The data that support the findings of this study are openly available in https://data.mendeley.com/datasets/tfh8rcwk2w/1.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Authors' Contributions

Jun Zhang conceptualized the study, proposed the methodology, and wrote, reviewed, and edited the article. Lingkun Tian wrote the original draft and was responsible for software. Zijuan Zhou performed formal analysis, contributed to visualization, and wrote, reviewed, and edited the article.

Acknowledgments

This research was supported by the Ministry of Education Social Sciences and Humanities Research Fund of China (22YJA630110) and the Hubei Provincial Natural Science Foundation of China (2022CFB315).

References

[1] S. Banker, “Robots in the warehouse: it’s not just amazon,” 2016, https://www.forbes.com/sites/stevebanker/2016/01/11/robots-in-the-warehouse-its-not-just-amazon/.

[2] N. Boysen, R. de Koster, and D. Füßler, “The forgotten sons: warehousing systems for brick-and-mortar retail chains,” *European Journal of Operational Research*, vol. 288, no. 2, pp. 361–381, 2021.

[3] J. Zhang, N. Zhang, L. Tian, Z. Zhou, and P. Wang, “Robots’ picking efficiency and pickers’ energy expenditure: the item storage assignment policy in robotic mobile fulfillment system,” *Computers and Industrial Engineering*, vol. 176, Article ID 108918, 2023.

[4] G. Jiao, H. Li, and M. Huang, “Online joint optimization of pick order assignment and pick pod selection in robotic mobile fulfillment systems,” *Computers & Industrial Engineering*, vol. 175, Article ID 108856, 2023.

[5] M. Merschformann, T. Lamballais, M. B. M. de Koster, and L. Suhl, “Decision rules for robotic mobile fulfillment systems,” *Operations Research Perspectives*, vol. 6, Article ID 100128, 2019.

[6] L. Xie, N. Thieme, R. Krenzler, and H. Li, “Introducing split orders and optimizing operational policies in robotic mobile fulfillment systems,” *European Journal of Operational Research*, vol. 288, no. 1, pp. 80–97, 2021.

[7] X. Xiang, C. Liu, and L. Miao, “Storage assignment and order batching problem in Kiva mobile fulfilment system,” *Engineering Optimization*, vol. 50, no. 11, pp. 1941–1962, 2018.

[8] W. Jiang, J. Liu, Y. Dong, and L. Wang, “Assignment of duplicate storage locations in distribution centres to minimise walking distance in order picking,” *International Journal of Production Research*, vol. 59, no. 15, pp. 4457–4471, 2020.

[9] R. Yuan, J. Li, W. Wang, J. Dou, and L. Pan, “Storage assignment optimization in robotic mobile fulfillment systems,” *Complexity*, vol. 2021, Article ID 4679739, 11 pages, 2021.

[10] Y. Wang, R. Man, W. Zhao, H. Zhang, and H. Zhao, “Storage assignment optimization for fishbone robotic mobile fulfillment systems,” *Complex and Intelligent Systems*, vol. 8, no. 6, pp. 4587–4602, 2022.

[11] X. Yang, G. Hua, L. Hu, T. C. E. Cheng, and A. Huang, “Joint optimization of order sequencing and rack scheduling in the robotic mobile fulfilment system,” *Computers and Operations Research*, vol. 135, Article ID 105467, 2021.

[12] N. Yang, “Evaluation of the joint impact of the storage assignment and order batching in mobile-pod warehouse systems,” *Mathematical Problems in Engineering*, vol. 2022, Article ID 9148001, 13 pages, 2022.

[13] I. Žulj, H. Salewski, D. Goeke, and M. Schneider, “Order batching and batch sequencing in an AMR-assisted picker-to-parts system,” *European Journal of Operational Research*, vol. 298, no. 1, pp. 182–201, 2022.

[14] A. Gharehgozli and N. Zaerpour, “Robot scheduling for pod retrieval in a robotic mobile fulfillment system,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, Article ID 102087, 2020.

[15] S. Teck and R. Dewil, “Optimization models for scheduling operations in robotic mobile fulfillment systems,” *Applied Mathematical Modelling*, vol. 111, pp. 270–287, 2022.

[16] Y. Zhuang, Y. Zhou, E. Hassini, Y. Yuan, and X. Hu, “Rack retrieval and repositioning optimization problem in robotic mobile fulfillment systems,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 167, Article ID 102920, 2022.

[17] C. K. M. Lee, B. Lin, K. K. H. Ng, Y. Lv, and W. C. Tai, “Smart robotic mobile fulfillment system with dynamic conflict-free strategies considering cyber-physical integration,” *Advanced Engineering Informatics*, vol. 42, Article ID 100998, 2019.

[18] Y. Sun, N. Zhao, and G. Lodewijks, “An autonomous vehicle interference-free scheduling approach on bidirectional paths in a robotic mobile fulfillment system,” *Expert Systems with Applications*, vol. 178, Article ID 114932, 2021.

[19] K. L. Keung, C. K. M. Lee, and P. Ji, “Industrial internet of things-driven storage location assignment and order picking in a resource synchronization and sharing-based robotic mobile fulfillment system,” *Advanced Engineering Informatics*, vol. 52, Article ID 101540, 2022.

[20] Í. R. Da Costa Barros and T. P. Nascimento, “Robotic Mobile Fulfillment Systems: a survey on recent developments and research opportunities,” *Robotics and Autonomous Systems*, vol. 137, 2021.

[21] G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, “Planning and control of autonomous mobile robots for intralogistics: literature review and research agenda,” *European Journal of Operational Research*, vol. 294, no. 2, pp. 405–426, 2021.

[22] L. Luo and N. Zhao, “An efficient simulation model for layout and mode performance evaluation of robotic mobile fulfillment systems,” *Expert Systems with Applications*, vol. 203, Article ID 117492, 2022.

[23] H.-J. Kim, C. Pais, and Z.-J. M. Shen, “Item assignment problem in a robotic mobile fulfillment system,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1854–1867, 2020.

[24] M. Guan and Z. Li, “Genetic algorithm for scattered storage assignment in kiva mobile fulfillment system,” *American Journal of Operations Research*, vol. 08, no. 06, pp. 474–485, 2018.

[25] M. Mirzaei, N. Zaerpour, and R. de Koster, “The impact of integrated cluster-based storage allocation on parts-to-picker warehouse performance,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 146, Article ID 102207, 2021.

[26] K. L. Keung, C. K. M. Lee, and P. Ji, “Data-driven order correlation pattern and storage location assignment in robotic mobile fulfillment and process automation system,” *Advanced Engineering Informatics*, vol. 50, Article ID 101369, 2021.

[27] 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.

[28] T. Ji, K. Zhang, and Y. Dong, “Model-based optimization of pod point matching decision in robotic mobile fulfillment system,” in *Proceedings of the 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, Bangkok, Thailand, April 2020.

[29] J. Cai, X. Li, Y. Liang, and S. Ouyang, “Collaborative optimization of storage location assignment and path planning in robotic mobile fulfillment systems,” *Sustainability*, vol. 13, no. 10, p. 5644, 2021.

[30] X. Li, G. Hua, A. Huang, J. B. Sheu, T. C. E. Cheng, and F. Huang, “Storage assignment policy with awareness of energy consumption in the Kiva mobile fulfilment system,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 144, Article ID 102158, 2020.

[31] Z. P. Li, J. L. Zhang, and H. J. Zhang, “Optimal selection of movable shelves under cargo-to-person picking mode,” *International Journal of Simulation Modelling*, vol. 16, no. 1, pp. 145–156, 2017.

[32] J. H. Holland, *Adaptation in Natural and Arti Cial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, The MIT Press, Cambridge, MA, USA, 1992.

[33] K. Deb and R. Datta, "A bi-objective constrained optimization algorithm using a hybrid evolutionary and penalty function approach," *Engineering Optimization*, vol. 45, no. 5, pp. 503–527, 2013.

[34] P. Matl, R. F. Hartl, and T. Vidal, "Workload equity in vehicle routing problems: a survey and analysis," *Transportation Science*, vol. 52, no. 2, pp. 239–260, 2018.

[35] X. Guo, Y. Yu, G. Allon, M. Wang, and Z. Zhang, "Ririshun logistics: home appliance delivery data for the 2021 manufacturing and service operations management data-driven research challenge," *Manufacturing and Service Operations Management*, Articles in Advance,pp. 1–14, 2021.

[36] G. Duan, C. Zhang, P. Gonzalez, and M. Qi, "Performance evaluation for robotic mobile fulfillment systems with time-varying arrivals," *Computers and Industrial Engineering*, vol. 158, Article ID 107365, 2021.