systems

MDPI

Article

Joint Optimization of Order Allocation and Rack Selection in the "Parts-to-Picker" Picking System Considering Multiple Stations Workload Balance

Fang Wang [1], Yu Wang [1,*] and Daofang Chang [2]

[1] Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China
[2] Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China
* Correspondence: wangyu@shmtu.edu.cn

Abstract: E-commerce companies generate massive orders daily, and efficiently fulfilling them is a critical challenge. In the "parts-to-picker" order fulfillment system, the joint optimization of order allocation and rack selection is a crucial problem. Previous research has primarily focused on these two aspects separately and has yet to consider the issue of workload balancing across multiple picking stations, which can significantly impact picking efficiency. Therefore, this paper studies a joint optimization problem of order allocation and rack selection for a "parts-to-picker" order picking system with multiple picking stations to improve order picking efficiency and avoid uneven workload distribution. An integer programming model of order allocation and rack selection joint optimization is formulated to minimize the racks' total moving distance and to balance the orders allocated to each picking station. The problem is decomposed into three sub-problems: order batching, batch allocation, and rack selection, and an improved simulated annealing (SA) algorithm is designed to solve the problem. Two workload comparing operators and two random operators are developed and introduced to the SA iterations. Random instances of different scales are generated for experiments. The algorithm solutions are compared with those generated by solving the IP model directly in a commercial solver, CPLEX, and applying the first-come-first-serve strategy (FCFS), respectively. The numerical results show that the proposed algorithm can generate order allocation and rack selection solutions much more efficiently, where the moving distances of the racks are effectively reduced and the workloads are balanced among the picking stations simultaneously. The model and algorithm proposed in this paper can provide a scientific decision-making basis for e-commerce companies to improve their picking efficiency.

Keywords: "parts-to-picker" picking system; order allocation; rack selection; workload balancing; improved simulated annealing algorithm

Citation: Wang, F.; Wang, Y.; Chang, D. Joint Optimization of Order Allocation and Rack Selection in the "Parts-to-Picker" Picking System Considering Multiple Stations Workload Balance. *Systems* **2023**, *11*, 179. https://doi.org/10.3390/systems11040179

Academic Editors: Mahmoud Efatmaneshnik, Shraga Shoval and Larissa Statsenko

Received: 15 February 2023
Revised: 23 March 2023
Accepted: 27 March 2023
Published: 29 March 2023

1. Introduction

In recent years, the proliferation of internet users and the growth of e-commerce platforms have made online shopping a significant mode of consumption. In 2021, the number of online shoppers in China was estimated at 842 million. The substantial increase in users presents a challenge for e-commerce warehouses, which must process an increasing volume of orders. The efficient processing of orders and the completion of picking activities have thus become a pressing issue for companies operating in this domain [1]. Order picking refers to extracting products from storage locations in a warehouse to fulfill customer orders [2]. The traditional "picker-to-parts" picking system entails workers continuously visiting racks, incurs non-productive costs and labor expenses, and is ill-equipped to meet the growing demand for order fulfillment [3]. With the advancement of Internet of Things technology and intelligent devices, the "parts-to-picker" order picking system has emerged to address these challenges, which becomes a crucial means for e-commerce warehouses to enhance order picking efficiency [4,5].

To improve the efficiency of order processing in distribution centers, major e-commerce companies have adopted "parts-to-picker" robotic mobile fulfillment systems (RMFS) with higher degrees of automation, such as Amazon's Kiva system [6,7]. As depicted in Figure 1, RMFS consists of a series of mobile robots, movable racks, picking stations, other hardware systems, encompassing inbound docks, outbound docks, storage area, picking area, and storage and charging areas for the handling robots [8]. Goods are stored on movable racks in the warehouse and are transported by robots between picking stations and storage areas, significantly reducing the labor costs and workload of pickers and improving order picking efficiency [9].

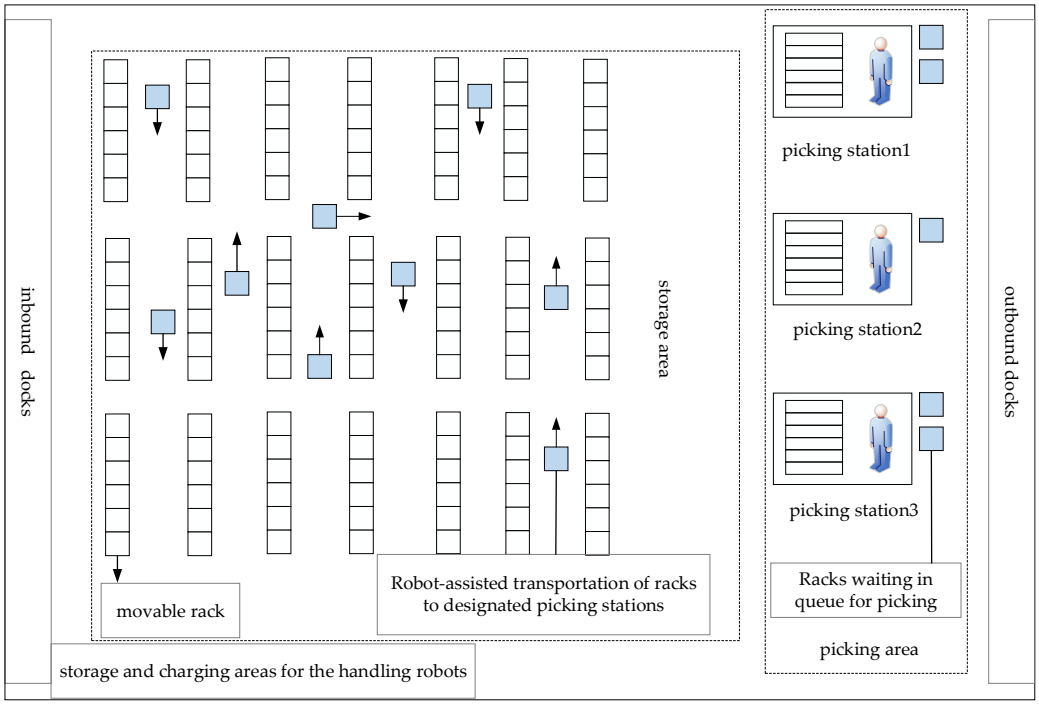

Figure 1. Warehouse layout of the robotic mobile fulfillment system.

However, the "parts-to-picker" picking system has also brought challenges. The daily order amount for large warehouses with multiple picking stations can reach tens of thousands, with various products in each order. The stations work in parallel with a shared storage strategy, i.e., the same rack can store different products, and the same product can be stored on different racks. Thus, multiple rack selection options exist for the same order set. The critical decisions include selecting the optimal combination of racks, allocating orders to picking stations, and efficiently completing all orders with the shortest rack moving distance [10,11]. For rack selection, the ordered quantity of goods and the storage capacity of goods on the racks must be matched. Moreover, since the orders are too large to handle and some may require overlapped items, the orders can be batched before rack selection to improve the working efficiency. For order allocation, since multiple racks could provide each given product, it is necessary to consider the allocation of orders among multiple stations with the decisions on rack selection simultaneously. The joint optimization of the two decisions is more challenging, and efficient algorithms are required to avoid conflicts of racks and improve efficiency.

Our literature review found a lack of research on order allocation and rack selection in "parts-to-picker" picking systems that consider the parallel operation of multiple picking stations and the workload balancing across these stations.

Therefore, we consider a joint optimization for the order allocation and rack selection problem with multiple picking stations for the "parts-to-picker" mobile robot fulfillment system, taking the workload balance among the stations into account. An integer pro-

gramming model is formulated to minimize the workload imbalance rate and the total moving distance of the selected racks. We also designed an improved simulated annealing algorithm to solve the problem efficiently. In an order-picking system with multiple picking stations, the pickers work in parallel, so workload balance has to be considered during order allocation and rack selection. If the workload is balanced, then the overall completion time of the entire warehouse system could be completed, which will cause employee dissatisfaction or even affect the picking efficiency in the long term.

The innovation and contribution of this paper can be summarized as follows:

(1) A more comprehensive mathematical programming model is developed, simultaneously addressing order batching, batch allocation, and rack selection while considering constraints on rack storage capacity and workload balancing for picking stations.

(2) The paper considers the shared storage strategy of racks in the "parts-to-picker" system and considers order-rack similarity when batching orders.

(3) We propose an enhanced simulated annealing algorithm. Specifically, we have designed two workload comparison operators and two random exchange operators based on the objective. The former serves as a targeted domain action to facilitate a faster workload balance, while the latter aims to expand the solution space. By integrating the two operators, we perform iterative optimization of the initial solution to identify optimal solutions.

The remaining parts of this paper are organized as follows: Section 2 reviews the relevant literature. Section 3 provides a detailed description of the joint optimization problem and the integer programming model. Section 4 proposes the improved simulated annealing algorithm, which first introduces the generation of initial solutions and then designs neighborhood search operators to find better solutions iteratively. Section 5 verifies the effectiveness and efficiency of the proposed model and algorithm through extensive numerical experiments. Section 6 provides further conclusions and discussions.

2. Literature Review

In the current state of research, there is limited study on the joint optimization problem of order allocation and rack selection in the "parts-to-picker" system. Most studies focused on considering each decision-making aspect separately, making it challenging to obtain optimal solutions.

The problem of rack selection in the "parts-to-picker" mobile robot fulfillment system refers to selecting a collection of racks with the shortest total transportation distance or the minimum total number of racks based on the number of product types required by a batch of orders and the number of product types stored on the racks. Furthermore, this combination of racks must satisfy the picking requirements of the batch of orders. The rack selection problem is unique to the "parts-to-picker" system and is a relatively new combinatorial optimization problem that has received relatively little research attention. Li et al. [12] studied the rack selection problem for a given batch of orders by establishing a 0–1 linear programming model, aiming to minimize the round-trip time of shelves and proving its NP-hardness. They proposed a three-stage hybrid heuristic algorithm to solve this model. Wang Zheng [13] also assumed a known set of orders to be picked and studied the best rack selection problem by establishing an integer programming model and designing a simulated annealing algorithm with six local search operators to update the initial solution. Through various example scenarios, their findings demonstrated the superiority of their algorithm. Similarly, Zhang Tingting [14] established an integer programming model to minimize the number of times shelves are moved and designed a heuristic algorithm to solve it. The study proved that rack selection in e-commerce warehousing systems directly impacts picking efficiency and cost. However, these studies only consider single-picking platforms and do not consider similar picking scenarios for multiple-picking platforms commonly found in existing warehouses.

The allocation of orders in the "parts-to-picker" mobile robot fulfillment system refers to grouping or sorting orders to maximize picking efficiency while minimizing duplicated

resources and labor. About this, Wang Shanshan [15] proposed an improved genetic algorithm to solve an order batch optimization model for a shuttle storage system to minimize the total number of container outbounds. This approach resulted in a reduced number of outtakes and improved the overall picking efficiency of the system. Li Zhenping [16] conducted a similar study of order batching in a shuttle-based "parts-to-picker" picking system and analyzed the impact of parameters such as the capacity of the picking station and the similarity weighting coefficient on the batching results. The research showed that as the capacity of the picking station increased, the number of outbound containers decreased. Boysen et al. [17] studied the order allocation and rack selection problem in a single picking station and designed a simulated annealing decomposition program to address each sub-problem. Yang et al. [18] jointly optimized order allocation and rack selection in a single picking station in a robot movement performance system. Despite this, neither study accounted for the specific quantities of products on the racks, assuming their mere presence was sufficient to fulfill orders.

Furthermore, the joint optimization problem has also been explored by various researchers. Valle and Beasley [19] addressed the combined issue of order allocation, rack allocation, and rack ordering by assigning customer orders and mobile racks to pickers and determining the rack sequence within a single picking station. Wang et al. [20] investigated the problem of order picking in a robot movement performance system with multiple picking stations, intending to reduce the number of rack movements. This problem encompasses sub-problems such as order allocation, sorting, rack selection, and scheduling. It considers the specific consideration of inter-station rack selection and the queueing effect at each picking station, proposing a two-stage hybrid heuristic algorithm. Qin et al. [21] created two-stage A* and adaptive large neighborhood search algorithms. They used these algorithms to solve the problem of order allocation and path planning in multiple picking stations. This helps in efficiently scheduling and configuring e-commerce warehousing resources and provides practical decision-making guidance for e-commerce warehousing intelligence. Zhao Jinlong et al. [1] have developed an optimized order-picking model that considers delivery deadlines. They propose improved algorithms and rules to optimize order allocation, sorting decisions, and rack access orders. Despite the research on joint optimization problems, these studies have not considered the workload balance in a multi-picking station system, which does not reflect the warehouse layout.

In summary, the existing literature, both domestic and foreign, has laid a theoretical foundation for this paper. However, our investigation has revealed that there is currently a lack of research in the joint optimization of order allocation and rack selection with workload balancing for multiple picking stations in the "parts-to-picker" picking system. Specifically, previous research has yet to address the challenge of determining order batches for picking stations and the appropriate set of racks while ensuring workload balance and minimizing the distance traveled by the racks. Thus, there is a need for further research on the joint optimization problem of order allocation and rack selection that considers workload balancing while minimizing the rack movement distance in order to improve the efficiency and workflow of the "parts-to-picker" picking system.

3. Problem Description and Formulation

3.1. Problem Description

The order picking process of the "parts-to-picker" system is shown in Figure 2 below. For the customer order set obtained from the e-commerce information system, the system first performs order allocation, including order batching and batch allocation, and establishes a correspondence between orders and picking stations. Workload balancing is essential as we consider multiple picking stations working in parallel in this paper. Given that each order has a different total number of items and thus requires different amounts of work, we evaluate the workload of a picking station based on the total number of items assigned to it. Secondly, appropriate racks are selected to meet the picking requirements of orders, and correspondence between racks and picking stations is established. Since the

racks to be picked need to be transported by robots to the corresponding picking stations and then returned to their storage locations after picking, the walking distance of the robot is a critical factor that affects the efficiency of order picking. The robot's walking path includes three parts: (1) from the current location to the target rack; (2) transporting the rack to the corresponding picking station; and (3) returning the rack to its storage location after picking. Among them, (1) is the distance traveled without a load, while (2) and (3) are the distances traveled with a load. It can be seen that the distance traveled with a load (i.e., the rack moving distance) accounts for the central part of the total distance; therefore, minimizing the rack moving distance is crucial for improving order picking efficiency [10].

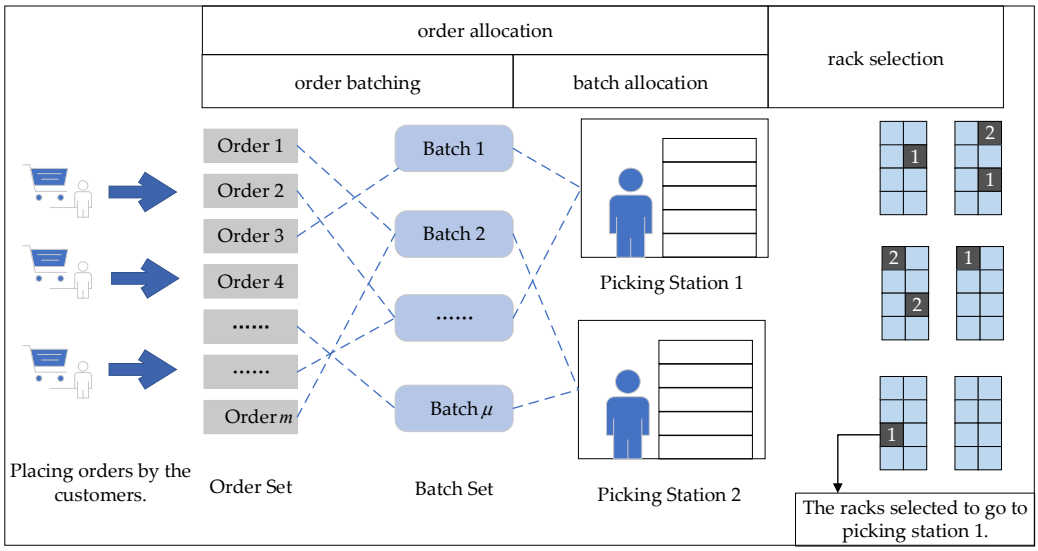

Figure 2. Illustration of order allocation and rack selection.

Furthermore, considering the limited capacity of the picking station, workers can handle up to I orders simultaneously, and the number of orders in each batch can be at most I. The problem is to batch orders, allocate them to picking stations, determine the combination of racks, and minimize the total rack moving distance while balancing the workload of each picking station.

3.2. Assumptions

In this study based on a rack-moving mobile robot picking system, the following assumptions are made:

- The location of the racks in the warehouse is fixed, and the racks return to their initial storage position after completing the picking task;
- All racks have the same specification attributes;
- All picking stations have the same capacity;
- The rack inventory can meet the order demand;
- Orders in the same batch can be processed simultaneously without considering the order or rack order within the same batch.

3.3. Integer Programming Model

The descriptions of relevant parameters and variables can be found in Tables 1 and 2.

Table 1. Description of symbols and parameters.

Parameters	Descriptions
J	Order Set, $J = \{j \mid j = 1, 2, \cdots, m\}$
R	Rack Set, $R = \{r \mid r = 1, 2, \cdots, n\}$
B	Batch Set, $B = \{b \mid b = 1, 2, \cdots, u\}$
P	Picking Station Set, $P = \{p \mid p = 1, 2, \cdots, s\}$
K	Commodity Category Set, $K = \{k \mid k = 1, 2, \cdots, t\}$
I	The maximum number of orders a batch can accommodate, which is also the maximum order capacity a picking station can handle simultaneously
W_p	The workload of picking station p
a_j	The number of SKUs contained in order j
a_{jk}	The number of SKUs k required by order j
d_{rk}	The storage capacity of SKU k on rack r
l_{rp}	The distance from rack r to picking station p
c_1	Representative unit product picking cost
c_2	Representative unit rack movement cost
q_{br}^k	The number of SKUs k provided by rack r for batch b

Table 2. Decision variables.

Decision Variables	Descriptions
x_{jb}	0–1 Variables: 0 indicates that order j is not assigned to batch b; 1 indicates that order j is assigned to batch b
y_{bp}	0–1 Variables: 0 indicates that batch b is not assigned to picking station p; 1 indicates that batch b is assigned to picking station p
z_{br}	0–1 Variables: 0 indicates that batch b does not select rack r as its provider of SKU; 1 indicates that batch b selects rack r as its provider of SKU
w_{jp}	0–1 Variables: 0 indicates that order j is not processed by picking station p; 1 indicates that order j is processed by picking station p
v_{br}^p	0–1 Variables: 0 indicates that rack r is not assigned to picking station p when it processes batch b; 1 indicates that rack r is assigned to picking station p when it processes batch b

Objective Function:

(a) Minimize the level of imbalance in the workload of the picking stations, i.e., minimize the difference between the maximum and minimum workloads of the picking stations:

$$F_1 = \min(\max W_p - \min W_p) \tag{1}$$

(b) To minimize the rack movement distance:

$$F_2 = 2\min \sum_{p=1}^{s} \sum_{b=1}^{u} \sum_{r=1}^{n} v_{br}^p l_{rp} \tag{2}$$

Since the uneven workload allocation increases the waiting time for workers, in order to reduce the waiting time, the unit product picking cost is introduced, transforming the objective function (a) into a penalty cost incurred due to the uneven workload allocation. At the same time, the unit distance movement cost is introduced to transform the objective function (b) into the rack movement cost. Additionally, two weight coefficients are introduced. Thus, the multi-objective function is transformed into a single objective function, as follows:

$$\min F = \omega_1 c_1 F_1 + \omega_2 c_2 F_2 \tag{3}$$

Constraints:

(a) Each order can only be assigned to one batch:

$$\sum_{b=1}^{u} x_{jb} = 1 \ \ \forall j \tag{4}$$

(b) The total number of orders in each batch cannot exceed its maximum capacity:

$$\sum_{j=1}^{m} x_{jb} \leq I \ \ \forall b \tag{5}$$

(c) A batch can only be assigned to one picking station:

$$\sum_{p=1}^{s} y_{bp} = 1 \ \ \forall b \tag{6}$$

(d) Each order is assigned to the picking station to which its batch is posted:

$$\omega_{jp} = x_{jb} \times y_{bp} \ \forall j \ b \ p \tag{7}$$

(e) Each batch selects the racks to go to the corresponding picking station:

$$v_{br}^{p} = y_{bp} \times z_{br} \ \forall b \ p \ r \tag{8}$$

(f) The number of products offered by the racks cannot exceed their storage quantity:

$$\sum_{b=1}^{\mu} q_{br}^{k} \leq d_{rk} \ \forall b \ r \ k \tag{9}$$

(g) The product requirements of each batch order are met:

$$\sum_{j=1}^{m} x_{jb} a_{jk} = \sum_{r=1}^{n} q_{br}^{k} \ \ \forall b \ k \tag{10}$$

(h) The range of values for the decision variables:

$$x_{jb}, y_{bp}, z_{br}, \omega_{jp}, v_{br}^{p} \in \{0, 1\} \tag{11}$$

$$q_{br}^{k} \in N \tag{12}$$

4. Improved Simulated Annealing Algorithm

The joint optimization problem of order allocation and rack selection is an NP-hard problem. This paper proposes an improved simulated annealing algorithm to rapidly and effectively solve this problem. The algorithm first generates an initial solution: an improved seed algorithm is used to batch the orders based on the order-rack similarity; a random strategy is used to allocate the batches to the picking stations; a greedy algorithm based on the rack cost–performance ratio generates the rack selection scheme. Subsequently, two types of neighborhood search operators are designed to generate new neighboring solutions: workload comparison operators and random operators. If the neighboring solution is better than the current solution, it replaces it; otherwise, it is accepted with a probability of $\exp(\frac{-\Delta f}{T})$.

4.1. Order-Rack Similarity

Order similarity is a numerical value that measures the similarity of the order's attributes, and allocating orders with high similarity to the same batch can reduce repeated

picking and improve picking efficiency [22,23]. In traditional "picker-to-parts" order picking systems, the order batching is mainly based on the similarity of order items, which refers to the ratio of the number of standard products contained in two orders to the total number of products. Orders usually contain various products in the "parts-to-picker" system considered in this paper. The same rack can store different products, and different products on the same rack can be picked simultaneously. Therefore, considering order item similarity in this system is difficult to obtain an effective order batching result. This paper proposes order-rack similarity, which is similar to order-item similarity [17] and is defined as $R_j = (R_{j1}, R_{j2}, \cdots, R_{jr})^T$.

$$R_{jr} = \begin{cases} 1, \text{ The existence of a certain product in order } j \text{ being stored on rack } r \\ 0, \text{ Otherwise} \end{cases};$$

The similarity between orders j and j' in terms of their respective racks is calculated as follows:

$$S_{jj'} = \frac{R_{jr}^T R_{j'r}}{R_{jr}^T R_{jr} + R_{j'r}^T R_{j'r} - R_{jr}^T R_{j'r}}$$

Example: Suppose four products are stored in racks $R_1 = \{a, a, a, b, b\}$, $R_2 = \{a, a, c, c\}$, $R_3 = \{b, b, b, d\}$, and $R_4 = \{b, c, c, d, d\}$, and there are three orders to be picked, $J_1 = \{b\}$, $J_2 = \{a, a\}$, and $J_3 = \{c, d\}$. Then, for $R_{1r} = (1, 0, 1, 1)^T$, $R_{2r} = (1, 1, 0, 0)^T$, and $R_{3r} = (0, 1, 1, 1)^T$, the rack similarity between order one and order two is $S_{12} = 0.25$, between order two and order three is $S_{23} = 0.25$, and between order one and order three is $S_{13} = 0.5$.

4.2. Generation of Initial Solutions

This section presents a heuristic algorithm to generate the initial solution for the improved simulated annealing algorithm, consisting of the following three parts:

Part 1: Generation of Initial Batch Result Based on the Similarity of Order Racks

Step 1: Calculate the ratio of the number of orders m and the maximum number of orders that a batch can accommodate I and round it up to $\mu = [\frac{m}{I}]$;
Step 2: Calculate the similarity S between any two orders, sort the order similarity values S in descending order, and select the first $\mu - order$ pairs with high similarity values as seed order pairs;
Step 3: Calculate the average similarity value of the remaining orders with each seed order pair and add it to the seed order pair with the maximum average similarity value until reaching the batch capacity limit or there are no orders to be added;
Step 4: Output the batch result B.

Part 2: Random Generation of Order Batch Allocation

Step 1: Calculate the picking workload for each batch $W_b = a_j x_{jb}$ based on the order batch result, wherein $a_j = \sum_{k=1}^{t} a_{jk}$.
Step 2: Sort the batches in descending order of picking workload.
Step 3: Allocate the batches to picking stations sequentially.
Step 4: Output the order batch allocation result X_0 and calculate the difference between the maximum and minimum picking station workload F_1.

Part 3: Greedy Algorithm Based on Rack Cost-effectiveness for Generating Initial Rack Selection Solution

We propose a rack cost-effectiveness method to determine which racks are more cost-effective. This method prioritizes racks with higher cost-effectiveness because each rack's moving distance is different.

Step 1: Initialize $R = (1, 2, \cdots, n)$, $Y = \varnothing$; use vector $Q_b = (a_{b1}, a_{b2}, \cdots, a_{bk})^T$ $(b = 1, 2, \cdots, u)$ to represent the vector of the demand for all goods in batch b, where a_{bk} represents the amount of goods k contained in batch b.

Step 2: Remove redundant racks. Set $b = 1$, and calculate $E = \sum_{k=1}^{t} a_{bk} d_{rk}$ for any rack $r = \{1, 2, \cdots, n\}$. If $E = 0$, remove the rack from R, effectively reducing the search space and improving solution speed; otherwise, go to step 3.

Step 3: Calculate the cost–performance of the rack. Define the cost–performance of the rack as a rate:

$$rate(r) = \frac{\sum_{k=1}^{t} q_{br}^{k}}{l_{rp}}$$

where: $q_{br}^{k} = \min\{a_{bk}, d_{rk}\}$ represents the number of products k that rack r provides for this batch, and l_{rp} represents the distance of rack r from picking station p.

Step 4: Sort the racks in descending order of cost–performance.

Step 5: Add the rack with the highest cost–performance to set Y and remove it from R, updating Q_b.

Step 6: If $Q_b = 0$, go to step 6; otherwise, go to step 2.

Step 7: Set $b = b + 1$. If $b > u$, go to step 7; otherwise, go to step 1.

Step 8: Calculation ends; output the rack set Y corresponding to each batch and calculate the distance F_2 required to move all racks for picking all batches.

4.3. Improved Simulated Annealing Algorithm

In order to search for better solutions to the problem at hand, two types of neighborhood search operators have been designed: workload comparison operators (Operators 1 and 2) and random exchange operators (Operators 3 and 4). Workload comparison operators steer the perturbation towards a balanced workload, serving as a targeted neighborhood action that leads to a faster workload balance. Random exchange operators are advantageous for expanding the solution space. Combining these two types of operators improves computational efficiency and increases the probability of obtaining the optimal global solution.

Revised Neighborhood Operation:

Operator 1: Select the order batch with the enormous workload from the picking station with the enormous workload and add it to the picking station with the minor workload, ensuring that the new solution satisfies the batch capacity constraint of the picking station.

Operator 2: Randomly select an order batch b from the picking station with the most considerable workload and exchange it with an order batch b' from the picking station with the minor workload where $W_b > W_{b'}$.

Operator 3: Randomly select an order and add it or exchange it with another batch, ensuring that the batch capacity constraint is satisfied after addition.

Operator 4: Randomly select two picking stations; select one batch from each and exchange the batches.

The specific steps of the algorithm are as follows:

Step 1: Initialization. The order allocation solution $X = X_0$ obtained in the above stage, the total cost $F = f(X)$ of the objective function, the initial temperature $T = T_0$, and the number of iterations L;

Step 2: Under the current temperature T, randomly select a neighborhood search operator to perturb the current solution, obtain a new solution X', and calculate the corresponding total cost $f(X')$;

Step 3: Calculate the incremental cost $\triangle f = f(X') - f(X)$. If $\triangle f < 0$, accept X' as the new current solution, $X = X'$, $F = f(X')$, $l = l + 1$; otherwise, accept the current solution with a probability of $\exp(\frac{-\Delta f}{T})$; that is, randomly generate a random number

rand in the interval (0, 1) if $\exp(\frac{-\Delta f}{T}) > rand$, accept X' as the new current solution, $X = X'$, $F = f(X')$, $l = l + 1$; otherwise, retain the current solution;
Step 4: Determine whether L iterations have been performed under temperature T. If $l \leq L$, go to step 2; otherwise, go to step 5;
Step 5: Set $T = \alpha T (0 < \alpha < 1)$ to reduce the temperature. If $T \leq T_{\min}$, output the current order allocation solution X, the corresponding rack set Y, and the total cost F, and end the program; otherwise, go to step 2.

5. Experimental Results and Discussions

In order to verify the validity of the proposed model and algorithm, the case in the literature [20] was adopted, and simulations were conducted using small-scale [10 × 15], medium-scale [10 × 30], and large-scale [20 × 30] warehouse grid maps as shown in Figure 3. The distances were measured in grid units, and the distance between the racks and the picking stations was their Manhattan distance. The racks, picking stations, and robots are the same size and occupy one grid, and the robot walking step length is 1. The parameters for the different-sized cases are shown in Table 3. The improved simulated annealing algorithm proposed in this paper was compared with the CPLEX solver and the commonly used first-come-first-served (FCFS) strategy in practice, respectively. The initial temperature T of the improved SA was set to 500; the decay coefficient α was set to 0.98; the number of iterations L was set to 100; and the final temperature $T_{\min}$ was set to 0.01. The results were finally compared with and without considering the balancing factor.

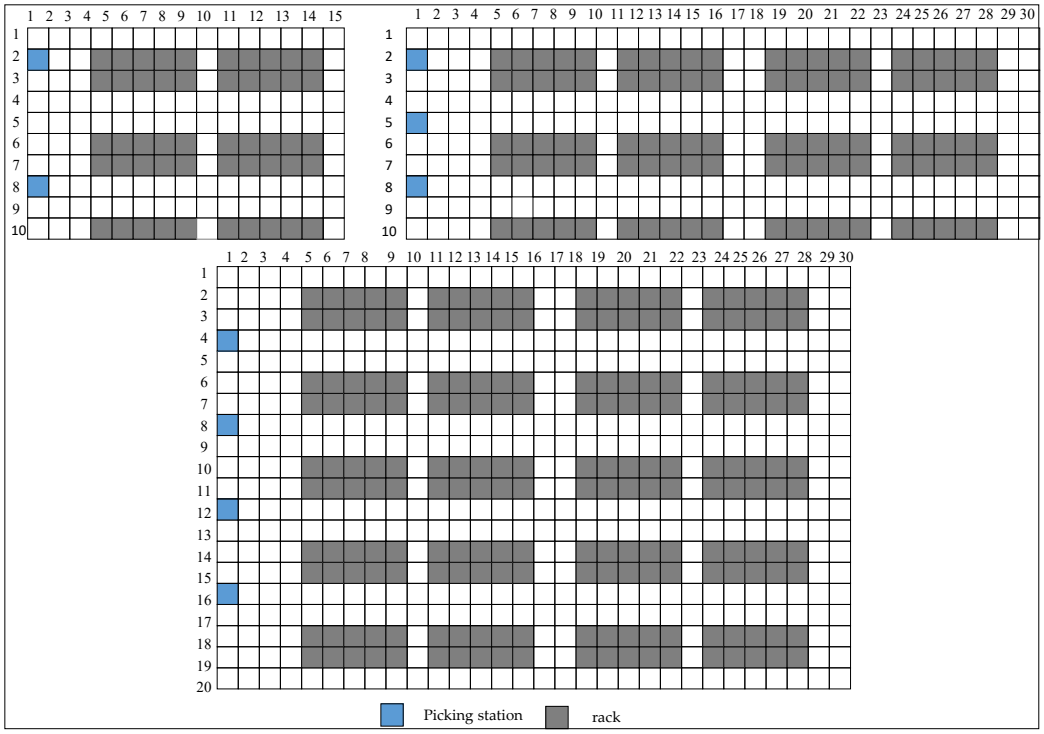

Figure 3. Small scale, medium scale, and large scale simulation warehouse grid map.

The methods were implemented using the Python programming language, and the experimental environment was Windows 10, Intel(R) Core(TM) i7-10875H CPU @ 2.30 GHz with 16.0 GB RAM.

Table 3. Setting of Different-Sized Instances.

Parameters	Small-Scale Instances	Medium-Scale Instances	Large-Scale Instances
Number of picking stations (s)	2	3	4
Number of racks (n)	50	100	200
SKUs	100	100/200	100/200
Batch capacity (W_p)	5	15	15
Unit product picking cost (c_1)		0.6	
Unit rack movement cost (c_2)		0.1	
weight coefficients		$\omega_1 = \omega_2 = 0.5$	

5.1. Comparison with CPLEX Results

The CPLEX solver and the improved simulated annealing algorithm proposed in this paper were used to solve small-scale cases. As shown in Table 4, when the number of orders is within 20, CPLEX can solve the problem within 1.1 s and find the optimal solution. However, as the number of orders increases, the computation time increases exponentially. When the number of orders is 40, the computation time reaches 7159.8 s, and CPLEX cannot solve the problem when the number of orders is 50. Compared to CPLEX's solution time, the improved simulated annealing algorithm proposed in this paper has a significant advantage, with an average computational time of 138.3 s. Compared to CPLEX's solution, the error rate of the improved simulated annealing algorithm proposed in this paper does not exceed 10% with an average error rate of 6.2%. When the number of orders is 30, the solution quality is the best with only a 1.9% error.

Table 4. Resolution Results of Small-Sized Instance.

Scale of Example		CPLEX		ISA [1]		Gap (%)
Example	Number of Orders	F	Running Time per Second (s)	F	Running Time per Second (s)	4
1	10	3.6	0.5	3.9	48.5	7.7
2	20	3.7	1.1	4.1	82.7	9.8
3	30	5.1	203.3	5.2	119.3	1.9
4	40	7.1	7159.8	7.5	147.6	5.3
5	50	-	-	10.4	201.8	-
6	60	-	-	15.2	230.1	-
Average			1841.2		138.3	6.2

[1] ISA: Improved Simulated Annealing Algorithm.

5.2. Comparison with FCFS Results

Due to difficulties in generating feasible solutions with CPLEX, our algorithm and the commonly used first-come-first-served (FCFS) strategy in the warehouse were applied to solve medium- and large-scale instances. In order to more clearly demonstrate their respective objectives for workload balancing and rack movement distance, a comparison was made between the two regarding their workload balancing level and rack distance. The results are shown in Tables 5 and 6.

Tables 5 and 6 demonstrate that the proposed algorithm significantly improves the workload balance. Compared to warehouses commonly used "first-come-first-served" allocation strategy, the proposed method reduces the rack-moving distance. With average reductions of 94.4% and 86.2%, the maximum workload imbalance can be reduced by 100%. The proposed method also results in average reductions of 47% and 47.2% in the maximum rack-moving distance, with a maximum reduction of 56.4%." When the number of racks in the warehouse increases to 200, the total number of products to 200, and the number of orders to 400, the algorithm's runtime remains acceptable.

Table 5. Resolution Results of Medium-Sized Instance.

Scale of Example			FCFS		Improved Simulated Annealing Algorithm			Gap (%)	
Example	SKUs	Number of Orders	F_1	F_2	F_1	F_2	Running Time per Second (s)	F_1 Decline Rate	F_2 Decline Rate
7	100	50	18	264	2	120	210.2	88.9	44.4
8		100	27	530	1	238	343.5	96.3	55.1
9		150	26	868	0	406	519.3	100	53.2
10		200	33	1152	3	574	666.0	91.0	50.2
11	200	50	25	396	3	268	220.8	88.0	32.3
12		100	29	916	0	460	410.4	100.0	49.8
13		150	18	1322	1	656	549.0	94.4	50.4
14		200	31	1958	1	1166	864.0	96.8	40.4
Average							472.9	94.4	47.0

Table 6. Resolution Results of Large-Sized Instance.

Scale of Example			FCFS		Improved Simulated Annealing Algorithm			Gap (%)	
Example	SKUs	Number of Orders	F_1	F_2	F_1	F_2	Running Time per Second (s)	F_1 Decline Rate/%	F_2 Decline Rate/%
15	100	100	32	448	1	226	663.0	96.9	49.6
16		150	24	778	1	386	1037.1	95.8	50.4
17		200	41	1010	1	594	1360.2	97.6	41.2
18		300	24	1390	3	846	1803.0	87.5	39.1
19		400	23	2128	2	1128	2160.0	91.3	47.0
20	200	100	23	816	4	432	767.7	82.6	47.1
21		150	22	1384	16	604	117.2	27.3	56.4
22		200	27	1836	0	888	1584.3	100.0	51.6
23		300	18	2440	0	1382	2357.7	100.0	43.4
24		400	47	3848	8	2082	2374.2	83.0	45.9
Average							1422.4	86.2	47.2

Furthermore, we observe that when the total number of Stock Keeping Units (SKUs) is 100, the running time is relatively short, and the reductions in the levels of unbalanced workload and rack distance are relatively high. On the other hand, when the total number of SKUs is 200, the running time is relatively long, and the reductions in the levels of unbalanced workload and rack distance are relatively low. The primary reason for this phenomenon is that when the total number of SKUs in the warehouse increases, the possibility of having the same SKUs in different orders decreases, which unavoidably requires more racks and results in a heavier picking workload. However, the proposed improved simulated annealing algorithm still exhibits excellent performance.

5.3. The Impact of Workload Balancing Factors

Table 7 compares six randomly selected cases from the set of instances to demonstrate the importance of workload balancing. The cases compare the picking station workload with and without considering the workload balancing objective to obtain the minimum rack moving distance. For the 6 cases, the solutions obtained by considering the workload balancing objective showed an average decrease of 92.27% in the imbalance of the picking station workload compared to the solutions obtained without considering this objective, with reductions of 83.3%, 94.1%, 100%, 98.1%, 84.2%, and 93.9%, respectively. On the other hand, the obtained rack moving distance increased by an average of 6.35%, with increases of 3.4%, 1.7%, 3.0%, 8%, 19.5%, and 2.5%, respectively. Figures 4 and 5 compare the maximum

workload difference and rack moving distance, respectively, between considering and not considering the balancing.

Table 7. The effect of balancing factors on the results.

Example	Ignoring Workload Balance		Considering Workload Balance		Comparison of Enhancements	
	F_1	F_2	F_1	F_2	F_1 Decline Rate/%	F_2 Decline Rate/%
7	6	116	1	120	83.3	−3.4
8	17	238	1	242	94.1	−1.7
9	27	394	0	406	100.0	−3.0
17	53	550	1	594	98.1	−8.0
18	19	708	3	846	84.2	−19.5
19	33	1100	2	1128	93.9	−2.5
		Average			92.3	−6.4

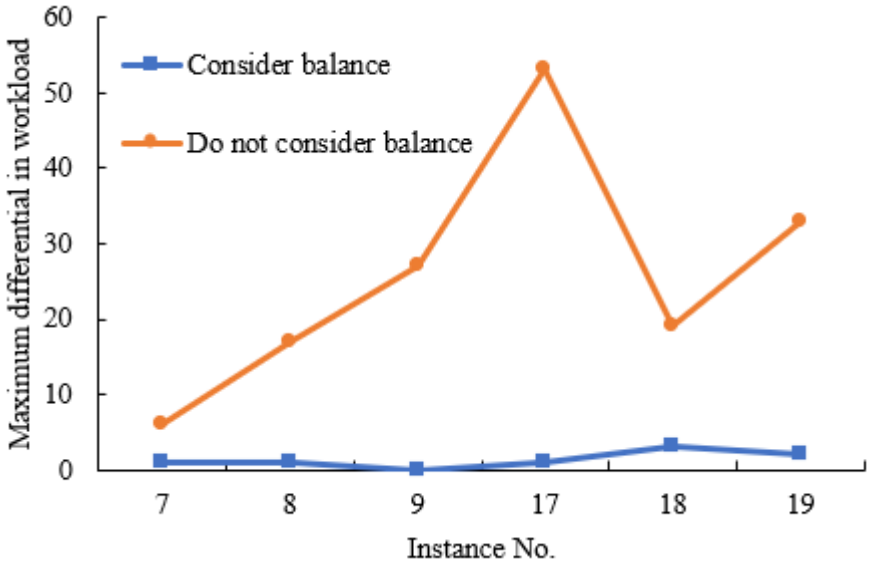

Figure 4. Comparing the maximum effort difference between considering equalization and not considering.

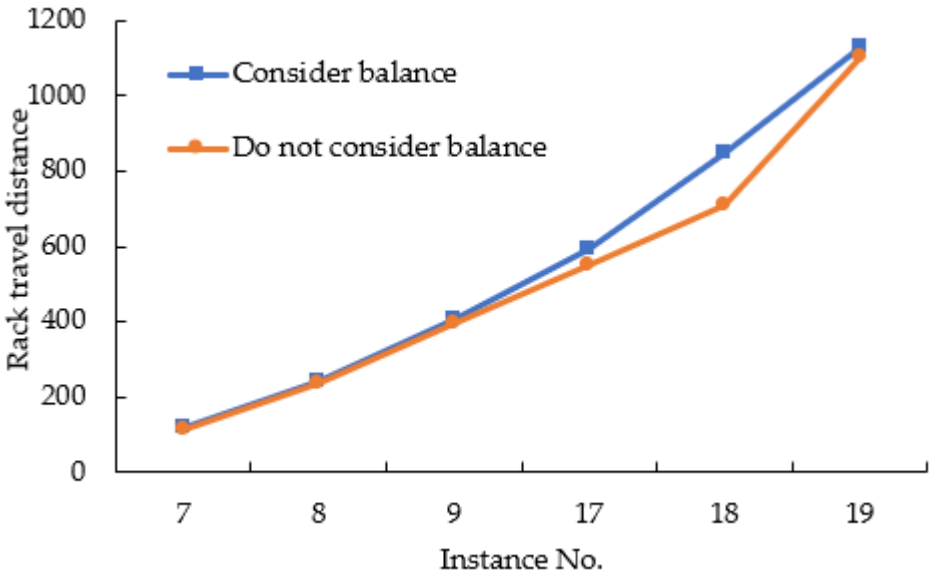

Figure 5. Comparison of rack travel distance between considering equilibrium and not considering.

Therefore, considering both factors, the improved simulated annealing algorithm proposed in this paper can effectively balance the order workload distribution while avoiding a significant increase in the rack moving distance.

5.4. Sensitivity Analysis of Weight Coefficients of Objective Functions

Due to the direct impact of weighting coefficients ω_1 and ω_2 on the solution of the simulated annealing algorithm in the objective function, this section selects a specific example from small-scale, medium-scale, and large-scale computations to conduct a sensitivity analysis of the maximum difference in workload between different picking stations and the total rack movement distance under varying ω_1 and ω_2. Figure 6 depicts the variation of the maximum workload difference as the weighting coefficients change. As the value of ω_1 increases, the workload imbalance between different sorting stations gradually becomes balanced. It can also be observed that the balancing speed is faster when $0 \leq \omega_1 \leq 0.5$ and slower when $0.5 \leq \omega_1 \leq 1$. Figure 7 illustrates the variation of the rack movement distance as the weighting coefficients change. As the value of ω_1 increases, the optimal rack movement distance gradually increases. The trends of the two objective functions are opposite and cannot be optimized simultaneously. Therefore, in practice, the weighting coefficients of the two objective functions should be determined based on specific circumstances to achieve satisfactory optimization results.

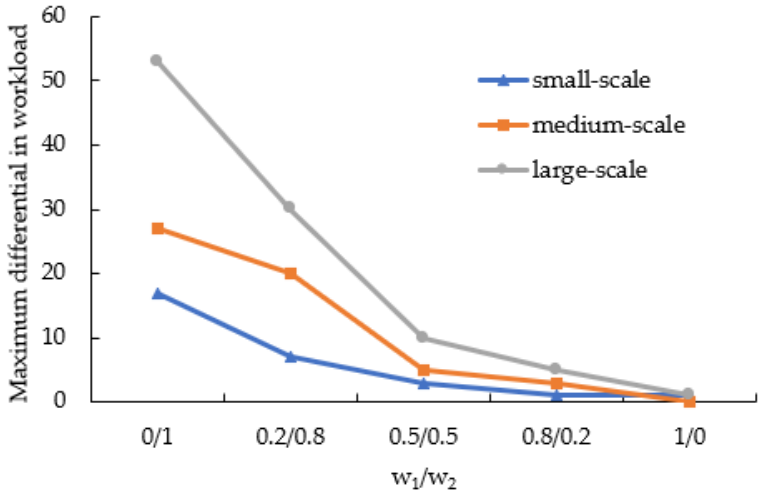

Figure 6. The relationship between the maximum workload difference and the weight coefficient.

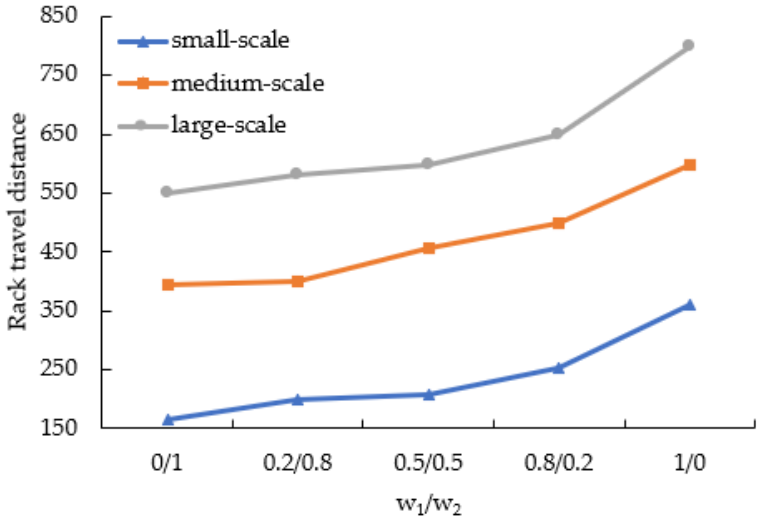

Figure 7. The relationship between the distance of the selected racks' movement and the weight coefficient.

6. Conclusions

This study explores the joint optimization of order allocation and rack selection in a "parts-to-picker" picking system, considering the picking stations' balancing factor. A mixed-integer programming model is developed to minimize workload imbalance and the distance to selected racks while addressing three sub-problems: order batching, batch allocation, and rack selection. The correspondence between multiple orders, picking stations, and racks are established in the system. An improved simulated annealing algorithm is designed, which includes two workload balancing operators and two random exchange operators to improve its efficiency. The former balances the workload quickly, while the latter helps expand the solution space. Combining the two can increase the probability of obtaining the optimal global solution, essential for multi-picking station environments where workload balancing is crucial while minimizing rack movement distance.

In addition, the study considers the characteristics of the "parts-to-picker" system when solving the simulated annealing algorithm's initial solution. A more scenario-specific order-rack similarity is defined to obtain more effective order-batching results. Furthermore, rack cost-effectiveness is proposed, prioritizing the selection of closer racks that can provide more of the required items. This approach can provide a good initial solution for the simulated annealing algorithm, enabling it to find the optimal solution quickly.

Finally, numerical experiments on different scales demonstrate the proposed algorithm's effectiveness. The algorithm can provide an effective solution for order allocation and rack selection in e-commerce warehouses, reducing labor costs, improving picking efficiency, and having significant implications for warehouse management and operation. The findings can provide practical guidance and reference for future applications.

Moreover, as e-commerce warehouses face many random events in actual scenarios, such as cancellations and urgent orders, future research can explore the joint optimization of order allocation and rack selection that considers workload balancing under uncertain conditions. Additionally, introducing more decision factors, such as path planning and robot task allocation, can make the algorithm more applicable to the actual operating scenario of the "parts-to-picker" system.

Author Contributions: Conceptualization, F.W. and Y.W.; methodology, F.W. and D.C.; investigation, Y.W. and D.C.; writing—original draft preparation, F.W.; writing—review and editing, Y.W. and D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was sponsored by Shanghai Sailing Program (21YF1416400).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, J.L.; Jiang, Z.Z.; Wan, M.C.; Zhang, C.Z. Optimization of Parts-to-picker Order Picking with Due Dates. *Chin. J. Manag. Sci.* **2023**. *accepted*. [CrossRef]
2. De Koster, R.; Le-Duc, T.; Roodbergen, K.J. Design and control of warehouse order picking: A literature review. *Eur. J. Oper. Res.* **2007**, *182*, 481–501. [CrossRef]
3. Schwerdfeger, S.; Boysen, N. Order picking along a crane-supplied pick face: The SKU switching problem. *Eur. J. Oper. Res.* **2017**, *260*, 534–545. [CrossRef]
4. Guizzo, E. Three engineers, hundreds of robots, one warehouse. *IEEE Spectr.* **2008**, *45*, 26–34. [CrossRef]
5. Lamballais, T.; Roy, D.; De Koster, M.B.M. Estimating performance in a robotic mobile fulfillment system. *Eur. J. Oper. Res.* **2017**, *256*, 976–990. [CrossRef]
6. Nigam, S.; Roy, D.; de Koster, R.; Adan, I. In 13th IMHRC Proceedings. Analysis of Class-Based Storage Strategies for the Mobile Shelf-Based Order Pick System, Cincinnati, OH, USA, 2014. Available online: https://digitalcommons.georgiasouthern.edu/pmhr_2014/19 (accessed on 14 February 2023).
7. D'Andrea, R. Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead. *IEEE Trans. Autom. Sci. Eng.* **2012**, *9*, 638–639. [CrossRef]
8. Chen, X.B.; Ma, Z.Q. Robotic Mobile Fulfillment Systems: State-of-the-art and Prospects. *Acta Autom. Sin.* **2020**, *48*, 1–20. [CrossRef]

9. Zhu, Y.Q.; Tang, S. Optimization strategy for warehousing and picking of e-commerce logistics based on improved K-Means Clustering. *Logist. Eng. Manag.* **2019**, *41*, 77–79. [CrossRef]
10. Weidinger, F.; Boysen, N.; Briskorn, D. Storage assignment with rack-moving mobile robots in KIVA warehouses. *Transp. Sci.* **2018**, *52*, 1479–1495. [CrossRef]
11. Merschformann, M.; Lamballais, T.; De Koster, M.B.M.; Suhl, L. Decision rules for robotic mobile fulfillment systems. *Oper. Res. Perspect.* **2019**, *6*, 100128. [CrossRef]
12. Li, Z.P.; Zhang, J.L.; Zhang, H.J.; Hua, G.W. Optimal selection of movable racks under cargo-to-person picking mode. *Int. J. Simul. Model.* **2017**, *16*, 145–156. [CrossRef]
13. Wang, Z.; Dan, Y.X.; Zang, X.J. Rack Selection Method for Order Picking in Mobile-Rack Warehouses. *Ind. Eng. Manag.* **2022**, *27*, 15–23. [CrossRef]
14. Zhang, T.T.; Wang, Z. Research on Rack Selection in E-commerce Warehouses Based on Heuristic Algorithm. *Ind. Control Comput.* **2022**, *35*, 31–32. [CrossRef]
15. Wang, S.S.; Zhang, J.H. Order Batch Optimization for "Part-to-Picker" Order Picking Systems. *Complex Syst. Complex. Sci.* **2022**, *19*, 74–80. [CrossRef]
16. Li, Z.P.; Han, Q.Q. Study on the Order Batching Problem of "Parts-to-Picker" Warehouse System Considering the Quantity of Items in Orders. *J. Syst. Sci. Math. Sci.* **2020**, *40*, 1456–1472. [CrossRef]
17. Boysen, N.; Briskorn, D.; Emde, S. Parts-to-picker based order processing in a rack-moving mobile robots environment. *Eur. J. Oper. Res.* **2017**, *262*, 550–562. [CrossRef]
18. Yang, X.; Hua, G.; Hu, L.; Cheng, T.C.E.; Huang, A. Joint optimization of order sequencing and rack scheduling in the robotic mobile fulfilment system. *Comput. Oper. Res.* **2021**, *135*, 105467. [CrossRef]
19. Valle, C.A.; Beasley, J.E. Order allocation, rack allocation and rack sequencing for pickers in a mobile rack environment. *Comput. Oper. Res.* **2021**, *125*, 105090. [CrossRef]
20. Wang, B.; Yang, X.; Qi, M. Order and rack sequencing in a robotic mobile fulfillment system with multiple picking stations. *Flex. Serv. Manuf. J.* **2022**, 1–39. [CrossRef]
21. Qin, J.; Yang, S.J.; Dai, B. Joint optimization of order allocation and path planning in e-commerce robotic mobile fulfillment system. *J. Railw. Sci. Eng.* **2022**. *accepted*. [CrossRef]
22. Li, J.; Huang, R.; Dai, J.B. Joint optimisation of order batching and picker routing in the online retailer's warehouse in China. *Int. J. Prod. Res.* **2017**, *55*, 447–461. [CrossRef]
23. Chen, M.C.; Wu, H.P. An association-based clustering approach to order batching considering customer demand patterns. *Omega* **2005**, *33*, 333–343. [CrossRef]