Research on Hybrid Real-Time Picking Routing Optimization Based on Multiple Picking Stations

Deyao Wang,1 Jun Jiang,2 Ran Ma,3 and Guicheng Shen3

1School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China
2Beijing Jingyibeifang Instrument Co., Ltd., Beijing 102600, China
3School of Information, Beijing Wuzi University, Beijing 101149, China

Correspondence should be addressed to Guicheng Shen; guichengshen@126.com

Received 26 February 2021; Revised 21 January 2022; Accepted 27 January 2022; Published 6 April 2022

1. Introduction

In recent years, with the booming development of e-commerce and the rapid popularity of mobile payment, online shopping has been developing at a high speed and the number of parcels has increased every year. Online shopping has grown at a quick pace in recent years, thanks to the blooming expansion of e-commerce and the rapid popularity of mobile payment. The number of deliveries has climbed every year. China’s e-commerce transactions were only 3.14 trillion yuan in 2008, but they have already risen to 31.63 billion yuan in 2018, a ten-year increase of tenfold, resulting in an explosive increase in the number of orders processed by e-commerce sorting centers per day [1].

However, as consumer demand continues to diversify and personalize, it is clear that the traditional picking center manual picking method is no longer able to control picking costs while responding quickly to customer order demands. Picking operations account for around 90% of the cost within the picking center, labor costs directly involved in picking operations account for approximately 0%, and picking time accounts for as much as 30% to 40% of the overall operating time. [2]. This shows that the real-time nature of picking operations is a key aspect of picking efficiency improvement and is crucial for e-commerce companies to enhance their core competitiveness. For example, Taobao’s “Double Eleven” and “Jingdong618’s” have raised the demand for real-time and efficient logistics sorting center solutions. Automated Guided Vehicles (AGVs) are increasingly being employed in logistics distribution centers as advanced automated logistics equipment in e-commerce, superstores, express delivery, and other industries. With the newest picking technology to manage finely orders and inventory items, a picking center can handle up to 2 million parcels in a single day and up to 4100 orders for custom clearance in one minute. The picking center’s real-time order picking efficiency places exceptionally high demands behind the massive order data.

There are currently two main types of logistics picking operation modes, namely, multi-picker synchronous picking
mode and multi-picker asynchronous picking mode. Multi-picker synchronous picking mode means that all picking stations must start at the same time for the same batch of tasks, and the next batch of tasks can only be performed when all picking stations have completed the current batch of tasks. Multi-picker asynchronous picking mode means that all logistics AGVs are grouped together, with each group serving one picker, and each picker can complete its work independently. It can be seen that the multi-picker synchronous picking mode has the problem of wasted picker resources waiting due to synchronous processing, while the multi-picker asynchronous picking mode has the problem of uneven task allocation of AGV resources due to the global grouping of AGVs.

Therefore, this paper proposes a hybrid picking mode for the problem of uneven task allocation between synchronous picking waiting time consumption and asynchronous picking, i.e., a real-time picking operation mode according to the order reach time and the picking table idle rate, which solves the problem of time wastage caused by waiting in synchronous picking and the problem of uneven task allocation for nonlocal AGVs in asynchronous.

2. Literature Review

In essence, the picking path optimization problem with multiple picking stations studied in this paper belongs to the special vehicle path optimization problem (Vehicle Routing Problem, VRP). In 1959, the vehicle path problem (VRP) was proposed by Dantzig [3]. The traveling salesman problem (TSP) in vehicle routing Problem (VRP) was proved by Gaery [4] in his paper to be an NP-hard problem, so VRP problem is also an NP-hard problem. The NP-hard problem is not only of great significance in theoretical research but also of great significance in real life, as evidenced by the vehicle path optimization problem.

In recent years, Praitk [5] has studied the effect of shelf quantity and shelf length on picking efficiency and proposed a picking equipment event model. Chan and Chan [6] investigated the effects of warehouse storage allocation strategy, picking path strategy, and picking density on picking efficiency. Lu et al. [7] focused on the implementation process of the order batch picking method (GASM) and the basic genetic algorithm picking method (GABM), and compared and analyzed the S-type picking path strategy and the midpoint slewing picking path strategy. Zhi and Ailan [8] suggested that the order picking strategy and path optimization were studied according to the distribution of goods in the order in the warehouse. And, Yi [9] proposed to establish an order management information system and strengthen the picking operation and the docking of the overbank distribution system and other strategies to optimize the picking operation with words. Caunhye et al. [10] studied the path optimization model in emergency logistics and analyzed the methods of path optimization in terms of the distribution situation of rescue supplies, the location of facilities, and casualty transportation, respectively.

Some scholars have considered both order batching and picking path optimization together, in order to improve the efficiency of picking operations. Scholz et al. [11] proposed a variable neighborhood descent algorithm capable of handling large amounts of data by considering order binning, order assignment, batch sorting, and picking paths, and a new heuristic initial solution construction method was designed in order to speed up the solution over the vehicle. Chen et al. [12] combined the three problems of order batching, path optimization, and batch sorting to design a mixed integer programming model, and used a hybrid coded genetic algorithm and an ant colony optimization algorithm to solve the problem, and the experimental results proved that the solution speed and solution quality of the algorithm were due to the multi-objective genetic algorithm and the due date optimization algorithm. Matusiak et al. [13] received inspiration from a large shopping mall and used simulated annealing algorithm and A* algorithm to solve the order batching and path planning problems.

In today’s era of rapid e-commerce development, how to respond quickly to customer needs is becoming a core issue for e-commerce and logistics companies. In order to improve the effectiveness of real-time picking, Amazon’s “KIVA” system and Jingdong’s “Tianwoldi Wolf System” have been born one after another. Various logistics companies in the industry have carried out innovation and exploration in picking system in picking route optimization. By combining the coordinate points of the storage area in the warehouse, the link path, using the ant colony algorithm and other intelligent algorithms to calculate the shortest picking path from the main channel to the picking table, improve the picking efficiency, and gradually improve the efficiency of real-time picking.

Using neural network to solve the problem of path optimization can be traced back to 1985. Hopfield [14] used Hopfield networks for solving small CVRP instances. In 2015, Vinyals et al. [15] proposed a Pointer Network (PN) based on Recurrent Neural Network (RNN), which made a qualitative leap in the study of combinatorial optimization problems. Kool et al. [16] drew on the transformer model for a variety of path optimization and proposed a transformer model with rollout baseline to remove position embedding for multiple path optimization problems, which was applied to the VRP example. Ibrahim et al. [17] developed a reinforcement learning technique to find the optimal path from the parking lot to the customer base. Zhang and Yin [18] studied the optimization of supply paths in emergency situations and solved the supply routes based on a hybrid particle swarm algorithm under the premise of obtaining emergency orders. Sun et al. [19] established a mathematical model with capacity constraints and optimized the picking path based on genetic algorithm for a dual-zone distribution center.

There are also some scholars to decompose the multi-pair multi-path optimization problem (MDVRP). David Pisinger and Ropke [20] proposed a heuristic algorithm for solving single-center VRP and applied it to solving MDVRP. Yu et al. [21] firstly simplified MDVRPTW into multiple VRPTW by clustering heuristic classification algorithm, and then used ant colony algorithm to solve each VRPTW. Jiali and Xiuping [22] designed a “two-stage adaptive genetic algorithm” based on multiple scanning operations and stage
3.1. Problem Description of the Hybrid Picking Model.

An intelligent storage system is known to have multiple picking stations and a certain number of goods picking operations; each picking station can individually initiate picking tasks in real time, the system order portfolio arriving at the picking station is divided by order items, the AGV can visit multiple shelves of bays at a time to pick the goods required by the order according to the items and deliver these goods to the picking station. The shelves are stationary, and each shelf includes vertical levels. The AGVs are located in a uniform buffer zone when idle, have their own storage shelves and are capacity-constrained, so they can pick multiple items at once (e.g., the AGV TURU). When an order comes in, the AGV immediately picks the goods at the goods level according to the order requirements and sends them back to the picking table for sorting. The requirement is to design a reasonable picking path from the current scenario based on the order requirements and the location of the goods, and to efficiently complete the real-time picking task. A schematic of the picking path with capacity constraints in mixed picking mode is shown as Figure 1.

The actual problem is abstracted and the following assumptions are made.

1. When an order comes in, priority is given to the picking table with a high degree of availability and a short distance from the center of gravity of the order item for picking.
2. All AGVs are in a unified buffer zone when they are idle; when an order comes, all AGVs will start from the same starting point and return to the starting point after completing the picking task.
3. AGVs depart from the picking table to each goods node for one-way pickup (one round), and an order can be completed by multiple rounds (AGVs) together.
4. The AGV comes with its own shelf and has the capacity to be idle; when the pickup is carried out, the current pickup round ends immediately when the AGV’s shelf is full.
5. The AGVs can go back and forth between the goods cargo and the picking table many times, and each round of the task accepts only one picking service from one AGV, and each time it departs to the goods node, it needs to meet the demand of the current goods picking at one time.
6. Once an order arrives, the AGV is immediately assigned to pick up the goods.

3.2. Matching Method of Idle Picking Stations and Orders

3.2.1. Calculating the Idle Degree of Picking Table. First, the idle degree of picking stations is calculated, and based on the idle degree of idle picking stations, picking orders are prioritized to the picking station with the highest idle degree. Idle degree \( t_{omp} \) The formula is shown in (1).

\[
\begin{align*}
\text{idlem} = \frac{t_{omp}}{t},
\end{align*}
\]

where \( t_{omp} \) denotes the idle time \( \text{idlem} \) of idle picking stations, \( t \) is the total working time, and \( M \) is the total number of picking stations in the idle state, where \( m \in \{1, 2, \ldots, M\} \).

3.2.2. Calculation of the Distance between the Center of Gravity of the Order Item and the Picking Table. Next, the distance between the center of the order item and the picking table is calculated with the following formula.

\[
\begin{align*}
\mathbf{x}_n = \frac{\sum_{i=1}^L x_{ni}}{L}, \quad \mathbf{y}_n = \frac{\sum_{i=1}^L y_{ni}}{L},
\end{align*}
\]

where \( x_{ni}, y_{ni} \) denote the \( x \)-axis coordinates and \( y \)-axis coordinates of the goods position where the \( i \)-th item order to be picked in the item order is located, and \( \mathbf{x}_n, \mathbf{y}_n \) is the center of the \( n \)-th item order, \( L \) is the number of spaces used for the goods in the order, and \( V_{xmn}, V_{ymn} \) are the horizontal and vertical coordinates of the free picking table.

3.3. CVRP Mathematical Model for Hybrid Real-Time Picking Mode. To facilitate model construction and subsequent calculations, the relevant variables are assumed as 3.3.1.

3.3.1. Symbol Description

- \( A = (V, E) \): Picking network, where \( V = \{v_0, v_1, v_2, \ldots, v_L\} \) represents the set of nodes and \( v_0 \) represents the picking station
- \( K \): the number of AGVs required for the current picking order, where \( k \in \{1, 2, \ldots, K\} \)
- \( G \): the total number of AGVs
Q: maximum load capacity of the AGV
q_j: the loading capacity required by cargo j
c_{ij}: the picking cost from slot i to slot j
w_{ijk}: the AGV k remaining loading capacity from cargo i to cargo j

(2) Decision-making variables.

\[ x_{ijk} = \begin{cases} 
1, & \text{the } k\text{'th AGV will travel from cargo } i \text{ to cargo } j, \\
0, & \text{otherwise}, 
\end{cases} \quad (3) \]

\[ y_{ik} = \begin{cases} 
1, & \text{cargo } i \text{ will be picked by the } k\text{'th AGV}, \\
0, & \text{otherwise}. 
\end{cases} \]

\[ K \leq G, \quad (5) \]

\[ \sum_{i=1}^{L} q_i y_{ik} \leq Q, \quad \forall k \in \{1, 2, \ldots, K\}, \quad (6) \]

\[ \sum_{k=1}^{K} \sum_{i=1}^{L} q_i y_{ik} \leq KQ, \quad (7) \]

\[ \sum_{k=1}^{K} y_{ik} = 1, \quad \forall i \in \{1, 2, \ldots, L\}, \quad (8) \]

\[ \sum_{j=1}^{L} x_{ijk} = y_{ik}, \quad \forall i \in \{1, 2, \ldots, L\}, \forall k \in \{1, 2, \ldots, K\}, \quad (9) \]

\[ w_{0jk} = x_{0jk}Q, \quad \forall j \in \{1, 2, \ldots, L\}, \forall k \in \{1, 2, \ldots, K\}, \quad (10) \]

\[ \sum_{k=1}^{K} x_{ijk}(w_{ijk} - q_j) \geq 0, \quad \forall i \in \{1, 2, \ldots, L\}, \forall k \in \{1, 2, \ldots, K\}, \quad (11) \]

\[ \sum_{i,j \in S \times S} x_{ijk} \leq |S| - 1, \quad S \subset \{1, \ldots, L\}, S \neq \emptyset, k \in \{1, 2, \ldots, K\}. \quad (12) \]

### 3.3.2. Objective Function

\[ \min H = \sum_{k=1}^{K} \sum_{i=0}^{L} \sum_{j=0}^{L} x_{ijk} c_{ij}. \quad (4) \]

### 3.3.3. Constraints

The objective function (4) indicates that the sum of the travel distances of all AGVs is the shortest; equation (5) indicates that the total number of AGVs required for order picking is less than the total number of AGVs; equation (6) indicates that the amount of goods picked by each AGV does not exceed its capacity limit; equation (7) indicates that the amount of goods required for this order picking does not exceed the sum of the capacity limits of all AGVs; equation (8) indicates that each goods position is guaranteed to be visited by only one AGV; equation (9) indicates that each cargo space is allowed to be visited only once in the same picking round; equation (10) indicates that each AGV is empty from the picking table; equation (11) indicates that the AGV has sufficient loading capacity for any cargo space before the AGV picks that space; equation (12) indicates the elimination of subloops.

### 4. Algorithm Design for Hybrid Picking Mode

In previous reinforcement learning algorithms, it is more effective to solve purely theoretical routing problems. However, in this paper, starting from a realistic picking scenario, the shortest picking path solution is sought while considering the two prerequisites of “the idleness of the picking table” and “the distance between the center of gravity of the order item and the picking table.” The algorithm design of the hybrid picking mode in this paper refers to Wouter’s transformer model [16], and on this basis, the initial screening of the picking table and the order length alignment processing are added, which means that the limitation of the order volume at the moment of order arrival can be broken, so that the resources of the picking table can be more fully utilized and
the picking efficiency of the intelligent AGV can be improved. The following is a detailed description of the algorithm.

4.1. Model Construction of Hybrid Picking Model. In order to solve the hybrid picking path optimization problem, this paper improves the transformer model. The specific idea is: using the method of placeholder expansion of picking nodes, a new placeholder encoding mechanism for aligning the order length of item order groups is proposed, so that it is transformed into an array of equal length, thus allowing the input data in the input batch to be solved for different numbers of picking positions; subsequently, in the encoding process, the input data are mapped into a high-dimensional space to obtain vector features. What’s more, in the encoding process, the input data are mapped into a high-dimensional space to obtain vector features, and then, in the decoding process, the solution model is constructed based on Markov Decision Process (MDP), and the algorithm model is shown in Figure 2.

4.1.1. Coding Model for Hybrid Picking

(1) Target Picking Table Selection

(A) Calculate the picking table idle rate

Based on the work duration of each picking table in the picking center during the day, the idle degree of each picking table in the idle state is calculated as shown in equation (1), and the idle rate is sorted from highest to lowest, and the picking table with the highest idle rate is selected for picking.

\[
A = \begin{bmatrix}
[y_{00}] & [y_{01}] & [y_{02}] & \ldots & [y_{0n}] \\
[y_{10}] & [y_{11}] & [y_{12}] & \ldots & [y_{1n}] \\
[y_{20}] & [y_{21}] & [y_{22}] & \ldots & [y_{2n}] \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
[y_{M0}] & [y_{M1}] & [y_{M2}] & \ldots & [y_{Mn_{M-1}}] \\
[y_{Mn_{M-1}}] & [y_{Mn_{M-2}}] & [y_{MN}] & \ldots & \end{bmatrix}_{M \times N}
\]

where \( M \) is the total number of batches, \( N \) is the length of the longest number of picks required in the batch order, \( v_{ij} \) is the demand for picking spaces in each order, \( n_i \) is the length of the number of picks required for each order, and \( v \) is the placeholder.

(B) Calculate the distance between the target picking table and the order center of gravity of the item to be picked

According to the warehouse management system and the warehouse control system, the order items are automatically divided according to the order commodity labels. In order to minimize the walking distance of the logistics AGV, the distance between the target picking table and the center of gravity of the order group to be picked for each item is calculated as shown in equation (2), and the picking table for entering the picking work is selected based on the principle of the shortest distance.

(2) Item Order Processing. During e-commerce promotions (e.g.,”Double 11” and “618”), the number of real-time orders arriving at the picking center management system increases sharply for a short period of time. To address the irregular nature of the real-time order quantity, this paper adds to the encoding structure of the transformer and proposes a new placeholder encoding mechanism for aligning the order lengths of item order groups into arrays of equal length to address the traditional reinforcement learning model in which the key to parallel computation is based on the input of the regularized batch tensor dimension. The principle is to determine the longest length of the required number of picks in the irregular batch order array, fill the placeholders for the blank, idle picking stations between the required picking orders and the longest length, and form structured data after tensor transformation, as shown in equation (13).

(3) Data Embedding Processing. First of all input the co-ordinates of the goods locations \( = [(x_0, y_0), (x_1, y_1), \ldots] \) and the demand for each good position in the order \( D = [d_1, d_2, \ldots] \) to the coding model. Then, the coordinates of the demand point \( (x_i, y_i) \) and the demand \( d_i \) are stitched together to obtain \( p_i \) (where \( P \) is a three-dimensional data \( (x, y, d) \) ) and the high-dimensional linear projection of \( p_i \). The feature vector of picking points is obtained by embedding \( h_i^0 \). The feature vectors of the neural network at the \( i \)th picking station are shown in Figure 3, where \( o \) represents the layer of the neural network and \( i \) indicates the picking points. Briefly speaking, this vector represents the feature vector of the neural network of layer 0 in the \( i \)th picking table.

(4) Picking Network Feature Extraction

(A) Attention mechanism

In order to calculate the features of the network graph between picking points and picking points and between picking points and picking tables, the features, \( h_i^0 \). The picking network feature vector is calculated by performing multiple high-dimensional mappings, \( h_i^N \). First, by using the model parameters \( W^Q, W^K, W^V \) of the matrix operation, the feature information \( q, k, v \) of each picking node is calculated. Subsequently, the \( q \) with \( k \) correlation calculation is
performed to obtain the weights. Similarly, we can calculate the picking node \(i\) with other picking nodes \(j\) correlation weights between \(u_{ij}\).

Next, the weights are normalized. We use the softmax function to normalize the correlation \(u_{ij}\) to normalize and calculate the attention weights \(a_{ij} \in (0, 1)\), which is the correlation coefficients received by the picked nodes.

Finally, the normalized weights are used to sum up with the weight of \(V\). Based on the \(a_{ij}\) and \(v_j\), calculate the current \(q\). The attention value under the value \(h'_i\) represents the relevance degree information between the current picking node and other picking nodes.

To facilitate the understanding of the attention mechanism, the detailed steps of the attention mechanism are shown in Algorithm 1, where the model parameters are presented above.

(B) Multi-headed attention mechanism

The multi-headed attention mechanism is an algorithm for weighted messaging between cargo locations, and consists of multiple attention mechanisms that are stitched together to calculate the correlation information between picking points based on different dimensions to obtain the picking network feature vector \(h_g^N\).

4.1.2. Decoding Model for Hybrid Picking

(1) Picking Node Relevance Embedding. First, the picking point features are obtained by coding \(h_i^N = \{h_i^N, h_i^N, \ldots\}\) and picking network diagram features \(h_g^N\), and the coordinates and demand of the goods and picking stations as input for decoding. The information of the picking table and the remaining loading capacity of the AGV are marked to make a start for solving the path solution.

(2) Attention Mechanism. Like the attention mechanism for encoding, the attention mechanism in this stage calculates the direct correlation between the last picking node in the identified path order, i.e., the picking point about to depart, and the outstanding picking point to calculate the individual probabilities of the next picking node and determine the next picking point to go to.

(3) Masking Mechanism. The masking mechanism is used to mask the next round of unsuitable cargo bits after the probability of the next visit to the node of interest is generated using the attention mechanism. And, the parameters of the masking process are set as shown in Table 1.

In the masking process, \(V_{\text{cd}}\) uses 0 and 1 to represent accessible nodes, where the 1 represents two different meanings: (1) the nodes that have been accessed and (2) the nodes that are not suitable for access in this sub-loop, i.e., after accessing the point in the sub-loop, the cargo nodes will exceed the car capacity and do not meet the realistic constraints but have not yet been picked. The picking path can then be indexed by the sequence of the mask. The masking process is shown in Figure 4 and the whole decoding process is shown in Figure 5.

4.2. Training Optimization Components for Hybrid Picking Patterns. Here, we choose a reinforcement learning algorithm with a rollout baseline [22] as the evaluation parameter to train the model, which is a hybrid algorithm combining value-based and policy gradient-based algorithms with two neural network models built in, i.e., the transformer network model and the rollout network model.

4.2.1. Reinforcement of the Learning Network Model. In a realistic picking center environment, it is generally a grid layout that presents a north-south or east-west orientation, thus using the Manhattan distance to design the objective function, see equation (14).
In this paper, the training network uses the above encoding and decoding model, i.e., the transformer model, with a $\theta$-parameterized stochastic policy $\pi$, which generates the conditional probability distributions of the nodes according to the attention mechanism, and sequentially generates the solution sequence points, see equation (14). Based on the generated solution sequence points, the loss function is calculated, see the definition of equation (15).

\[ r(\pi|s) = \sum_{t=1}^{n} |x_{\pi(t)} - x_{\pi(t-1)}| + |x_{\pi(n)} - x_{\pi(1)}| + \sum_{t=1}^{n} |y_{\pi(t)} - y_{\pi(t-1)}| + |y_{\pi(n)} - y_{\pi(1)}|. \quad (14) \]
wherein $I_t^{BL}(X^{n}_0; \sigma)$ is described in detail in the rollout, and is the time step of decoding.

\begin{equation}
\sum_{i=t}^{T} L_t^{n_i}(\pi_t) - L_{t}^{BL}(X^{n}_0;\sigma) \geq 0.
\end{equation} (15)

Based on the loss function, the gradient optimization parameters of the objective function are calculated, which are defined as shown in equation (16).

\begin{equation}
\nabla \theta - 1 \sum_{n=1}^{N} \left( \sum_{i=t}^{T} L_t^{n_i}(\pi_t) - L_{t}^{BL}(X^{n}_0;\sigma) \right) \nabla \log P(Y^{n}|X^{n}_0).
\end{equation} (16)

Finally, the traditional stochastic gradient descent (SGD) process is replaced by the Adam first-order optimization algorithm, which backpropagates the gradients and updates the neural network weights iteratively using the training data, see the defined equation (17).

\begin{equation}
\theta \leftarrow \text{adam}(\theta, \nabla \theta).
\end{equation} (17)

4.2.2. Reinforcing the Learning Baseline Model. Inputs: picking table and shelf locations $s = (x, y)$, the number of picking orders demanded is $d$, the number of AGVs is $k$, the loading capacity of the AGV is $D$.

Training: calculated according to the Adam optimization algorithm $\theta$, update back propagation gradient, optimize transformer model parameters and update rollout, evaluate the network parameters, the detailed steps are shown in Algorithm 2.

Step (1) evaluate the network model parameters with 0 initializing transformer and rollout; step (2) with step (3) is to select the best picking station; step (4) iterative module; steps (5)–(7) randomly initialize the network strategy; step (9) perform batch training; step (10) start decoding; steps (11)–(15) generate decoding sequence, steps (16)–(29) calculate update gradient values and optimize the neural network parameters based on Adam’s algorithm. Steps (21)–(23) whether to update the evaluation function parameters.

5. Example Simulation and Analysis

5.1. Example Simulation of Hybrid Real-Time Picking Mode. A picking center is arranged in a regional layout according to commodity items. There are multiple picking stations in the center. The AGVs are in a unified charging cache area, and they work together to complete picking. The intelligent warehouse is set to have 5 picking stations and 360 goods slots, and the demand for each task in each batch of order is less than 1. The quantity of AGV is not limited, which is enough to meet the
Figure 4: Masking process.

Figure 5: Attention-based decoder.
and the computer configuration used was: Intel(R) Core(TM) i5-6300U CPU @2.40 GHz, 2496Mhz, 22 cores, 41 logical processor, 4GB of RAM, and a 64 bit of Windows 10. Operating system.

To analyze the effect of the proposed model, we choose three algorithms such as OR_ARC, OR_CHR, and PATH_CHEAPEST_ARC (PATH-ARC) from Google OR-Tools to perform the algorithm comparison analysis of length dimension and time dimension on the public dataset (Augerat standard dataset set A). The datasets can be found at the URL https://neo.lcci.uma.es/vrp/vrpinstances/capacitated-vrp-instances/download. Firstly, the advantages and disadvantages of the algorithm are evaluated from three aspects: the optimal distance dissociation of the example (best), the distance solution obtained by the algorithm model (min), and the error between the experimental solution and the optimal solution of the example (DEV%) (DEV = (min best)/best). The specific experimental results are shown in Table 4.

Subsequently, in order to summarize the overall evaluation of each algorithm in the 22 group instances, four comparative analyses were performed in terms of the mean length of the algorithms, the total time, the mean DEV, and the standard deviation of the DEV, as shown in Table 5.

As can be seen from Table 5, compared with the three heuristics of ortools, PRL algorithm proposed in this paper is the closest to the exact solution from the perspective of distance and length. And, from the time dimension analysis, PRL algorithm has obvious advantages of parallel algorithm, shortens the time, and can be solved quickly.

### Algorithm 2: Reinforcement learning algorithms under mixed models.

| Inputs: | coordinates of picking table p, position s, and the demand for the picking table D |
| Output: | The pickled sequential cargo position is the solution π |
| (1) | Initialize transformer and rollout evaluation network model parameters. Initialize optimizer parameters and initialize learning rate |
| (2) | Calculation of picking table idleness |
| (3) | Calculation of the distance between the pingle order and the center of gravity of the picking table |
| (4) | For i = 1: E do- |
| (5) | Initialization parameters. θ, θBL ← θ |
| (6) | si ← random (1234) vi ∈ [1, . . . , Batch] |
| (7) | πi ← rollout (si, pi) vi ∈ [1, . . . , Batch] |
| (8) | ΠBL ← rollout (sL, pL) vi ∈ [1, . . . , Batch] |
| (9) | For n = 1, . . . , N do |
| (10) | Initialization step time. t ← 0 |
| (11) | Repeat |
| (12) | According to the probability distribution \( p(y^n_i | Y^n, X^n) \), select sequence points \( y^n_i \) |
| (13) | Observing the new state \( X^n_{i+1} \) |
| (14) | \( t ← t + 1 \) |
| (15) | Until the termination conditions are met |
| (16) | Calculate the objective function of the training \( L(\pi_i) = L(Y^n, X^n) \) |
| (17) | Calculate the evaluation value of the rollout baseline \( L(\pi_{BL}) \) |
| (18) | To find the gradient for the transformer model network. \( \nabla \theta \rightarrow (1/N) \sum_{i=1}^{N} \left( \sum_{i=1}^{T} L_i(\pi_i; \sigma) - L_i^{BL}(X^n_i; o) \right) \nabla P(Y^n | X^n) \) |
| (19) | Gradient updates to the transformer model network. \( \theta \rightarrow \text{Adam}(\theta, \nabla \theta) \) |
| (20) | End for |
| (21) | If one-sided paired t-test \( < \alpha \) then |
| (22) | \( \theta_{BL} ← \theta \) |
| (23) | End if |
| (24) | End for |

5.2 Algorithm Comparison. The experiments were implemented using Python 3.6, pycharm2019.1.3 of pytorch0.4.1, and the computer configuration used was: Intel(R)
Figure 6: Coordinate assignment chart.

Table 2: Picking table idleness and item order center of gravity distance.

<table>
<thead>
<tr>
<th>Picking table</th>
<th>Location</th>
<th>Vacancy rate</th>
<th>Distance from the center of gravity of the first group of item orders</th>
<th>Distance to the center of gravity of the second group of item orders</th>
<th>Distance to the center of gravity of the third group of item orders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Picking table1</td>
<td>(6, 17)</td>
<td>0.6</td>
<td>9.75</td>
<td>17.22</td>
<td>31.77</td>
</tr>
<tr>
<td>Picking table2</td>
<td>(14, 17)</td>
<td>0.3</td>
<td>13.39</td>
<td>11.35</td>
<td>24.42</td>
</tr>
<tr>
<td>Picking table3</td>
<td>(22, 17)</td>
<td>0.57</td>
<td>19.79</td>
<td>9.46</td>
<td>17.65</td>
</tr>
<tr>
<td>Picking table4</td>
<td>(30, 17)</td>
<td>0.4</td>
<td>27.05</td>
<td>13.34</td>
<td>12.44</td>
</tr>
<tr>
<td>Picking table5</td>
<td>(38, 17)</td>
<td>0.5</td>
<td>34.64</td>
<td>19.86</td>
<td>11.22</td>
</tr>
</tbody>
</table>

Table 3: Original data for picking nodes.

<table>
<thead>
<tr>
<th>Num</th>
<th>Picking stations 1</th>
<th>Picking stations 3</th>
<th>Picking stations 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Coordinates</td>
<td>Demand</td>
<td>Coordinates</td>
</tr>
<tr>
<td>1</td>
<td>(1, 9)</td>
<td>0.13</td>
<td>(27, 12)</td>
</tr>
<tr>
<td>2</td>
<td>(4, 14)</td>
<td>0.03</td>
<td>(19, 13)</td>
</tr>
<tr>
<td>3</td>
<td>(9, 11)</td>
<td>0.17</td>
<td>(22, 8)</td>
</tr>
<tr>
<td>4</td>
<td>(3, 11)</td>
<td>0.20</td>
<td>(15, 12)</td>
</tr>
<tr>
<td>5</td>
<td>(13, 8)</td>
<td>0.10</td>
<td>(25, 2)</td>
</tr>
<tr>
<td>6</td>
<td>(9, 0)</td>
<td>0.23</td>
<td>(21, 0)</td>
</tr>
<tr>
<td>7</td>
<td>(6, 11)</td>
<td>0.13</td>
<td>(27, 5)</td>
</tr>
<tr>
<td>8</td>
<td>(3, 13)</td>
<td>0.03</td>
<td>(19, 6)</td>
</tr>
<tr>
<td>9</td>
<td>(10, 11)</td>
<td>0.27</td>
<td>(16, 14)</td>
</tr>
<tr>
<td>10</td>
<td>(0, 8)</td>
<td>0.27</td>
<td>(27, 3)</td>
</tr>
<tr>
<td>11</td>
<td>(10, 13)</td>
<td>0.40</td>
<td>(21, 5)</td>
</tr>
<tr>
<td>12</td>
<td>(3, 5)</td>
<td>0.02</td>
<td>(24, 11)</td>
</tr>
<tr>
<td>13</td>
<td>(0, 6)</td>
<td>0.03</td>
<td>(25, 5)</td>
</tr>
<tr>
<td>14</td>
<td>—</td>
<td>—</td>
<td>(19, 1)</td>
</tr>
<tr>
<td>15</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Then, this paper conducted significance tests for best solution and RL, OR_CHR, PATHARC, and OR_ARC, respectively, as shown in Table 6.

From Table 6, it can be seen that the significance of the RL method and OR_ARC method is less than 0.05 and passes the level of the significance test, while the significance of the OR_CHR method and PATHARC method is bigger than 0.05 and does not pass the level of the significance test, given the significance of 5%. And, it can also be seen that there is a significant difference between the results obtained by the OR_CHR method and PATHARC method and Best solution, while there is no significant difference between the RL method and OR_ARC method and Best solution, and the difference between the RL method and Best solution is much smaller than the OR_AHC difference. It can be seen that the RL algorithm in this paper can solve the storage picking path problem well.

Next, we analyzed the time and space complexity of the algorithm, and the results are shown in Table 7.

Table 7 shows that the spatial complexity of both the heuristic algorithm and Transformer is $O(1)$, and the spatial complexity is the same. As for time complexity, Google's ortools heuristic algorithm uses simulated annealing algorithm, which is based on probability and is a general optimization algorithm derived from the solid annealing principle. Theoretically, the algorithm has global optimization performance, so its complexity cannot be calculated. As a result, PRL’s time complexity is determined in...
comparison to heuristic algorithms, which use probabilistic algorithms and may keep iterating because the optimal solution is not found.

As we all know, Google orTools is developed in C++, and its statements are executed much more efficiently than Python as a scripting language. However, according to the experimental data, the Google ortools heuristic algorithm developed by C++ has no advantage in computing time and optimal path length, and is obviously weaker than the PRL algorithm. Therefore, the PRL algorithm can be regarded as a good choice when solving the problem of warehouse routing.

To sum up, although the average path length planned by the PRL method is slightly longer than the optimal solution accurately solved, it is closer to the optimal solution compared with the three heuristic algorithms of Google ortools. From the mean value of DEV, the error with the optimal solution is 5.1%. From the standard deviation of DEV, the error between the path length solved by the PRL method and the optimal solution is controlled within 7.45% with a probability of 0.8413, while the other three methods are 12.22%, 16.08%, and 12.3%, respectively. From the perspective of time dimension, PRL has strong parallel computing capability, and the solving time is 1/12 of Google OR_tools heuristic algorithm, or even shorter.

### 6. Conclusion

With the rapid development of the Internet of Things technology, optimize the warehouse management system, establish and improve the intelligent picking system, for unmanned, intelligent transformation is the key direction of reform and development of e-commerce enterprises, courier enterprises.

This thesis addresses the picking path planning problem of multiple picking stations in a warehouse in a zoned area, and designs a PRL algorithm using placeholder control to solve the path optimization problem in a shared mode where the number of picking orders is unequal and the picking stations do not need to work synchronously, taking into account two factors: the idleness of the picking stations and the distance of the center of gravity of the order items. The simulation results have shown that the PRL algorithm in this paper is very close to the optimal solution and far better than the three heuristic algorithms of Google’s ortools in terms of solving accuracy. In terms of solving speed, it has the advantage of fast solving and can almost realize real-time solving. In the dimension of structure, it has the advantage of variable input and avoiding retuning. Therefore, this method is very effective to solve the problem of multiple picking stations with different order numbers in intelligent warehouse. It can solve the picking path effectively and reduce the picking cost [23].

---

### Table 5: Algorithm comparison table.

<table>
<thead>
<tr>
<th>Projects</th>
<th>Distance mean (m)</th>
<th>Distance mean error rate (%)</th>
<th>DEV mean value</th>
<th>DEV standard deviation</th>
<th>Total time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best solution</td>
<td>966.68</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>PRL</td>
<td>1015.09</td>
<td>5.01</td>
<td>4.85</td>
<td>2.60</td>
<td>0.7883</td>
</tr>
<tr>
<td>OR_ARC</td>
<td>1049.09</td>
<td>8.52</td>
<td>8.36</td>
<td>3.86</td>
<td>7.0147</td>
</tr>
<tr>
<td>OR_CHR</td>
<td>1071.73</td>
<td>10.87</td>
<td>10.39</td>
<td>5.69</td>
<td>5.8683</td>
</tr>
<tr>
<td>PATH_ARC</td>
<td>1038.23</td>
<td>7.40</td>
<td>6.84</td>
<td>5.46</td>
<td>6.8282</td>
</tr>
</tbody>
</table>

---

### Table 6: Inter-subject effect test.

<table>
<thead>
<tr>
<th>Source</th>
<th>Class III sum of squares</th>
<th>Degree of freedom</th>
<th>Mean square</th>
<th>F</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correction model</td>
<td>1571745.41a</td>
<td>4</td>
<td>392936.353</td>
<td>1772.163</td>
<td>4.97709E-22</td>
</tr>
<tr>
<td>Intercept distance</td>
<td>534.751</td>
<td>1</td>
<td>534.751</td>
<td>2.412</td>
<td>0.138845022</td>
</tr>
<tr>
<td>PRL</td>
<td>14975.580</td>
<td>1</td>
<td>14975.580</td>
<td>67.541</td>
<td>2.52493E-07</td>
</tr>
<tr>
<td>OR_CHR</td>
<td>4.489</td>
<td>1</td>
<td>4.489</td>
<td>.020</td>
<td>0.888531759</td>
</tr>
<tr>
<td>PATHARC</td>
<td>305.100</td>
<td>1</td>
<td>305.100</td>
<td>1.376</td>
<td>0.256955406</td>
</tr>
<tr>
<td>OR_ARC</td>
<td>6074.522</td>
<td>1</td>
<td>6074.522</td>
<td>27.396</td>
<td>6.7336E-05</td>
</tr>
<tr>
<td>Error</td>
<td>3769.359</td>
<td>17</td>
<td>221.727</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>22139373.000</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total after correction</td>
<td>15755147.73</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: a. R-squared = 0.998 (adjusted R-squared = .997)

---

### Table 7: Time complexity and space complexity.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time complexity</th>
<th>Spatial complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR_CHR</td>
<td>—</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>PATHARC</td>
<td>—</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>OR_ARC</td>
<td>—</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>PRL</td>
<td>$O(n^2d)$</td>
<td>$O(1)$</td>
</tr>
</tbody>
</table>
7. Outlook

This paper focuses on the picking center intelligent AGV picking path planning problem, cleverly converting the MDVRP problem into a CVRP problem by two-stage method, and designing an improved Transformer algorithm for path planning, thus realizing real-time picking in picking centers with improved picking efficiency compared to traditional methods. Meanwhile, this paper can be promoted for scenarios such as UAV take-out delivery, port container loading and unloading, and logistics distribution. Since this paper is carried out without considering the limitation of AVG number of picking centers, the following research direction is the optimal path planning of picking centers under the constraint of AVGs’ number.

Data Availability

The data used to support the findings of this study are included within the article. The initial data used to support the findings of this study have been deposited in the NEO repository (https://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/). The data sets are valid and can be accessed.

Disclosure

Deyao Wang, Jun Jiang, and Ran Ma are the co-first authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors’ Contributions

Deyao Wang, Jun Jiang, and Ran Ma contributed equally to this work. Deyao Wang performed conceptualization (providing part ideas), performed formal analysis (some formal analysis, such as time & space analysis and correlational analyses), wrote the final draft, developed the methodology (the modification of MDVRP two-stage solution of picking routing), performed visualization, reviewed & edited the article, and provided the software (support of computer code). Jun Jiang performed conceptualization, performed data curation, performed formal analysis, developed the methodology (MDVRP two-stage solution of picking routing), provided the software (algorithm design for hybrid picking mode, simulation example, and comparison of algorithms), performed validation, wrote the original and final drafts, performed visualization, and reviewed & edited the article. Ran Ma performed conceptualization (providing part ideas), developed the methodology (design of coding model in reinforcement learning algorithm model), provided the software (support of computer code, such as comparison between simulation examples and some algorithms), wrote the original draft, and performed visualization (visualization of simulation example results). Guicheng Shen performed conceptualization and supervision and was responsible for funding acquisition.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (NO: 71831001) and Beijing Key Laboratory of Intelligent Logistics Systems (NO: BZ20211).

References


