Complex & Intelligent Systems (2023) 9:5937–5948
https://doi.org/10.1007/s40747-023-01023-5

ORIGINAL ARTICLE

Collaborative optimization of task scheduling and multi-agent path planning in automated warehouses

Zhang Honglin[1] · Wu Yaohua[1] · Hu Jinchang[2] · Wang Yanyan[1]

Received: 4 December 2022 / Accepted: 25 February 2023 / Published online: 15 April 2023

Abstract

Task scheduling (TS) and multi-agent-path-finding (MAPF) are two cruxes of pickup-and-delivery in automated warehouses. In this paper, the two cruxes are optimized simultaneously. Firstly, the system model, task model, and path model are established, respectively. Then, a task scheduling algorithm based on enhanced HEFT, a heuristic MAPF algorithm and a TS-MAPF algorithm are proposed to solve this combinatorial optimization problem. In EHEFT, a novel rank priority rule is used to determine task sequencing and task allocation. In MAPF algorithm, a CBS algorithm with priority rules is designed for path search. Subsequently, the TS-MAPF algorithm which combines EHEFT and MAPF is proposed. Finally, the proposed algorithms are tested separately against relevant typical algorithms at different scales. The experimental results indicate that the proposed algorithms exhibited good performance.

Keywords Intelligent warehouse · Picking · Task scheduling · Multi-agent path finding · HEFT

Introduction

With industrial development and technological progress, the warehousing system has experienced the evolution from manual warehousing, semi-automated warehousing, automated warehousing to intelligent warehousing. Of which automated warehousing is the most widely used mode in the current industrial world, and intelligent warehousing is the current academic research frontier hotspots.

Automated warehousing is characterized by the widespread use of WCS and WMS to replace manual labor; WMS is responsible for inbound and outbound management, picking management, inventory management, etc., and sends commands to PLCs through information interaction with WCS, while WCS updates and reflects PLC system status in real time. Various types of logistics robots, such as handling robots, shuttle robots, unmanned forklifts, robotic arms, AGVs, etc., are also widely deployed to replace manual labor in a comprehensive manner. However, in the automation stage, WCS and WMS are mainly used in centralized control of warehousing systems, which can no longer get satisfactory results in the face of large-scale storage requirements and distributed control. Therefore, intelligent storage is evolving into a new generation of storage systems and has become a current research hotspot. Intelligent warehouses achieve efficient warehouse management through information technology, Internet of things, mechatronics, etc. IoT technology based on cloud and edge computing gives the devices the ability to compute and share information, 5G technology makes communication more efficient, and distributed systems make dynamic management more flexible.

Whether it is automated storage or intelligent storage, however, its function of storing goods and picking goods remains the same. The picking process has a direct impact on the efficiency of the goods leaving and entering the warehouse, which in turn affects the subsequent delivery schedule of the goods. The optimization of the picking process is therefore, of particular importance. Take the JingDong e-commerce platform as an example, the customer's online

✉ Wu Yaohua
yaohua.wu@sdu.edu.cn

Zhang Honglin
201920522@mail.sdu.edu.cn

Hu Jinchang
jinchang@sdnu.edu.cn

Wang Yanyan
1978wyy@163.com

[1] Faculty of Control Science and Engineering, Shandong University, Jinan, China

[2] School of Business, Shandon Normal University, Jinan, China

orders are sent to the warehouse and picking begins. Orders within a period of time are firstly sequenced with the rules or algorithms, then picking commands are sent to the cluster of picking robots according to the sequencing results, and finally the robots reach the dedicated cargo locations to pick the items and deliver them to the picking stations. From the foregoing description of the picking process we can tell that two critical phases in the process are order sorting phase, which is a task scheduling problem, and pick-up phase, which is a multi-agent path planning problem.

Task scheduling is widely available in various systems, and the order task scheduling in warehouse systems is highly similar to task scheduling in cloud (edge) computing. In the IAAS layer of cloud computing, the common process of task scheduling mainly consists of two steps, task ordering and resource allocation. Firstly, the tasks are divided into several task streams, DAG constraint relations are established, and the tasks are sorted according to the priority design algorithm; then the sorted task streams are assigned to the appropriate servers or virtual machines. In the picking process of a warehouse system, task scheduling also includes two processes: order sorting and order assignment. The orders are first sorted into categories and then assigned to the corresponding picking robots. Therefore, in this paper, we design a task scheduling algorithm for smart warehouse systems with reference to the cloud computing task scheduling.

MAPF is another challenge for order picking. Several robots start from their respective starting points, design, and plan a path to reach the target picking location and deliver the goods to the picking table, a process that requires consideration of two issues. One is how to design the best path globally, and the other is how to ensure the elimination of local conflicts. This is also the difficulty and focus of current MAPF research. In this paper, a streamlined and efficient MAPF algorithm is designed.

In the past three decades, with the continuous innovation and development of warehouse systems, a number of researchers and engineers have focused on solving the order task scheduling problem and the MAPF problem. Most of the existing research has been focused on optimizing for either of the two. These approaches do not consider the coupling relationship that exists between task scheduling and robot path planning in the picking process. However, in the practical picking we must realize that the two are related. On the one hand, the quality of task scheduling directly affects the makespan and efficiency of the system, assuming that the path planning algorithm is known, and on the other hand, the efficiency of the MAPF path affects the progress of secondary task scheduling, assuming that the task scheduling algorithm is known, since orders arrive dynamically. Therefore, considering the integrated task scheduling and MAPF optimization in a dynamic storage environment is the most main purpose of this research.

The remainder of this paper is organized as follows. In the "literature review" section, we list the relevant typical studies. In the "problem description and models" section, we define the problem and build several important models. In the "Task Scheduling Algorithm" section, we design an enhanced HEFT task scheduling algorithm. In the "MAPF algorithm" section, we design a heuristic algorithm based on priority and traffic rules. In the "TS-MAPF" section, we propose a dynamic TS-MAPF co-optimization algorithm considering JIT constraints. In the "Experiments" section, the three proposed algorithms are compared with each other and with several other typical algorithms. Finally, in the "Conclusion" section, we summarize the research of this paper.

Literature review

The classical task scheduling problem is a typical scheduling problem which exists widely in industrial production, computer systems, human resource management and other industries. Since Bruker et al. [1] systematically described the model and solution of task scheduling problem in 1996, many scholars have performed a considerable amount of research work on it. In terms of task scheduling for warehouse systems, the representative research results are as follows. A detailed literature review of order batching and partition scheduling problems in the design and control of manual picking processes is presented by René et al. [2]. Wang et al. [3] investigated the task scheduling problem of multi-tier shuttle warehousing system (MSWS), developed a mixed-integer planning model with the objective of minimizing the maximum completion time, and designed three heuristic algorithms to solve this problem. Wang et al. [4] focused on multiple trips VRP with TW and uncertain travel times. Liu et al. [5] further expanded the problem of [4], and proposed a robust optimization method for pickup and delivery problem. Yin et al. [6] developed the idea of production, inventory and delivery scheduling with two competing agents. Ma et al. [7] transformed the automated warehouse scheduling problem into a multi-objective constrained optimization problem and proposed an ensemble multi-objective biogeography-based optimization (EMBBO) algorithm. Peng et al. [8] explored the integrated optimization problem of location assignment and sequencing in multi-shuttle automated storage/retrieval systems under the modified 2n-command cycle pattern.

As we can see from the analysis in the introduction section, the similarity between warehouse task scheduling and cloud computing task scheduling is very high, so the research related to cloud computing task scheduling has relatively extensive implications for warehouse task scheduling. Hu et al. [9] proposed a multi-objective workflow scheduling algorithm based on particle swarm optimization algorithm to minimize the completion time and cost while satisfying reliability constraints. Wu et al. [10] studied the cloud computing task scheduling problem with deadline constraints and proposed a meta-heuristic algorithm LCAO for minimizing cost. For minimizing the maximum completion time cloud computing task scheduling problem, Tong et al. [11] combined the HEFT algorithm and Q-learning algorithm to design a novel QL-HEFT task scheduling algorithm. The algorithm is divided into two phases: a QL-based task ordering phase and a processor allocation phase based on the earliest completion time policy. Tong's study considers the correlation between task ordering and resource allocation in actual task scheduling, and this paper adopts a comparable solution idea in the processing of storage task scheduling. Zhang et al. [12] proposed an EHEFT-R algorithm to optimize task execution efficiency, QoS, and energy consumption. Sun et al. [13] proposed a heuristic algorithm task type first algorithm (T2FA) for solving deadline-constrained workflow scheduling in cloud with multicore resource. Wang et al. [14] consider a multitasking scheduling model with multiple agents on a cloud manufacturing platform, in which the maximum of a regular function, the total completion time and the weighted number of late jobs are considered as objective functions.

There are also valuable studies in the field of MAPF. Multi-Agent Pathfinding (MAPF) is the problem of finding paths for multiple agents such that every agent reaches its goal and the agents do not collide [15]. Čáp et al. [16] developed an adapted version of classical prioritized planning called revised prioritized planning. Furthermore, Čáp [17] proposed an asynchronous decentralized prioritized planning as a collision avoidance mechanism. Li et al. solved the path conflict and deadlock problem of collaborative robots by combining a priority planning algorithm with a fast re-planning algorithm based on D*Lite. Sharon et al. [18] proposed a novel Conflict Based Search algorithm to optimize MAPF problem. Ma et al. [19] studied the TAPF (combined target-assignment and path-finding) problem and proposed a CBM(Conflict-Based Min-Cost-Flow) algorithm for minimizing the maximum completion time.

Most of the studies have optimized only for one problem in task scheduling or MAPF, which is obviously not realistic. A limited number of comprehensive optimization studies are also reported. Kulak et al. [20] studied joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms. Gils et al. [21] formulated and solved the integrated batching, routing, and picker scheduling problem in a real-life spare parts warehouse. However, both of the above articles categorize the path problem as TSP, which is obviously improper. The MAPF problem is significantly different from TSP, and in addition, conflicts between agent paths in MAPF have to be considered. To the best of the authors' knowledge, no relevant literature has been reported that considers both task scheduling and MAPF for the picking process. Considering the importance of the picking process for the warehousing process, new integrated optimization methods need to be considered and designed.

Problem description and models

System description

This paper describes a task scheduling and MAPF problem for an e-commerce smart warehouse picking process, where there are orders from customers and a group of robots act as actuators to pick the orders and deliver them to the picking table. A concise schematic diagram is shown in Fig. 1, and detailed diagram is shown in Fig. 2.

To describe the TS-MAPF problem, the symbols in the picking process are defined as follows: the picking robot set is presented by R $= \{R_1, R_2, \ldots R_m\}$, and order set is denoted by O $= \{O_1, O_2, \ldots, O_n\}$, each of which consists of some tasks. o_{ij} means the jth task of order O_i. $\omega_{o_{ij}}$ denotes the picking time of o_{ij}, which represents the time a robot waits for the lift to bring down the goods from the target shelf. For instance, robot R_1 is to pick up the item o_{45} located on the sixth layer, then $\omega_{o_{45}} = 6$. $P = \{P_1, P_2, \ldots P_m\}$ represents the set of paths of all robots. $C_{o_{ij}}$ is the completion time of o_{ij}, that is

$$C_{O_{ij}} = P_k^{O_{ij}} + \omega_{O_{ij}} \tag{1}$$

And the completion time of last task is C_{max},

$$C_{max} = \max C_{O_{ij}},\ i \in (1, n),\ j \in (1, m) \tag{2}$$

Task scheduling model

When an order arrives at the warehouse, a complete picking process begins by sorting the order and then assigning it to the appropriate robot to perform the pick. Therefore, task scheduling is the first essential part of the process. In actual, task scheduling consists of two processes: task sequencing and task assignment. The key essence of task sequencing is to classify different task flows from different clients. Task assignment is the assignment of sequenced tasks to the applicable robots.

Typically, users submit one or more groups of orders with priority restrictions, and upon receipt of the orders, the control center sends scheduling requests to the task scheduler. Taking Fig. 3 as an example, assume that o_{41}-o_{34} are 7 subtasks from different user orders. Only when the task o_{41} is completed can o_{13}, o_{26} and o_{15} be executed. Similarly, o_{72} must wait for the completion of the task o_{15} before it can

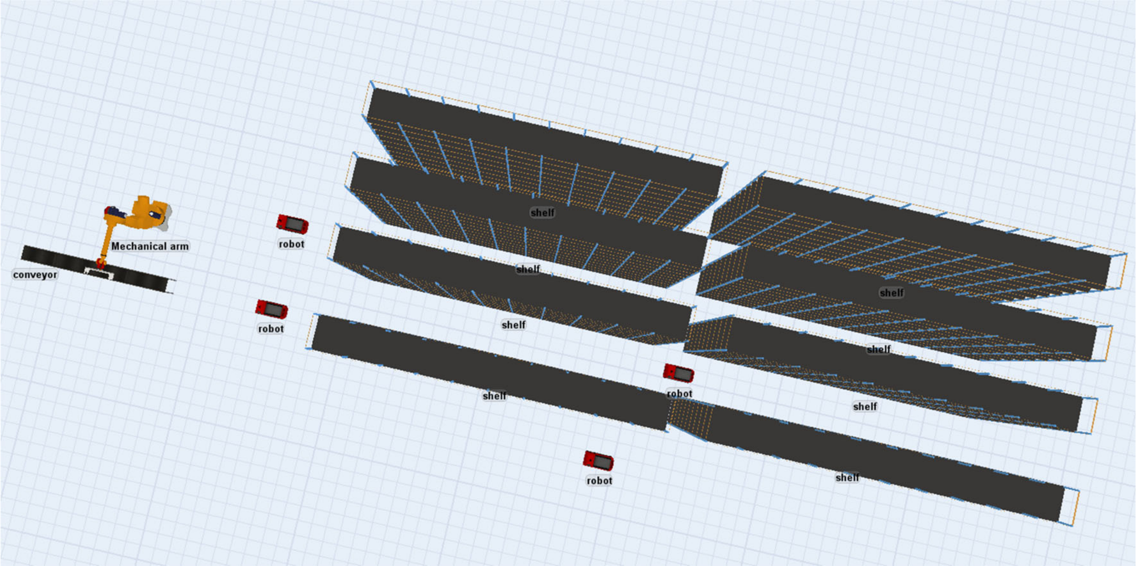

Fig. 1 Sketch of intelligent warehouse system

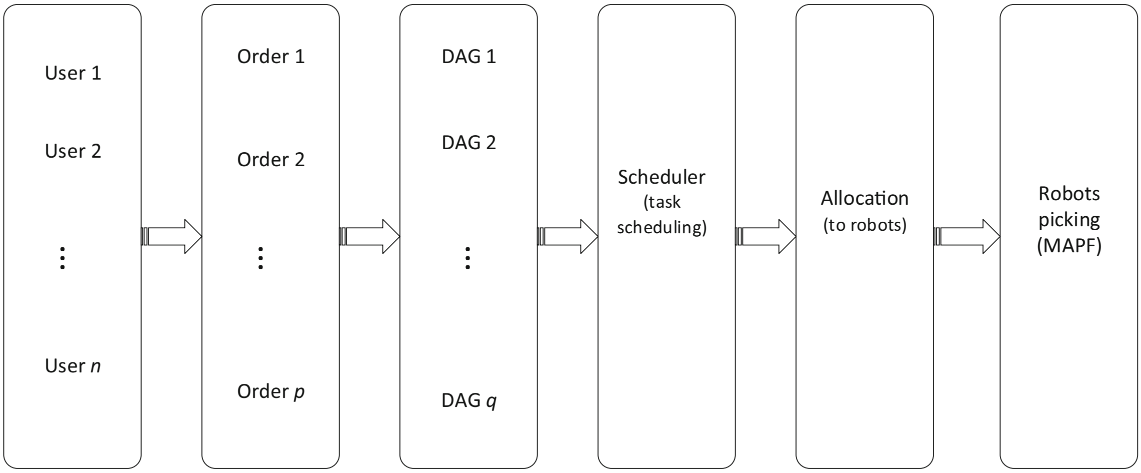

Fig. 2 Diagram of picking process

be executed. From the characteristics of Fig. 3, it can be obtained that task scheduling with priority constraints can be expressed by a directed acyclic graph (DAG).The DAG is expressed as:

$$G = (O, E, S, C) \tag{3}$$

where $\mathrm{O} = \{o_{ij}|\ \ i = 1, \ldots, n;\ j = i, \ldots, n\}$ denotes the set of all tasks in the DAG, in which o_{ij}represents a task, n is the quantity of the tasks. $E = \{E_{ij} = (o_{ij}, o_{lk})|\ \ o_{ij}, o_{lk}\epsilon O\}$represents the set of edges from task to task, in which E_{ij}indicates the edge between $o_{ij}$and o_{lk}. $S = \{S_{ij}^{kl}|\ \ ijkl\}$ denotes the scheduling costs between tasks with edges connected, in which S_{ij}^{kl}represents the scheduling cost of $o_{ij}$and o_{lk}. $C = \{C_{ij}|\ \ i, j = 1, 2, \ldots, n\}$ denotes the completion times of all tasks in a DAG, in which C_{ij}represents the completion time of o_{ij}.

Thus, the makespan of task scheduling equals to the completion time of the last picked task, i.e., the maximum completion time. Therefore, makespan can be calculated by the following formula:

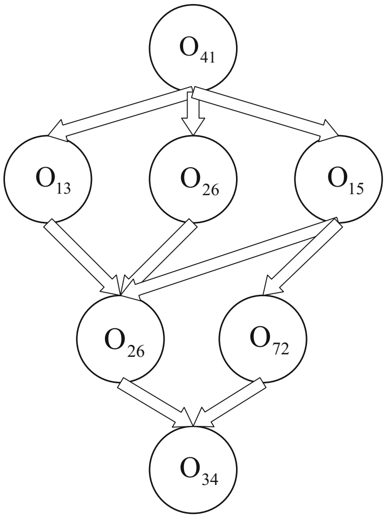

Fig. 3 An example of DAG

$$C_{max} = \max\left(\sum_{i=1}^{n}\sum_{j=1}^{n} t_{ij,1}F_{ij,1} \cdots \sum_{i=1}^{n}\sum_{j=1}^{n} t_{ij,k}F_{ij,k}\right) \tag{4}$$

where

$$F_{ij,k} = \begin{cases} 1, & \text{if} \quad o_{ij} \rightarrow R_k \\ 0, & \text{else} \end{cases} \tag{5}$$

In Eq. (5), $o_{ij} \rightarrow R_k$ indicates that task o_{ij} is assigned to robot R_k, $t_{ij,k}$ denotes the picking time of o_{ij} by R_k.

MAPF model

Once the task is scheduled and assigned, the selected robot instantly performs the picking process. The task execution process is that the robot starts from the starting point, plans an optimal path that will reach the picking point, reaches the target point, picks the goods and delivers them to the conveyors or picking stations. The most important part of this process is path planning.

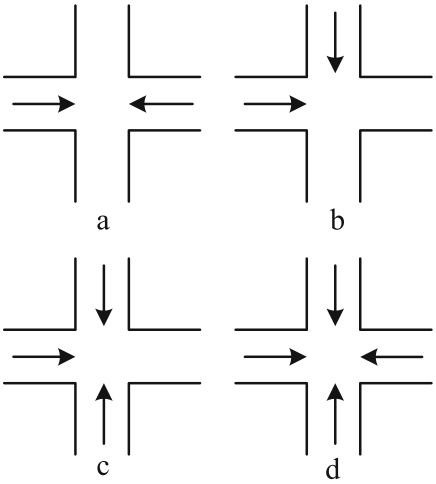

Fig. 5 Four typical conflicts

The plane diagram of a warehousing system is shown in Fig. 4 [22]. Different from the classical VRP problem, MAPF requires more considerations. The primary goal of MAPF is to find the optimal feasible path for each picking robot. In addition, since the aisles in the warehouse are usually narrow, there is a high probability that the robots will have conflicts in their movements. There are several common types of conflicts as shown in Fig. 5. In addition, since the aisles in the warehouse are usually narrow, there is a high probability that the robots will have conflicts in their movements. There are several typical conflicts as shown in Fig. 5. If conflicts are not effectively eliminated, the picking task becomes less efficient and may even lead to picking failure.

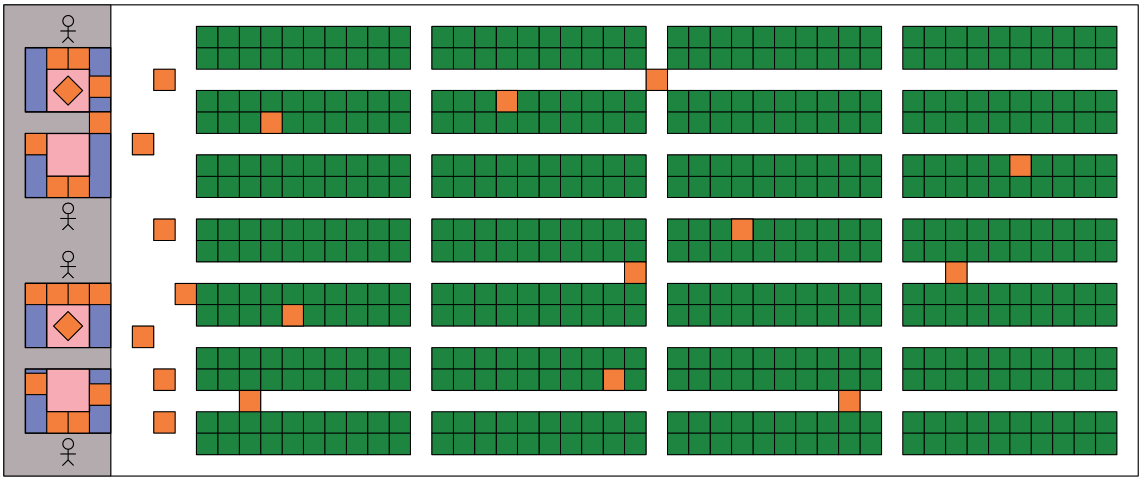

Fig. 4 The plane diagram of a warehousing system

The picking time of the robot to perform the task is expressed as

$$T_R = \sum_{k=1}^{m} t_k \tag{6}$$

$$t_k = g(R_k) \tag{7}$$

$$g(R_k) = \frac{d}{v_R} \tag{8}$$

where T_R denotes the total picking time of all robots; t_k represents the picking time of R_k; d is the length of path of R_k, and v_R denotes the speed of robots.

Mathematical formulation

In the fast-paced e-commerce race, customer satisfaction is usually the most important factor in determining a company's growth prospects. In other words, the quality of service determines the competitiveness of a company. In the case of the picking process in smart warehouses, it is the picking efficiency that determines the quality of service. We choose Quality of Service (QoS) as the objective function of the problem, which is denoted as

$$\text{QoS} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij}^{SPL}}{C_{max}} \tag{9}$$

where T_{ij}^{SPL} represents that the actual picking path of the robot is not considered, and only the shortest path between the nearest target position to the picking table and the picking station is calculated. The QoS function is designed in this way since the actual customer satisfaction during service needs to be quantified in a more demanding way, so it is reasonable to arrange the idealized path length.

Task scheduling algorithm

HEFT algorithm

HEFT algorithm is a heuristic task scheduling algorithm, which is efficient for optimizing the scheduling of tasks with DAG constraints. The principle of the HEFT algorithm is to divide task scheduling into two related phases: task sequencing and resource allocation [12]. The steps of classic HEFT algorithm are as follows [11]:

Step 1. Get scheduling costs of tasks; get scheduling costs of edges;

Step 2. Compute rank value of tasks;

Step 3. Sequence the ranked tasks with non-increasing order into list L_u;

Step 4. Compute T_{ij}^{SPL} of all tasks. Take the first task T_1^u from L_u, for each robot R_k, compute $T_{ij}^{SPL'}(T_1^u, R_k)$, allocate T_1^u to R_k^* that minimize $T_{ij}^{SPL'}$ of T_1^u.

Step 5. Loop until all tasks are assigned.

Step 6. Obtain C_{max}.

EHEFT

Although HEFT is a concise and efficient task scheduling algorithm, HEFT suffers from the same problem as most other two-stage optimization algorithms, i.e., HEFT splits task ordering and resource allocation directly into two sequentially related processes, ignoring the coupling between them, which is not rigorous. To fill the drawback of HEFT, an Enhanced HEFT (EHEFT) algorithm is proposed. The improvement process of EHEFT is as follows.

Firstly, once one or several tasks are determined in their current order, these tasks are immediately assigned to the corresponding robots according to the SPL. For example, suppose tasks o_{13} and o_{31} are ranked first and second, and their corresponding SPL robots are R_3 and R_6, respectively, then o_{13} is assigned to R_3, and o_{31} is assigned to R_6. While all tasks are assigned to robots, an initial resource allocation is also completed. Although such an ordering and assignment is not the optimal choice, an approximate synchronization of task ordering and resource assignment is achieved by such an improvement.

Secondly, since such an ordering can only guarantee scheduling in the optimal order among tasks of different priorities in the DAG, but optimality cannot be guaranteed among tasks of the same level. For example, in Fig. 3, o_{41} has a significantly higher priority than o_{13}, o_{26} and o_{15}, but o_{13}, o_{26} and o_{15} of the same level need to be sorted twice. Therefore, the second improvement strategy of EHEFT is the rescheduling of tasks in the same level of priority. The flow of rescheduling is shown in Fig. 6. Take two neighboring tasks o_{ij} and $o_{ji}(rank_{ij} > rank_{ji})$, if o_{ij} and o_{ji} are at the same level of DAG and assigned to the same robot R_k, then assign o_{ij} to R_k.

MAPF algorithm

As mentioned in the "MAPF model" sub-section, the two goals of MAPF are path planning and conflict elimination. Among them, path planning is the main objective and conflict elimination is the local optimization of the path. Therefore, in this section, we first design a Priority-based Global Multi-Agent Path Planning (PGMAPF) algorithm without considering local conflicts; then we take local conflicts into account and design a conflict avoidance strategy based on traffic rules; finally, in order to make the algorithm have better performance, we further propose an improvement strategy

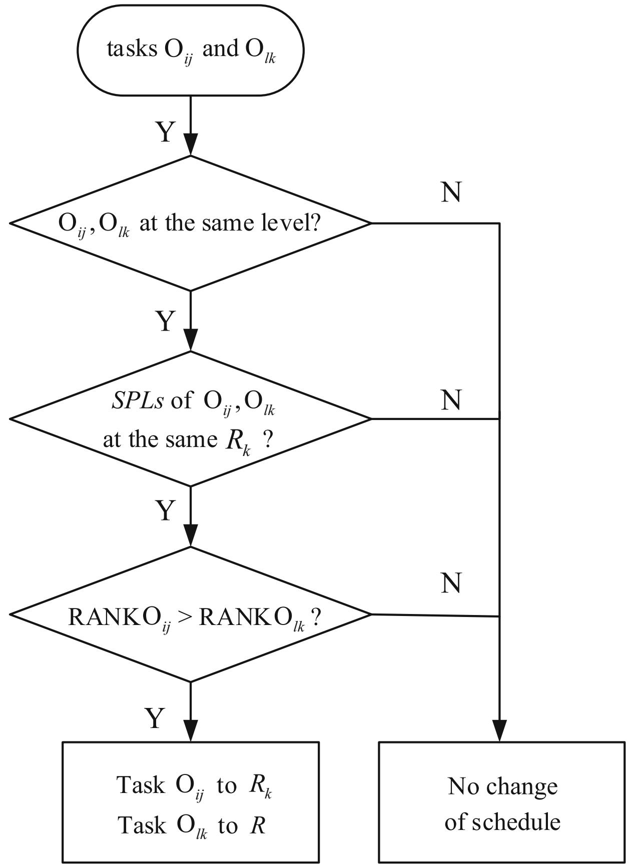
tasks O_{ij} and O_{lk}
Y
O_{ij},O_{lk} at the same level?
N
Y
SPLs of O_{ij},O_{lk} at the same R_k ?
N
Y
RANKO_{ij} > RANKO_{lk} ?
N
Y
Task O_{ij} to R_k
Task O_{lk} to R
No change of schedule

Fig. 6 Flow chart of EHEFT algorithm

considering turning penalties, considering the phenomenon of long path times due to frequent turning of robots in practical situations.

TS-MAPF

The design idea of TS-MAPF is to assign a priority to each robot, and clusters of robots are first sorted in order of priority from highest to lowest. As each robot performs its own path search and planning, it needs to avoid static obstacles in the warehouse as well as all robots with higher priority than it. The pseudo-code of TS-MAPF is shown in Algorithm TS-MAPF. The algorithm starts from the robot with the highest priority and iterates until it iterates to the robot with the lowest priority. Assuming that the highest priority robot is R_1 and the lowest is R_m, the traversal order is R_1-R_m. In the *i*th iteration, the algorithm plans the path of robot R_i while avoiding the spatio-temporal intersection region with robot R_1-R_{i-1}. The trajectory of the robot is obtained from the equation $Best - traj(\omega^{'}, \Delta)$ of Čáp (2017).

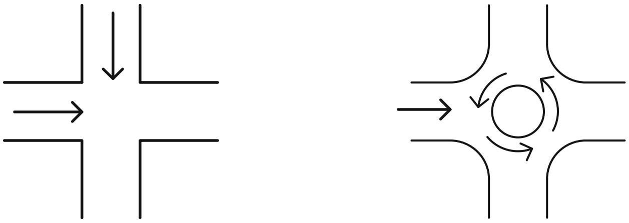

Fig. 7 Design of traffic circles

Algorithm TS-MAPF

```
Algorithm TS-MAPF
  $\Delta \leftarrow \varnothing$;
for $i \leftarrow i, \ldots m$ do
$\pi_i \leftarrow Best - traj(\omega, \Delta)$;
if $\pi_i = \varnothing$ then
return failure and stop the iteration
$\Delta \leftarrow \Delta \cup R_i^{\Delta}(\pi_i)$;
Function $Best - traj(\omega', \Delta)$
return available path for $R_i$
otherwise return $\varnothing$
```

Traffic rules

Although TS-MAPF has constrained the motion rules of different classes of robots by setting the priority of robots, conflicts as shown in Fig. 5 are still inevitable. To further reduce and eliminate these four types of conflicts, we designed a dynamic obstacle avoidance strategy based on traffic rules. The specific traffic rules (TR) are as follows:

TR 1. One-way path rule. Any robot is only allowed to travel in the same direction on the same unchanging path. This resolves the type (a) conflict in Fig. 5 and part of the type (c) and (d) conflicts.

TR 2. Adjacent parallel road direction opposite rule. This rule is designed with reference to the principle of traffic circles, which makes the probability of robots meeting at intersections lower, so as to effectively avoid types (b), (c) and (d) conflicts. The schematic diagram of TR2 is shown in Fig. 7.

TR 3. Robots cannot be reversed. This rule serves as a strong constraint to further constrain the robot's motion behavior.

Turning penalty

In a real-world warehouse environment, the robot's slow turning motion will lead to a consequence that is easily ignored. As shown in Fig. 8, the lengths of path 1 and path 2 are the

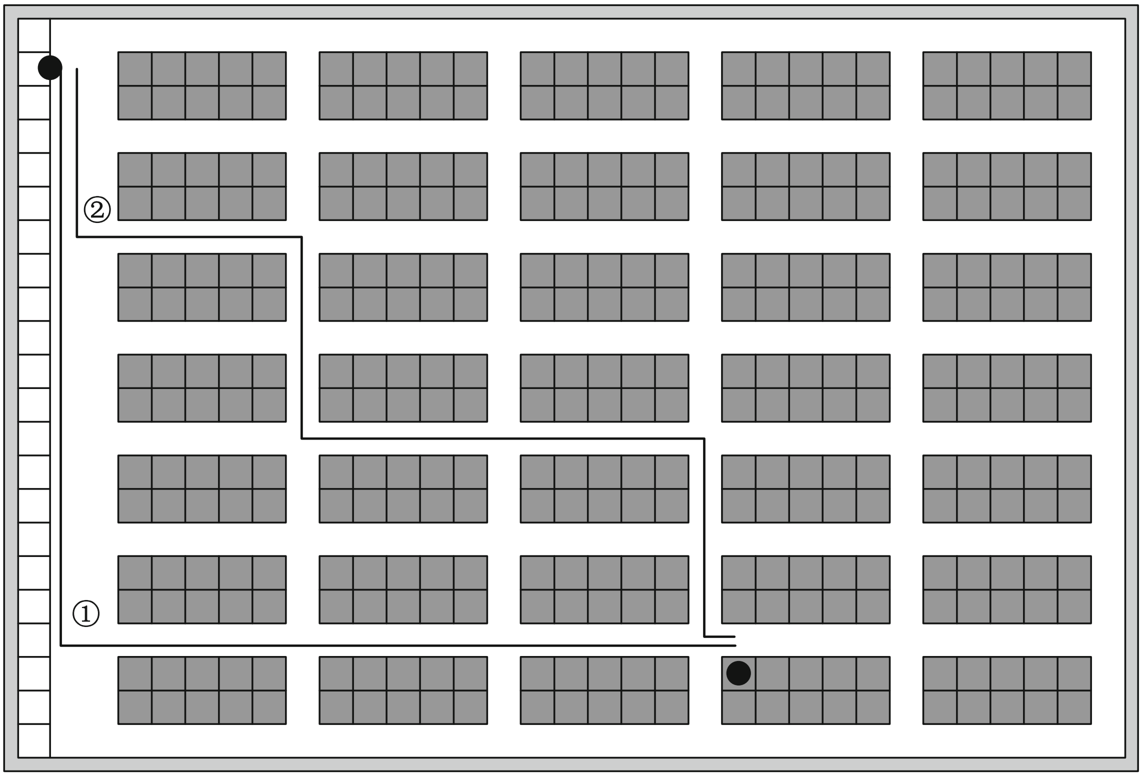

Fig. 8 The effect of turning

same, but since the number of turns for path 2 is 5, while the number of turns for path 1 is 1, this results in path 2 bearing more time cost for the same path length.

Therefore, in order to minimize the number of turns, we modify the distance formula in the "MAPF model" section by adding a steering penalty, and the modified formula is:

$$g(R_k) = \frac{d}{v_R} + \omega t_{turn} \tag{10}$$

where ω is s the penalty factor, which is set to 2, and t_{turn} is the steering time. Once the robot makes a turn, a penalty of the turning time will be added due to $\omega = 2$.

Experiments

In this section, we conducted a large number of comparative experiments. We first validate the effectiveness of the proposed EHEFT task scheduling algorithm; then we validate the effectiveness of the proposed MAPF algorithm with TS-MAPF as the main algorithm and traffic rules and turn penalties as the augmentation strategy and finally we combine the two algorithms to compare with other relevant algorithms.

Experimental setup

The number of robots m is set to 5, 10, 20, and 30. The task sizes are 100, 200, 500, and 1000. These tasks are assigned in a random number of orders. The robot motion speed is set as $v_R = 2m/s$ with turning penalty $\omega = 2$. The measurement of the warehouse is 300 m long and 200 m wide and 50 m high. There are 50 groups of shelves in the warehouse, each group of shelves consists of 10 rows (layers) and 10 columns (length) of 100 compartments. All experiments are conducted under the environment of intel i3-9100F CPU, 16G memory, 500G SSD hard disk, GTX 1605 s graphics board. Three groups of experiments were designed to test the EHEFT task scheduling algorithm, proposed MAPF algorithm and the TS-MAPF algorithm, respectively. All experiments were repeated for 20 times, and the average values are selected as the comparison data.

Performance of EHEFT algorithm

To test the performance of the EHEFT algorithm, multi-agent A* is chosen as the MAPF algorithm and the performance of EHEFT, FCFS, LPT, HEFT, SPT task scheduling algorithms are compared, with QoS as the evaluation function. The experiments were conducted at robot_task numbers of 5_100, 5_200, 10_100, 10_200, 10_500, 10_1000, 20_200, 20_500, 20_1000, 30_200, 30_500, 30_1000 scales, respectively. The experimental results (QoS) are shown in Table 1.

The experimental results are shown in Table 1. From the experimental data, in the small-scale test of 5_100, QoS performance of EHEFT is in the dominant position, where LPT has the worst performance. In the small scale of 5_200, SPT performs better than HEFT, and LPT is still the worst algorithm. In the medium scale 20_500 experiments, EHEFT has established a clear lead. When the problem size becomes further large, EHEFT shows a dominant advantage. In particular, at the 30_1000 scale, EHEFT's QoS is much higher than all other compared algorithms. This indicates that HEFT has good adaptability to both small and medium-sized picking, and has particularly good performance in large-scale picking problems.

Performance of MAPF algorithm

To test the performance of the EHEFT algorithm, we chose HEFT as the task scheduling algorithm and compared the performance of CBS, A*, RRT, PP, and the proposed MAPF algorithms with QoS as the evaluation function. The experiments were conducted at Robot_Task pairs of 5_100, 5_200, 10_100, 10_200, 10_500, 10_1000, 20_200, 20_500, 20_1000, 30_200, 30_500, 30_1000 scales, respectively. The experimental results (QoS) are shown in Table 2.

Table 2 shows the results of MAPF algorithm comparison. In the small- and medium-scale tests of 100 and 200 tasks, selected MAPF algorithms lead alternately, and the proposed MAPF algorithm has a closer performance to other algorithms. As the problem size increases to 500 and 1000 tasks, the proposed MAPF algorithm starts to show much higher QoS than other MAPF algorithms, which indicates that the proposed MAPF algorithm outperforms the mainstream MAPF algorithms in the large-scale path planning problem.

Performance of EHEFT-MAPF algorithm

The two sets of experiments above have verified the effectiveness of the EHEFT task scheduling algorithm and the MAPF algorithm proposed in this paper, respectively. In this section, we combine the two proposed algorithms to optimize the picking process of the system. We compare the tabu search algorithm of [21] and the combination of HEFT and A*, the combination of CBS and RRT.

From the results in Table 3, we can clearly see that the proposed EHEFT-MAPF algorithm is dominant for all problem sizes used for testing. This indicates that the EHEFT-MAPF algorithm can effectively perform picking tasks of various scales within the storage system. In addition, the results in

Table 1 Performance of task scheduling algorithms

	100					200					500					1000				
	EHEFT	HEFT	SPT	LPT	FCFS	EHEFT	HEFT	SPT	LPT	FCFS	EHEFT	HEFT	SPT	LPT	FCFS	EHEFT	HEFT	SPT	LPT	FCFS
5	**0.785**	0.772	0.690	**0.418**	0.503	0.621	0.620	**0.625**	**0.240**	0.505	–	–	–	–	–	–	–	–	–	–
10	**0.803**	0.783	0.702	**0.539**	0.542	**0.756**	0.603	0.682	**0.257**	0.549	**0.518**	0.382	0.494	**0.088**	0.469	–	–	–	–	–
20	–	–	–	–	–	**0.830**	0.662	0.704	**0.419**	0.583	**0.698**	0.427	0.515	**0.190**	0.518	**0.591**	0.174	0.267	**0.031**	0.198
30	–	–	–	–	–	**0.961**	0.730	0.783	0.653	**0.586**	**0.839**	0.601	0.684	**0.247**	0.593	**0.702**	0.228	0.312	**0.094**	0.474

The best and worst values in each group of experiments are highlighted in bold

Table 2 Performance of path planning algorithms

	100					200					500					1000				
	CBS	A*	RRT	PP	Proposed	CBS	A*	RRT	PP	Proposed	CBS	A*	RRT	PP	Proposed	CBS	A*	RRT	PP	Proposed
5	0.337	**0.482**	0.431	**0.301**	0.429	**0.219**	0.364	0.341	**0.391**	0.358	–	–	–	–	–	–	–	–	–	–
10	**0.432**	0.660	**0.691**	0.418	0.583	**0.496**	**0.441**	0.491	0.452	0.452	0.382	**0.339**	0.399	**0.416**	0.368	–	–	–	–	–
20	–	–	–	–	–	0.572	0.518	**0.492**	0.507	**0.693**	0.442	0.498	**0.319**	0.383	**0.584**	**0.284**	0.316	0.328	0.301	**0.425**
30	–	–	–	–	–	**0.608**	**0.497**	0.526	0.513	0.599	0.475	0.498	**0.403**	0.441	**0.531**	**0.314**	0.401	0.341	0.408	**0.485**

The best and worst values in each group of experiments are highlighted in bold

Table 3 Performance of TS–MAPF algorithms

	100				200				500				1000			
	Tabu	HEFT + A*	CBS + RRT	Proposed	Tabu	HEFT + A*	CBS + RRT	Proposed	Tabu	HEFT + A*	CBS + RRT	Proposed	Tabu	HEFT + A*	CBS + RRT	Proposed
5	0.586	**0.497**	0.518	**0.798**	**0.399**	0.387	0.413	**0.667**	–	–	–	–	–	–	–	–
10	0.631	0.574	**0.531**	**0.863**	0.421	**0.413**	0.447	**0.726**	**0.334**	0.371	0.404	**0.628**	–	–	–	–
20	–	–	–	–	0.581	**0.526**	0.553	**0.884**	0.498	0.412	**0.410**	**0.801**	0.431	0.339	**0.318**	**0.599**
30	–	–	–	–	0.778	0.693	**0.678**	**0.931**	0.589	**0.501**	0.593	**0.857**	0.442	0.541	**0.416**	**0.708**

The best and worst values in each group of experiments are highlighted in bold

Table 3 illustrate the superiority of the proposed TS-MAPF algorithm.

Conclusion

This paper investigates the problem of integrated task scheduling and multi-agent-path-finding of pickup-and-delivery in automated warehouses. The goal is to maximize quality of service. To improve the efficiency of task sequencing and allocation, an enhanced HEFT algorithm is proposed. Subsequently, a MAPF algorithm with priority planning as the main algorithm and traffic rules and steering penalties as augmented strategies is proposed. Simulation experiments validate the effectiveness of the proposed algorithms and demonstrate that the co-optimization algorithm shows very favorable performance in the case of TS-MAPF co-optimization. The main contributions are as follows:

(1) An intelligent warehouse picking optimization problem that integrates task scheduling and path planning is proposed, and the system model, task scheduling model, and MAPF model are constructed.
(2) This paper analyzes the correlation between picking and distribution task scheduling and cloud computing task scheduling, and implements the scheduling of storage tasks by improving the cloud computing task scheduling algorithm.
(3) An EHEFT task scheduling algorithm based on rescheduling strategy is proposed; a MAPF algorithm based on priority planning, traffic rules and turning penalties is presented.
(4) Experiments of different scales verify the effectiveness of both algorithms and the superiority of both algorithms when applied simultaneously.

In the next research, we plan to further implement task scheduling and online real-time path scheduling, and further eliminate the efficiency loss caused by offline scheduling.

Acknowledgements The authors would like to thank Prof. Yin Yunqiang for his professional guidance and thank BlueSword Intelligent Technology Co. Ltd (688557.SH) for providing the experimental environment and equipment. They would also like to thank the National Natural Science Foundation of China (62273204) and the Youth Foundation of Shandong Natural Science Foundation (ZR2022QF109).

Data availability The data are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The corresponding author states that there is no conflict of interest.

References

1. Brucker P, Krämer A (1996) Polynomial algorithms for resource-constrained and multiprocessor task scheduling problems. Eur J Oper Res 90(2):214–226
2. Roodbergen L (2007) Design and control of warehouse order picking: A literature review. European Journal of Operational Research, 2007.
3. Wang Y, Liu Z, Huang K, et al (2020) Model and solution approaches for retrieval operations in a multi-tier shuttle warehouse system. Computers & Industrial Engineering, 141(Mar.):106283.1–106283.9.
4. Dujuan W, Jiaqi Z, Xiaowen W, Cheng TCE, Yunqiang Y, Yanzhang W (2019) Integrated production and multiple trips vehicle routing with time windows and uncertain travel times. Comput Oper Res 103(2019):1–12
5. Xiaochang L, Dujuan W, Yunqiang Y, Cheng TCE (2023) Robust optimization for the electric vehicle pickup and delivery problem with time windows and uncertain demands. Comput Oper Res 151(2023):106119
6. Yunqiang Y, Yongjian Y, Dujuan W, Cheng TCE, Chin-Chia W (2018) Integrated production, inventory, and batch delivery scheduling with due date assignment and two competing agents. Nav Res Logist 2018(65):393–409
7. Ma H, Su S, Simon, D, Fei M (2015) Ensemble multi-objective biogeography-based optimization with application to automated warehouse scheduling. Engineering Applications of Artificial Intelligence, 44(SEP.), 79–90.
8. Peng, Y, Peng, Y (2015) Integrated optimization of storage location assignment and sequencing in multi-shuttle automated storage/retrieval systems under modified multi-command cycle. 2015 IEEE International Conference on Information and Automation (ICIA) IEEE.
9. Hu H, Li Z, Hu H, Chen J, Ge J, Li C, et al (2018) Multi-objective scheduling for scientific workflow in multicloud environment. Journal of Network and Computer Applications, 114(JUL.), 108–122.
10. Wu Q, Ishikawa F, Zhu Q, Xia Y, Wen J (2017) Deadline-constrained cost optimization approaches for workflow scheduling in clouds. IEEE Transactions on Parallel and Distributed Systems, PP (12), 1–1.
11. Tong Z, Deng X, Chen H, Mei J, Liu H (2019) QL-HEFT: a novel machine learning scheduling scheme base on cloud computing environment. Neural Computing and Applications.
12. Zhang H, Wu Y, Sun Z (2021) EHEFT-R: multi-objective task scheduling scheme in cloud computing. Complex & Intelligent Systems, 1–8.
13. Sun Z, Gu C, Zhang H, Huang H (2021) T2FA: A Heuristic Algorithm for Deadline-constrained Workflow Scheduling in Cloud

with Multicore Resource. 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), 345–354.

14. Dujuan W, Yugang Y, Yunqiang Y, Cheng TCE (2021) Multi-agent scheduling problems under multitasking. Int J Prod Res 59(12):3633–3663
15. Stern R. Multi-agent path finding - an overview. Ben Gurion University of the Negev, Be'er Sheva, Israel.
16. Čáp M, Novak P, Kleiner A, Selecky M (2015) Prioritized planning algorithms for trajectory coordination of multiple mobile robots. IEEE Trans Autom Sci Eng 12(3):835–849
17. Čáp M (2017) Centralized and Decentralized Algorithms for Multi-Robot Trajectory Coordination.
18. Sharon G, Stern R, Felner A, Sturtevant N. R (2015) Conflict-based search for optimal multi-agent pathfinding. Artificial Intelligence, 219(feb.), 40–66.
19. Ma H, Koenig S (2016) Optimal target assignment and path finding for teams of agents. International Foundation for Autonomous Agents and Multiagent Systems.
20. Kulak O, Sahin Y, Taner, & M., E. (2012) Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms. Flex Serv Manuf J 24(1):52–80
21. Van Gils T, Ramaekers K, Braekers K et al (2018) Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions. Int J Prod Econ 197(3):243–261
22. Ma H, Koenig S. AI buzzwords explained. AI Matters, 2017.