International Journal of Industrial Engineering Computations 17 (2026) 505–520

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Research on collaborative scheduling of shelves, mobile robots, and storage location resources in mobile shelf storage systems

Hongtao Tang[a]*, Penglong Chen[a], Fei Dai[b], Qingfeng Chen[a] and Xuesong Xu[a]

[a]*College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, P.R.China,310023*
[b]*R&D Center, Shanghai Medconn Medical Diagnostics Technology Co., Ltd., Shanghai, China, 201508*

CHRONICLE

Article history:
Received January 6 2026
Received in Revised Format January 28 2026
Accepted March 5 2026
Available online March 5 2026

Keywords:
Mobile shelf storage systems
Collaborative optimization
Material order task allocation
AGV path planing
Integrated multi-decision making

ABSTRACT

To improve the efficiency of mobile robots carrying shelves for material handling in manufacturing enterprises, this study considers multiple material requirements of orders in a coordinated manner and investigates the transfer of multi-load shelves between storage areas and picking workstations. Three types of resources, shelves, automated guided vehicles (AGVs), and storage locations, are jointly considered, and the task allocation and path planning problem for AGVs is studied within an integrated framework. With the objective of minimizing the overall order picking completion time, a mixed-integer programming model is established to simultaneously represent decisions including resource allocation, path selection, and task sequencing, enabling the optimization of multiple interdependent decisions within a unified scheduling framework. To solve the proposed model, an improved genetic algorithm is developed. The algorithm adopts a chromosome encoding scheme based on the "pick–sort–deliver" shelf-handling task unit and constructs multi-granularity crossover operators at the order level, AGV level, and task level to effectively address the multi-resource and multi-level decision structure of the problem. In addition, five neighborhood search operators are designed to form a hierarchical neighborhood search mechanism, further enhancing local solution quality. Experimental results show that the hybrid genetic algorithm significantly outperforms the comparative methods in maximum order completion time, with average improvements of 19.06% and 16.38% over the greedy algorithm and greedy-local search algorithm, respectively. The algorithm also exhibits satisfactory stability and robustness across various problem scales.

1. Introduction

With the rapid development of manufacturing industry, factories' requirements for logistics timeliness continue to increase, and intelligent storage systems have become a key part of modern logistics supply chain (Jiang et al., 2025). In the mixed flow assembly production line, there are diverse product types and there are fewer material requirements for the products at the edge of the line, resulting in higher material distribution frequency requirements. In the traditional picking mode where people mainly search for goods, the picking personnel need to walk long distances between shelves, which is labor-intensive. The picking efficiency is easily affected by human factors, making it difficult to stably support high-frequency and volatile order demands in the long run. To adapt to the distribution of multiple varieties and small batches of material orders in production, an AGV assisted order picking system is used to meet the material picking needs in manufacturing production. The goods to people delivery system can effectively cope with material picking in production. In the Robot Mobile Fulfillment System (RMFS), AGV transports the moving shelves to the picking workstation, and the picking personnel can complete order picking without moving, which improves picking efficiency. AGV needs to frequently transport shelves to the picking workstation and material warehouse. To improve picking efficiency, multiple materials are stored on the shelves, which can transport more materials at once and reduce the number of shelf handling times caused by the diversity of material types. Under given system conditions, a reasonable resource scheduling strategy can effectively improve system operational efficiency. During the operation of RMFS, the system needs to coordinate multiple types of decision-making problems: on the one hand, it is necessary to develop appropriate order batching and allocation strategies while meeting the multi material

* Corresponding author
E-mail tanght@zjut.edu.cn (H. Tang)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2026 Growing Science Ltd.
doi: 10.5267/j.ijiec.2026.3.004

requirements of orders; On the other hand, it is necessary to arrange the transfer method of multi load shelves between the storage area and the picking area, allocate tasks and plan paths for multiple AGVs, and determine the order in which shelves arrive at the picking area to control order completion time and system congestion level. These decisions are interrelated and mutually constrained, and their essence can be abstracted as a multi resource collaborative scheduling optimization problem under certain layout and job constraints.

At present, there is relatively little research on multi resource collaborative scheduling and multi decision integrated collaboration in the optimization of RMFS systems. In terms of shelf scheduling, Yang et al. (2021) studied the joint optimization problem of order sorting and shelf scheduling in RMFS, and the results significantly reduced the number of AGV tasks; Gharehgozli and Zaerpou (2020) described the basic problem as an asymmetric traveling salesman problem, providing different priorities for orders of different urgency levels. In terms of AGV scheduling and path planning, Rahman and Nielsen (2019) studied the path planning and task scheduling of a single AGV, using genetic algorithms and greedy algorithms to ensure the timeliness of material delivery tasks in container terminal environments under single objective conditions; Dai et al. (2019) established a multi-objective optimization model that considers both energy consumption and completion time; Zhang et al. (2020) considered the flexible job shop scheduling problem with multiple time constraints and improved the initial solution generation, crossover method, and adaptive weight mechanism using an improved genetic algorithm. In terms of collaborative improvement between order processing and shelf scheduling, Shi et al. (2024) proposed a performance oriented synchronous assignment method (Fulfillment Focused Simulated Assignment, FFSA), which can simultaneously determine the allocation of orders and shelves to multiple picking stations for large-scale RMFS order picking optimization problems. Jiang et al. (2020) studied the Picking Replenishment Synchronization Mechanism (PRSM) in a robot forward storage warehouse and proposed a synchronization mechanism to optimize the balance between replenishment workload and picking efficiency. Valle and Beasley (2021) studied the problems of order allocation, shelf allocation, and shelf sorting in a mobile shelf environment, providing important methods for the collaborative optimization of order processing and shelf scheduling. Xie and Otten (2024) introduced the concept of combination stations, which allow workstations to simultaneously perform picking and restocking operations. They analyzed the efficiency of combination stations using queuing theory, and the results showed that combination stations can reduce the number of robots required for stability and significantly reduce order turnover time. In terms of collaborative optimization of order processing and AGV scheduling, Yuan and Gong (2017) studied Bot In time delivery in RMFS, established a queuing network model to describe RMFS with two robot sharing protocols, calculated the optimal number and speed of robots, and provided effective design rules for RMFS. Zou et al. (2017) studied the RMFS allocation rules for online retailers, proposed an allocation rule based on processing speed, designed a neighborhood search algorithm to find the approximate optimal allocation rule, established a semi open queuing network model, and used a two-stage approximation method for performance estimation.

However, given the escalating complexity of modern warehousing systems, comprehensive optimization necessitates multi-resource coordination beyond single or pairwise collaborations.The system needs to consider the integration and collaborative scheduling of multiple resources to improve order fulfillment efficiency. Hashemi and Ranjbar (2024) studied the resource allocation and path planning problems in RMFS, considering order sorting, shelf allocation, shelf scheduling, and robot paths. The study also considered multiple decision variables, reflecting the idea of multi resource collaborative optimization. However, the main focus was on the collaboration between shelves and AGVs, and the consideration of location resources was relatively insufficient. Gharehgozli and Zaerpour (2020) studied the robot scheduling problem for shelf retrieval in RMFS, considering the collaborative optimization of shelf scheduling and AGV scheduling.

Based on a systematic review of relevant literature, current research on AGV collaborative scheduling in warehousing systems has the following characteristics: most studies focus on single resource optimization or pairwise resource collaboration, with few studies on unified modeling and collaborative scheduling of the three types of resources: shelf AGV location; Existing research often adopts fixed coupling or phased optimization methods, which makes it difficult to achieve flexible resource scheduling and deep collaboration.

Therefore, this paper takes the line edge warehouse of manufacturing enterprises as the research object, and based on the collaborative scheduling of the three resources of shelf AGV location in the AGV storage system, optimizes the problem of material order picking, selection, and delivery, comprehensively considering the two decisions of order task allocation and AGV path optimization. This article aims to minimize the maximum completion time and regards workstation allocation as a scheduling decision variable deeply coupled with order allocation, shelf selection, and path planning, rather than an independent long-term planning decision or local optimization objective; Simultaneously optimizing based on the overall performance of the system, rather than local optimization or phased optimization of individual AGVs; By using the concept of the “pick sort deliver” task unit, the complete process of AGV's shelf handling (from picking station, transportation to picking area, picking operation, transportation to storage station) is modeled as a unified decision-making unit to achieve collaborative decision-making throughout the entire process.

Based on these findings, this study has three main contributions. Firstly, a unified optimization model was established for the four elements of order shelf AGV workstation, providing a relatively more complete modeling perspective for multi resource collaborative scheduling problems. Secondly, from the perspective of static scheduling, analyze the collaborative relationship between multiple resources, so that order task allocation, shelf selection, workstation selection, and AGV path work together within the same optimization framework, rather than indirectly coupling through multi-stage or external rules, to achieve unified scheduling of multiple decisions. Thirdly, a hybrid genetic algorithm is designed for multi resource collaborative scheduling problems. When constructing individuals, chromosome encoding is carried out with a single "pick put" complete job process as the encoding granularity. In the evolutionary stage, multi granularity crossover operators such as order level, AGV level, task level, and uniform crossover are constructed to adapt to multi resource and multi-level decision structures. Finally, multiple neighborhood operators including intra path task exchange, cross AGV task relocation, task migration, and shelf position adjustment are designed around the characteristics of multi resource collaboration, forming a multi-level sub neighborhood search system from large to small.

The remaining structure of this study is arranged as follows. The second section reviewed relevant literature, the third section established theoretical models and analyzed relevant mechanisms, the fourth section introduced improved genetic algorithms, the fifth section conducted numerical experimental analysis, and the sixth section summarized.

2. Literature Review

To more intuitively demonstrate the differences between this study and existing research, Table 1 systematically compares representative literature from three dimensions: problem consideration factors, optimization methods, and solution methods.

Table 1

Comparative analysis between our research and representative literature Model Notation

Document	Order allocation	Shelf allocation	Workstation allocation	Path Planning
Zou et al. (2017)	×	√	×	×
Gharehgozli et al.(2020)	×	√	L	√
Kim et al. (2020)	×	√	×	×
Valle & Beasley (2021)	√	√	×	×
Yang et al. (2021)	√	√	×	×
Li et al. (2022)	√	√	×	√
Justkowiak et al. (2023)	√	√	×	×
Hashemi et al. (2024)	√	√	×	√
Liu et al. (2024)	×	√	P	√
Lu et al. (2024)	×	√	P	×
Zhuang et al. (2024)	√	√	P	×
Li et al. (2025)	√	√	×	√
Tang et al. (2025)	√	√	×	√
Yang et al. (2025)	√	√	B	×
This study	√	√	√	√

Note: √ indicates consideration of the factor, × indicates no consideration of the factor; L=Local (single AGV local optimization); P=Planning (Long term Planning); B=Balancing (workload balancing)

Most studies focus on single resource optimization or pairwise resource collaboration, with few studies on unified modeling and collaborative scheduling of shelf AGV location resources; Existing research often adopts fixed coupling or phased optimization methods, which makes it difficult to achieve collaborative resource scheduling. Existing research often only covers partial combinations of order allocation, shelf allocation, workstation allocation, and path planning when modeling problems.

Early research such as Zou et al. (2017) and Kim et al. (2020) mainly focused on single resources or single decision levels, such as shelf workstation allocation rules or item allocation problems, without involving collaborative decision-making between multiple types of resources. As research deepens, Valle and Beasley (2021), Yang et al. (2021) and Justkowiak and Pesch (2023) have begun to focus on collaborative optimization of orders and shelves, reducing shelf access times through joint optimization of order allocation, shelf selection, and sorting. However, it is usually assumed that shelves return to their original positions or do not explicitly model workstation selection. The role of warehouse layout and workstation resources in scheduling is relatively indirect. Some studies have attempted to incorporate more combinations of decision elements into the model, but there are still certain limitations in terms of decision hierarchy and coupling methods. Although Li et al. (2022) and Liu et al. (2024) considered both task allocation or storage allocation and path planning, they adopted a phased optimization approach of “allocation before planning”, with different decisions mainly coupled in sequence; Hashemi and Ranjbar (2024) simultaneously considered order sorting, shelf scheduling, and robot path planning, and established a mixed integer programming model, but required shelves to return to their original positions after picking, and workstation allocation was still based on a fixed layout. Gharehgozli and Zaerpour (2020) allow shelves to return multiple candidate storage locations, modeling the problem as a Generalized Traveling Salesman Problem (GATSP). However, the research focuses on shelf retrieval scheduling for a single AGV, with optimization objectives primarily focused on single vehicle travel time, without incorporating multi AGV collaboration and system level scheduling decisions. Lu et al. (2024) and Zhuang et al. (2024) considered storage allocation as a long-term planning problem based on historical data, mainly optimizing the allocation relationship between SKUs shelves or shelves regions, with limited coupling with scheduling decisions such as order allocation and path planning during operation. Yang et al. (2025) considered workstation allocation to some extent, but focused more on workload balancing rather than dynamically optimizing warehouse layout during scheduling, and did not involve AGV path decision-making.

Therefore, this study establishes a unified optimization model for the four elements of order shelf AGV workstation, incorporating workstation selection as a scheduling decision variable at the same level as order allocation, shelf selection, and AGV path into the model, and analyzing it around the maximum completion time of the system in a static scheduling scenario. This study depicts four types of decision elements simultaneously under the unified model, which provides a more complete modeling perspective for multi resource collaborative scheduling problems in terms of the coverage of decision elements.

3. Model

Consider an RMFS system for a manufacturing enterprise, which includes three main functional areas: storage area, picking area, and AGV parking area. Each storage station accommodates one shelf, and the AGV carries the shelf through the passage to quickly reach the target area; The picking area is equipped with a picking workbench, and there are personnel next to the workbench to perform order picking tasks.

During the operation of the system, the order management system decomposes the order into material requirements. The AGV matches the corresponding shelves according to the order material requirements and transports them to the picking workbench. After the picking personnel complete the material picking of the order, the AGV carries the shelves back to the warehouse storage area.

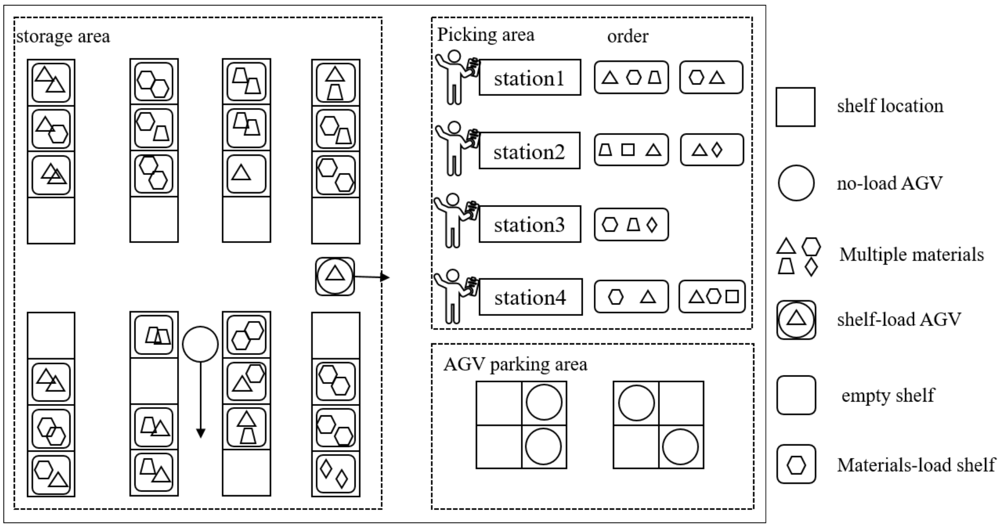

Fig. 1. Schematic diagram of warehouse system layout

This study coordinates the resources of shelves, AGVs, and storage locations to optimize system operation with the goal of timely material order picking. Consider coordinating the order picking process and shelving within each picking station under the conditions of static orders, warehouse inventory, and station layout AGV、 Coordinate the operation of three types of resources in the storage location to ensure that all orders are picked before their respective deadlines.

To cope with the complexity brought by multi resource collaborative decision-making, this study proposes a resource collaborative scheduling mechanism. This mechanism models and collaboratively optimizes AGV, shelves, and workstations as independent and controllable resources, breaking through the limitations of fixed resource binding or phased optimization in traditional research. Realize flexible collaboration of three types of resources through the following methods:

(1) Refined path planning: AGV can select the optimal path based on the current system state, without being limited by fixed paths. AGV determines the order of accessing workstations based on task requirements and system status,.

(2) AGV reuse and multi station visit: The same AGV can continuously process multiple shelves, forming a task sequence; AGV can visit multiple different workstations according to task requirements.

(3) Rack reallocation: After picking, the shelves may not necessarily return to their initial workstations, but can be placed in other vacant workstations to further optimize the warehouse storage layout.

(4) Optimization of picking order: The order in which shelves arrive at the picking area can be adjusted according to order requirements, rather than being executed in a fixed order.

In order to formalize and symbolically define the mathematical model for collaborative scheduling research of shelf AGV location resources, the required symbols and parameters are defined using Table 2:

Table 2
Model Notation

symbol	Description
I	Collection of shelves,$i \in I$
J	Collection of shelf workstations,$j \in J$
J_p	Picking area workstation,$J_p \subset J$
J_s	Storage area workstation,$J_s \subset J$
J_a	AGV parking area workstation,$J_a \subset J$
K	Collection of AGV,$k \in K$
M	Collection of material types,$m \in M$
O	Collection of select order,$o \in O$
R_i	The set of available visit numbers for shelf i， $r \in R_i = \{1,2,\ldots,R_{i,\max}\}$
$\mathcal{H}$	Collections of visit， $\mathcal{H} = \{(i,r) \mid i \in I, r \in R_i\}$， R_i denote the r-th visit to shelf i
$d_{j,j'}$	Distance from workstation j to workstation j'
v_k	The moving speed of AGV k
t^L	Time required for AGV to load shelves
t^U	Time required for AGV to unload shelves
t_m^P	Time required to select one unit of material m
$q_{i,m}$	The initial quantity of material m stored on shelf i
$q_{o,m}$	The demand for material m in order o
d_o	Deadline for order o
c_i	Capacity of shelf i
l_i	The initial workstation of shelf i,$l_i \in J_s$
l_k	The initial workstation of AGV k,$l_k \in J_a$
$R_{i,\max}$	Upper bound on the maximum number of times shelf i can be visited repeatedly
M_L	A sufficiently large parameter
$x_{k,i,j,j',r}$	If AGV k transports it from workstation j to workstationj'during the r-th visit to shelf i, take 1; otherwise, take 0
$\theta_{k,i,r}$	If AGV k is assigned to perform the r-th visit to shelf i, take 1; Otherwise, take 0.
$\lambda_{i,r}$	If the r-th visit to shelf i is activated (i.e. the visit does need to be executed), take 1; Otherwise, take 0.
$y_{k,i,r,i',r'}$	If AGV k completes the visit (i,r) first and then executes the visit (i',r'), take 1; Otherwise, take 0.
$w_{i,o,m,r}$	On the r-th visit to shelf i, select the quantity of material m for order o
$\rho_{i,m,r}$	Remaining inventory level of material m on shelf i after the r-th visit.
$\lambda_{i,r}$	If the service order o is visited for the r-th time on shelf i, take 1; otherwise, take 0
$\gamma_{i,o,r}$	If visiting (i, r) to provide services for order o, take 1; Otherwise, take 0
$s_{k,i,j,r}$	The time when AGV k arrives at workstation j during the visit (i,r).
$f_{k,i,r}$	The r-th visit of AGV k moving shelf i completed picking in the picking area
$\delta_{i,r,i',r'}$	If the visit (i,r) occupies the picking station before the visit (i',r') , take 1; Otherwise, take 0.
T	Upper bound of system completion time (makespan)

The objective function is to minimize the maximum completion time of the system, that is, to complete the last picking task as early as possible.

$$min\ T \tag{1}$$

To ensure that T accurately reflects the completion time of the system, it is necessary to associate T with the completion time of all executed visits:

$$T \geqslant f_{(k,i,r)} - M_L\left(1 - \theta_{(k,i,r)}\right), \quad \forall k \in K, (i,r) \in H \tag{2}$$

The constraints of the model are divided into five categories: task assignment and travel path of shelf AGV workstation, time linkage during AGV execution of "pick put" operation, demand and inventory between shelf order material, order completion time and delivery deadline, and space occupation and decision variable range of workstation shelf AGV.

The first type of constraint pertains to the task assignment and path structure of the shelf AGV storage location.

$$\sum\nolimits_{(k \in K)} \sum\nolimits_{(j \in J_s)} \sum\nolimits_{(j' \in J_p)} x_{(k,i,j,j',r)} = \lambda_{(i,r)}, \quad \forall (i,r) \in H \tag{3}$$

$$\sum\nolimits_{(k \in K)} \sum\nolimits_{(j_p \in J_p)} \sum\nolimits_{(j \in J_s)} x_{(k,i,j_p,j,r)} = \lambda_{(i,r)}, \quad \forall (i,r) \in H \tag{4}$$

Constraints (3) and (4) jointly stipulate that when visit (i, r) is activated($\lambda_{i,r} = 1$), there must be a complete path departing from a certain shelf station, heading towards the picking station p_k bound to AGV k, and finally returning to a certain shelf station; If the visit is not activated ($\lambda_{i,r} = 0$), the corresponding pickup arc and return arc will all be set to 0, and the system will not arrange any AGV travel for them.

$$\sum\nolimits_{(j \in J_s)} \sum\nolimits_{(j' \in J_p)} x_{(k,i,j,j',r)} = \sum\nolimits_{(j_p \in J_p)} \sum\nolimits_{(j \in J_s)} x_{(k,i,j_p,j,r)}, \quad \forall k \in K, (i,r) \in H \tag{5}$$

$$\sum_{(j\in J_s)}\sum_{(j'\in J_p)}x_{(k,i,j,j',r)}=\theta_{(k,i,r)},\quad \forall k\in K,(i,r)\in H \tag{6}$$

Constraints (5) and (6) ensure the path integrity of AGV.

$$\sum_{(k\in K)}\sum_{(j'\in J_p)}x_{(k,i,l_i,j',1)}=\lambda_{(i,1)},\quad \forall i\in I \tag{7}$$

$$\sum_{(k\in K)}\sum_{(j'\in J_p)}x_{(k,i,j,j',1)}=0,\quad \forall i\in I,j\in J_s,j\neq l_i \tag{8}$$

$$\sum_{(k\in K)}\sum_{(j'\in J_p)}x_{(k,i,j,j',r)}=\sum_{(k\in K)}\sum_{(j_p\in J_p)}x_{(k,i,j_p,j,r-1)},\quad \forall i\in I,r\in R_i,j\in J_s \tag{9}$$

Constraints (7), (8), and (9) are shelf position inheritance constraints that enable the model to correctly handle shelf reuse and position reallocation.

$$y_{(k,i,r,i',r')}+y_{(k,i',r',i,r)}\leqslant 1,\quad \forall k\in K,(i,r),(i',r')\in H,(i,r)\neq(i',r') \tag{10}$$

$$\sum_{(j\in J_s)}x_{(k,i,j,p_k,r)}+\sum_{(j\in J_s)}x_{(k,i',j,p_k,r')}-1\leqslant y_{(k,i,r,i',r')}+y_{(k,i',r',i,r)} \tag{11}$$

$$s_{(k,i',r')}\geqslant f_{(k,i,r)}-M_L\left(1-y_{(k,i,r,i',r')}\right),\quad \forall k\in K,(i,r)\neq(i',r'). \tag{12}$$

Constraints (10), (11), and (12) are constraints on the sequence of AGV tasks, ensuring the rationality of shelf visit order.

The second type of constraint is in the time dimension, using a set of time relationships to describe the AGV's first departure, pick-up to the picking area, picking operation, returning to the shelf station, and undertaking the next task, connecting the evolution of a “pick sort deliver” operation on the timeline.

$$s_{(k,i,l_i,r)}\geqslant\frac{1}{v_k}\sum_{(j\in J_s)}d_{(l_k,j)}x_{(k,i,j,p_k,r)}-M_L\sum_{((i',r')\neq(i,r))}y_{(k,i',r',i,r)}-M_L\left(1-\theta_{(k,i,r)}\right) \tag{13}$$

$$s_{(k,i,r)}\geqslant s_{(k,i,l_i,r)}+\sum_{(j\in J_s)}\frac{d_{(j,p_k)}}{v_k}x_{(k,i,j,p_k,r)}+t^L-M_L\left(1-\theta_{(k,i,r)}\right) \tag{14}$$

$$f_{(k,i,r)}\geqslant s_{(k,i,r)}+\sum_{(o\in O)}\sum_{(m\in M)}t_m^P w_{(i,o,m,r)}-M_L\left(1-\theta_{(k,i,r)}\right) \tag{15}$$

$$s_{(k,i,j,r)}\geqslant f_{(k,i,r)}+\frac{d_{(p_k,j)}}{v_k}x_{(k,i,p_k,j,r)}+t^U-M_L\left(1-x_{(k,i,p_k,j,r)}\right),\ \forall j\in J_s \tag{16}$$

$$s_{\left(k,i',l_{i'},r'\right)}\geqslant s_{(k,i,j,r)}+\frac{1}{v_k}\sum_{(j_s\in J_s)}d_{(j,j_s)}x_{(k,i',j_s,p_k,r')}-M_L\left(2-y_{(k,i,r,i',r')}-x_{(k,i,p_k,j,r)}\right)-M_L\left(1-\theta_{(k,i',r')}\right) \tag{17}$$

The third type of constraint requires from the perspective of material balance that the material requirements of each order are met, and the cumulative picking quantity across multiple visits to the same shelf does not exceed the initial inventory.

$$\sum_{(i\in I)}\sum_{(r\in R_i)}w_{(i,o,m,r)}\geqslant q_{(o,m)},\quad \forall o\in O,m\in M \tag{18}$$

$$\sum_{(o\in O)}\sum_{(r\in R_i)}w_{(i,o,m,r)}\leqslant q_{(i,m)},\quad \forall i\in I,m\in M \tag{19}$$

Constraints (18) and (19) ensure that the total amount of materials on the shelves meets the requirements of material picking orders, and that the sum of picking quantities in all visits on a single shelf does not exceed its initial inventory.

$$w_{(i,o,m,r)}\leqslant q_{(i,m)}\gamma_{(i,o,r)},\quad \forall i\in I,o\in O,m\in M,r\in R_i \tag{20}$$

$$\gamma_{(i,o,r)}\leqslant\lambda_{(i,r)},\quad \forall i\in I,o\in O,r\in R_i \tag{21}$$

$$\gamma_{(i,o,r)}+\theta_{(k,i,r)}\leqslant 1,\quad \forall k,o,(i,r),p_k\neq p_o \tag{22}$$

Constraints (20), (21), and (22) ensure a consistent relationship between material AGV visit and picking stations.

$$\rho_{(i,m,0)}=q_{(i,m)},\quad \forall i\in I,m\in M \tag{23}$$

$$\rho_{(i,m,r)}\leqslant\rho_{(i,m,r-1)}-\sum_{(o\in O)}w_{(i,o,m,r)}+M_L\left(1-\lambda_{(i,r)}\right),\quad \forall i\in I,m\in M,r\in R_i \tag{24}$$

$$\rho_{(i,m,r)}\geqslant\rho_{(i,m,r-1)}-\sum_{(o\in O)}w_{(i,o,m,r)}-M_L\left(1-\lambda_{(i,r)}\right),\quad \forall i\in I,m\in M,r\in R_i \tag{25}$$

$$\rho_{(i,m,r)}\leqslant\rho_{(i,m,r-1)}+M_L\lambda_{(i,r)},\quad \forall i\in I,m\in M,r\in R_i \tag{26}$$

$$\rho_{(i,m,r)}\geqslant\rho_{(i,m,r-1)}-M_L\lambda_{(i,r)},\quad \forall i\in I,m\in M,r\in R_i \tag{27}$$

$$\sum_{(o\in O)}w_{(i,o,m,r)}\leqslant\rho_{(i,m,r-1)},\quad \forall i\in I,m\in M,r\in R_i \tag{28}$$

$$\sum_{(o\in O)}w_{(i,o,m,r)}\leqslant M_L\lambda_{(i,r)},\quad \forall i\in I,m\in M,r\in R_i \tag{29}$$

$$\rho_{(i,m,r)}\geqslant 0,\quad \forall i\in I,m\in M,r\in R_i\cup 0. \tag{30}$$

Constraints (23) - (30) ensure the rationality of updating the material quantity after each AGV visit picking.

The fourth type of constraint relates the order completion time to the visit completion time and order deadline for which it serves:

$$C_o \geqslant f_{(k,i,r)} - M_L\left(1-\gamma_{(i,o,r)}\right), \quad \forall o \in O, k \in K, (i,r) \in H \tag{31}$$

$$C_o \leqslant d_o, \quad \forall o \in O. \tag{32}$$

As long as a certain visit actually provides picking services for order o, the picking completion time of that visit will be counted as a candidate value for the order completion time, and ultimately C_o will be equal to the latest one among all related visits. It is required that the completion time of the order should not exceed the given deadline d_o, otherwise it constitutes an infeasible solution.

The fifth type of constraint is oriented towards workstation resources.

$$\sum\nolimits_{(k \in K)} s_{(k,i,j,r)} \geqslant \sum\nolimits_{(k' \in K)} f_{(k',i',r')} - M_L\left(3 - \delta_{(i,r,i',r')} - \sum\nolimits_{(k'' \in K)} x_{\left(k'',i,p_{\left(k''\right)},j,r\right)} - \sum\nolimits_{(k'' \in K)} x_{\left(k'',i',p_{\left(k''\right)},j,r'\right)}\right) \tag{33}$$

$$\sum\nolimits_{(k \in K)} s_{(k,i,j,r)} \geqslant \sum\nolimits_{(k' \in K)} f_{(k',i',r')} - M_L\left(2 + \delta_{(i,r,i',r')} - \sum\nolimits_{(k'' \in K)} x_{\left(k'',i,p_{\left(k''\right)},j,r\right)} - \sum\nolimits_{(k'' \in K)} x_{\left(k'',i',p_{\left(k''\right)},j,r'\right)}\right) \tag{34}$$

Constraints (33) and (34) ensure that when the shelves are returned to the same shelf workstation j during two visits, the two return operations are forced to occupy the workstation in an orderly manner in terms of time, avoiding the situation where two shelves stop at the same shelf workstation at the same time.

$$s_{(k,i',r')} \geqslant f_{(k,i,r)} - M_L\left(2+\delta_{(i,r,i',r')} - \theta_{(k,i,r)} - \theta_{(k,i',r')}\right), \ \forall k \in K, (i,r),(i',r') \in H, (i,r) \neq (i',r') \tag{35}$$

$$s_{(k,i,r)} \geqslant f_{(k,i',r')} - M_L\left(\left(1-\delta_{(i,r,i',r')}\right) + \left(2-\theta_{(k,i,r)} - \theta_{(k,i',r')}\right)\right), \ \forall k \in K, (i,r),(i',r') \in H, (i,r) \neq (i',r') \tag{36}$$

Constraints (35) and (36) ensure that visit to shared picking stations on the same AGV is subject to temporal constraints, such that any picking station is occupied by only one shelf at any given time.

$$\delta_{(i,r,i',r')} \leqslant \lambda_{(i,r)}, \quad \forall (i,r),(i',r') \in H \tag{37}$$

$$\delta_{(i,r,i',r')} \leqslant \lambda_{(i',r')}, \quad \forall (i,r),(i',r') \in H \tag{38}$$

$$\delta_{(i,r,i',r')} + \delta_{(i',r',i,r)} \geqslant \lambda_{(i,r)} + \lambda_{(i',r')} - 1, \quad \forall (i,r),(i',r') \in H \tag{39}$$

Constraints (37) - (39) link mutually exclusive variables with visit activation states, ensuring that only when both visits are activated, a unique sequence needs to be selected between them.

The mathematical modeling of the problem has been completed above, and a mixed integer programming model has been established to accurately describe the three resource collaborative scheduling problem of shelf AGV location. This model achieves collaborative scheduling of three types of resources through refined decision variable design, ensuring the feasibility and rationality of scheduling from multiple levels such as resource allocation, path traffic, task sequence, time calculation, picking operations, and spatial capacity.

4. Algorithm Design

In the previous section, a mixed integer programming model was established for the shelf AGV location collaborative scheduling problem, which has NP hard complexity characteristics. For medium to large-scale cases, directly solving the mathematical model is often difficult to obtain satisfactory solutions in an acceptable time. In this section, an improved hybrid genetic algorithm is designed to approximate the problem.

4.1Hybrid Genetic Algorithm Framework

The basic idea of hybrid genetic algorithm is to adopt a three-stage strategy of "probabilistic greedy construction local search optimization genetic evolution". Its core feature is the iterative mode of "crossover optimization": after each crossover operation generates offspring, local search optimization is immediately performed on them, allowing the algorithm to explore new solution space regions while finely optimizing the quality of solutions.The algorithm framework is shown in Fig. 2.

The specific execution process of the algorithm includes: (1) initialization stage, using probabilistic greedy algorithm to generate the initial population, which balances the quality and diversity of solutions through greedy score evaluation and probability selection mechanism, and performs local search optimization on each initial individual; (2) In the evolutionary stage, a roulette wheel mechanism is used to select parent individuals, and gene recombination is carried out through four different granularity crossover operators: order level, AGV level, task level, and uniform crossover. Five neighborhood search operators (including task exchange within a single AGV path, task relocation between multiple AGV paths, task migration between multiple AGV paths, workstation and shelf visit exchange, and shelf placement replacement) are immediately applied to the generated offspring for local optimization, and a mixed mutation strategy (combining guided mutation and random mutation) is used to maintain population diversity; (3) Iterate the above evolutionary process until the maximum number of iterations is reached, and output the global optimal solution.

Below are several core components: encoding of solutions and construction of initial solutions, crossover operators, and neighborhood search operators.

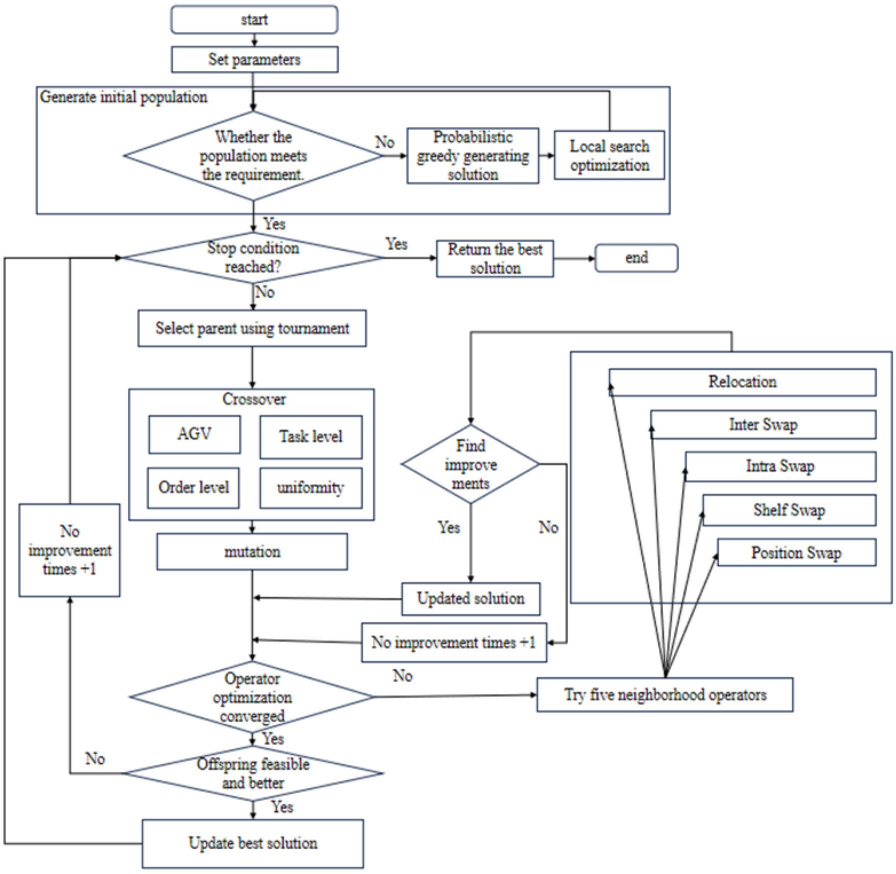

Fig. 2. Hybrid Genetic algorithm framework

4.2Encoding method and initial solution construction

This article defines the "pick sort deliver" task unit based on the targeted path structure of AGV, which serves as a gene locus in the genetic algorithm chromosome coding. The path of each AGV can be viewed as a chromosome composed of multiple 'task' genes. The paths of multiple AGVs form an individual solution.

Each AGV needs to perform the following tasks when delivering materials to a shelf: go to a workstation where the shelf is placed to retrieve the shelf, move the shelf to the picking area for picking personnel to pick up the goods, and then move the shelf to an empty workstation where there are no shelves stored. As shown in Fig. 3.

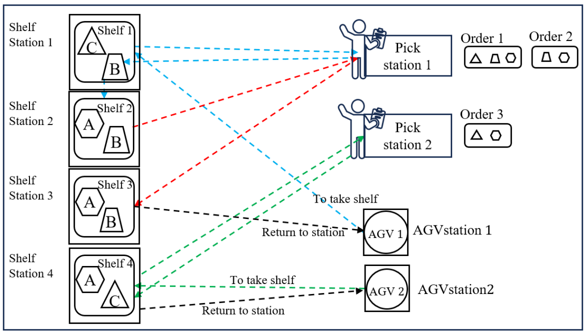

Fig. 3. Schematic diagram of AGVs executing tasks

Define a complete 'pick sort-deliver' task unit. Task 1 (blue dotted line):AGV 1 go to shelf station 1 to take shelf 1 → go to the picking station to unload materials B C → go to shelf station 1 to put shelf 1; Task 2 (red dotted line):AGV 1 go to station 2 to take shelf 2 → go to the picking station to unload materials A and B → go to shelf station 3 to put shelf 2; Task 3 (green dotted line):AGV 2 go to shelf station 4 to take shelf 4 →go to the picking station to unload materials A, C → go to station 4 to put shelf 3. The action process of an AGV is to stand by at the (current/parking area) station, then (reciprocate) perform some (multiple) tasks, and finally return to the parking area station (black dotted line). The path of a single AGV can be encoded as a task sequence consisting of [AGV parking space, task 1, task 2,..., AGV parking space].

After establishing the encoding method for the solution, the probability greedy algorithm is used to gradually add task units to each AGV to achieve AGV path construction. The pseudocode of the probability greedy algorithm is shown in Table 3.

Table 3
Probability Greed Strategy

```
Algorithm 1: Probability Greedy
Input: Parameters and initial state of the system
Output: Return AGV path and shelf, workstation, order status
Initialization status: Initialize AGV, shelves, orders, and workstation status
While (all orders have not been picked yet){
    Traverse all combinations: check each (AGV, task) combination, enumerate and select AGV and selectable tasks
    Calculate the time it takes for AGV to complete picking within the current task after adding each task
    Check feasible combinations as candidate combination list
    Calculate the greedy score for each combination
    Choose a combination of roulette wheel according to the probability calculation formula
    Execution (AGV, task) combination: Assign the optimal task to the AGV
    Update system status: update AGV, shelf, workstation, and order status
}
Return all AGVs to their initial parking spaces
```

The algorithm uses probabilistic greedy selection to choose the next action to be executed, denoted by the following symbols: the current available AGV index is k, the set of available shelves is S, each available shelf index is s, the position of shelf s is l_s, the set of materials on shelf s is M_s, the maximum quantity of each material m on shelf s that can meet the current order is q_m^s, the deadline for the current order is T_o, the arrival time of AGV at the current position is t_k, the time to go to workstation l_s is t_{l_s}, and the unit picking time is ψ.

The greedy score f_{ks} for selecting the k-th AGV to transport shelf s is:

$$f_{ks}=\frac{(T_o-t_k)\sum_{(m\in M_s)}q_m^s}{t_k+t_{(l_s)}+\psi\sum_{(m\in M_s)}q_m^s} \tag{40}$$

The numerator of formula (40) represents the amount of time required to carry out the current task and reserve it for subsequent tasks, while the denominator represents the time it takes for AGV k to transport shelves s to the picking area and complete the picking process. Use the following formula to select the next decision combination (AGV, task).

$$p_{ks}=\begin{cases} max_{(k\in K,s\in S)}f_{ks}, & r\leq r_0 \\ f_{ks}/\sum f_{ks}, & r>r_0 \end{cases} \tag{41}$$

Among them, r_0 is a greedy threshold parameter within the range of [0,1], and r is a temporarily generated random number within the range of [0,1]. If the random number is less than the threshold, the current optimal task is selected. Otherwise, the roulette wheel is used to select the next stage of tasks.

4.3 Crossover operator and neighborhood search operator

Due to the fact that the encoding of this problem involves multiple levels such as task allocation, execution order, and resource selection, a single cross method is difficult to balance the inheritance and diversity of solutions. For this purpose, this algorithm has designed four different granularity crossover operators: order level crossover, AGV level crossover, task level crossover, and uniform crossover. Coarse grained operators (order level, AGV level) exchange on larger structural units, resulting in offspring with minimal differences from the parent, which is beneficial for preserving high-quality structures in the parent; Fine grained operators (task level, uniform) exchange on smaller units, resulting in significant differences between offspring and parents, which is beneficial for exploring new solution structures.

When crossing, a weight based random selection mechanism is used to determine which operator to use. Coarse grained operators are given higher selection weights because their offspring are usually more feasible; Fine grained operators are assigned lower selection weights to control the proportion of infeasible solutions generated. After the crossover is completed, the algorithm conducts feasibility verification on the offspring, checks the completion status of orders and task time sequence, and filters out individuals that do not meet the constraint conditions.

This article designs and implements five different types of neighborhood search operators based on the structure of the problem: task exchange operator within a single AGV path, task exchange operator between multiple AGV paths, task relocation operator between multiple AGV paths, workstation and shelf visit exchange operator, and shelf placement replacement operator.

The task exchange operator in a single AGV path is a basic local search operator, and its operation scope is limited to a single AGV. The operator re optimizes the task sequence of the AGV by exchanging the execution sequence of two tasks on the same AGV, so as to reduce the total running time and waiting time. As shown in Fig. 4.

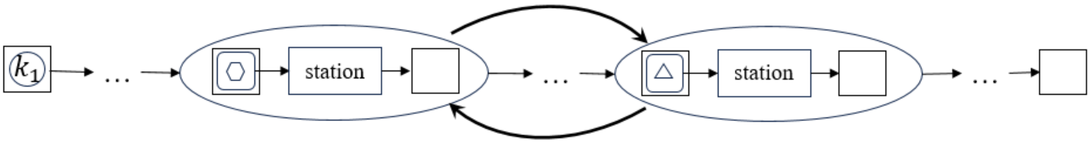

Fig. 4. Schematic diagram of task exchange operator within a single AGV path

The task exchange operator between multiple AGV paths extends the optimization range from a single AGV to multiple AGVs. Each of the two AGVs selects a task to exchange, so as to change which AGV will perform and the internal sequence of the two AGVs. The purpose of interchange is to make the task closer to the vehicle location, reduce unnecessary movement, and minimize the maximum completion time on the premise of meeting the constraints. As shown in Figure 5.

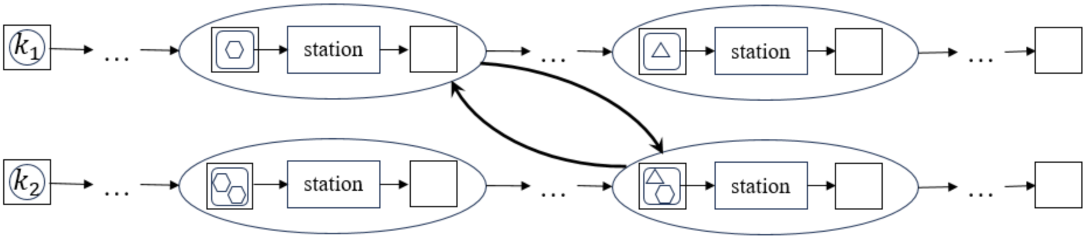

Fig. 5. Schematic diagram of task exchange operators between multiple AGV paths

The task relocation operator between multiple AGV paths transfers a task from one AGV to another AGV, and selects the insertion position in the task sequence of the receiver. Unlike task exchange, relocation will change the number of tasks undertaken by each AGV, which is more suitable for use when the task allocation is obviously uneven or the critical path is determined by a single vehicle. As shown in Figure 6.

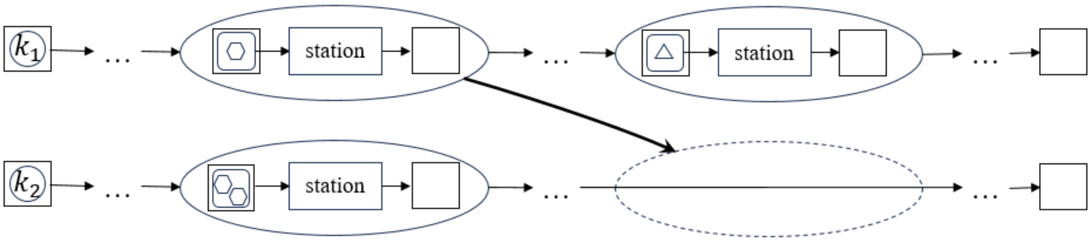

Fig. 6. Schematic diagram of task relocation operator between multiple AGV paths

The workstation and shelf visit exchange operator is oriented towards a single task, selecting new visit objects from multiple shelves that can provide the same material. The goal is to shorten the time or distance corresponding to the three paths from AGV to shelf, from shelf to picking area, and from picking area to destination. As shown in Figure 7.

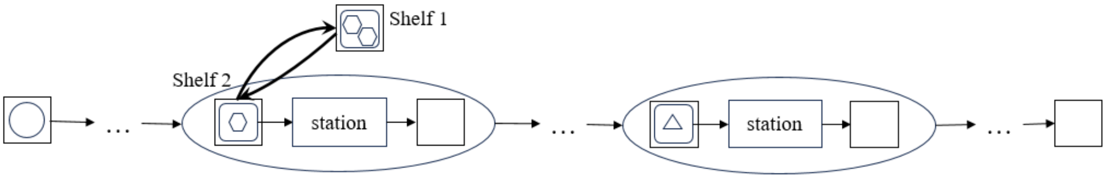

Fig. 7. Schematic diagram of workstation and shelf visit exchange operator

The shelf placement replacement operator reselects the position of the shelf after the task is completed. On the premise of satisfying constraints such as unoccupied and type matching, compare the differences in subsequent paths caused by placing different workstations back, and prioritize selecting the workstation closer to the next relevant pickup or handling location. As shown in Fig. 8.

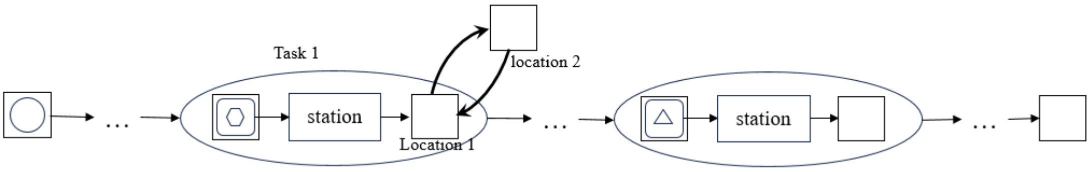

Fig. 8. Schematic diagram of shelf location replacement operator

In order to balance the improvement range and computational cost, local search is organized in order from coarse to fine. Firstly, adjusting across AGVs, including task relocation and task exchange, often results in a faster reduction of the objective function value. Subsequently, resource selection replacement is carried out, which involves exchanging visit between workstations and shelves, to better match the visit objects and spatial locations of individual tasks. Finally, the sequence and placement positions within a single AGV are refined, including task exchange and shelf replacement within a single AGV path, to continue improving the current solution with lower computational costs.

5. Experimental Analysis

To verify the performance of the designed algorithm in solving the collaborative scheduling problem of shelf AGV location resources, experiments were conducted on a general-purpose personal computer. In terms of hardware, the processor uses AMD Ryzen 9 7950X 16 Core Processor 4.50 GHz. 16GB of memory, operating system Windows 11. Implement and run algorithms using Python version 3.12 programming.

5.1 instances generation

This study used a structured random generation strategy to construct test cases, referring to Yang et al.'s case generation method (Yang et al. 2021). By setting different random seeds, the diversity and representativeness of the generated cases were ensured. To ensure the rationality and authenticity of the examples, the generation of examples follows mathematical layout rules and resource allocation principles., and each example includes warehouse layout (workstation position, type, and occupancy status), AGV information (initial position and quantity), shelf information (position, number, and type and quantity of stored materials), order information (order number, required material type and quantity, deadline, etc.), and material information (material number and shelf location).

In terms of warehouse layout generation, the total number of storage workstations $|J_s|$ is determined based on the size of the case study. Small, medium, and large-scale cases are configured with 32, 72, and 128 storage workstations, respectively. The storage area adopts a modular layout based on workstation groups, with each group containing 4 workstations arranged horizontally, and the number of workstation groups $G=\lceil |J_s|/4 \rceil$. To achieve a compact and balanced layout, the workstation groups are arranged in an approximately square shape, with a column count of $G_c = \lceil \sqrt{G} \rceil$ and a row count of $G_r = \lceil G/G_c \rceil$. The picking platform and AGV parking space are arranged longitudinally on the right side of the storage area, with a quantity relationship of $|J_p| = |J_a| = |K|$, achieving one-to-one binding between AGV and picking station. In terms of resource allocation ratio, based on actual RMFS system operation experience, the shelf to AGV configuration ratio $R_{|I|/|K|}$ for different scale cases is approximately 3.5, 5.0, and 6.5 in small, medium, and large-scale cases, respectively. The order to AGV configuration ratio $R_{|O|/|K|}$ is approximately 1.4, 2.0, and 2.5, respectively. The number of material types is determined based on the complexity of the case and actual demand. For small-scale cases, there are 4-8 types, for medium scale cases, there are 18-24 types, and for large-scale cases, there are 37-52 types. To ensure that order demands are met, a demand driven inventory allocation strategy is adopted, with a total inventory level of material m of $I_{m,total} = 1.5 \times D_{m,total} + 10$, where $D_{m,total}$ is the total demand for material m from all orders, a coefficient of 1.5 provides sufficient margin, and a constant of 10 is a safety buffer. The parameter settings for each scale instance are shown in Table 4.

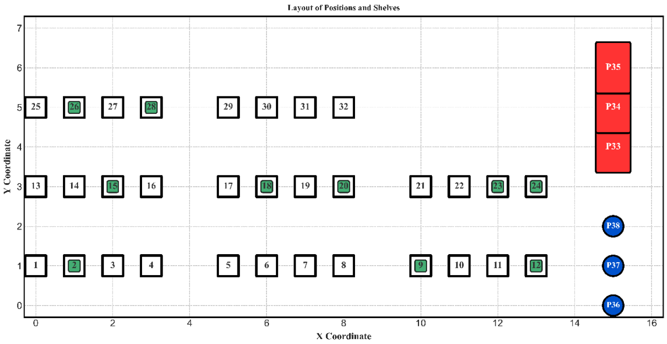

Fig. 9. Schematic diagram of parameter layout for small-scale examples

Fig. 9 shows a schematic diagram of a warehouse layout generated from a small-scale instances, where boxes represent workstations, green rounded rectangles represent shelves, blue circles represent AGV parking stations, and red rectangles represent picking areas.

Table 4
Parameter settings for different scale instances

Instance	AGV quantity	shelf quantity	Material types	order quantity
1	3	10	4	4
2	5	17	7	4
3	6	21	8	4
4	6	21	8	5
5	6	21	8	6
6	9	45	18	9
7	10	50	20	10
8	11	55	22	11
9	12	60	24	12
10	12	60	24	14
11	15	97	37	17
12	18	117	45	20
13	19	123	47	21
14	20	128	50	23
15	21	128	52	24

Note: instances 1-5 are small-scale, instances 7-10 are medium-scale, and instances 11-15 are large-scale.

The relevant parameter settings for the experiment are shown in Table 5.

Table 5
Parameter settings for different scale instances

Parameter Category	Parameter Name	Symbol	Value
Genetic algorithm	population size	G	20
	Maximum Evolutionary Algebra	T_{max}	50
	Cross probability	ρ	0.8
	mutation probability	σ	0
Probability greedy	Greedy threshold	r_0	5
the math problem	AGV moving speed	v_k	1.3m/s
	Pick time	ψ	10s/unit
	Shelf loading time	ω	15s
	Shelf unloading time	ζ	15s

Multiple comparative algorithms were set up in the experiment to comprehensively evaluate the performance advantages of the algorithms used. The first comparative algorithm is the greedy algorithm, which is based on a probabilistic greedy strategy and has the characteristic of fast computation speed. It is suitable for generating initial solutions quickly and serves as a benchmark algorithm to evaluate the improvement effect of other algorithms. The second comparative algorithm is the greedy local search algorithm, which optimizes the initial solution generated by the greedy algorithm by applying local search to improve the quality of the solution. The algorithm has a moderate computation time and is used to evaluate the effectiveness of local search in problems.

5.2 Analysis of experimental results

Table 6 summarizes the overall performance of the three algorithms on 15 test cases. The improvement rate in the table represents the performance of the proposed algorithm relative to the greedy algorithm for each corresponding instance.

Table 6
Experimental results of three algorithms

	Greedy		*Greedy +local search*			*hybrid genetic algorithm*		
Instance	Makespan (s)	Solution time (s)	Makespan (s)	Solution Time (s)	Relative improvement (%)	Makespan (s)	Solution Time (s)	Relative improvement (%)
1	212.05	0.00	206.09	0.02	2.81	204.71	1.15	3.46
2	476.81	0.00	375.06	0.08	21.34	300.35	0.38	37.01
3	161.51	0.00	161.51	0.03	0.00	127.71	0.42	20.93
4	410.65	0.00	400.91	0.08	2.37	272.64	0.42	33.61
5	328.81	0.00	328.81	0.04	0.00	286.16	0.34	12.97
6	807.35	0.01	807.35	0.30	0.00	749.49	0.87	7.17
7	988.29	0.02	981.82	1.10	0.65	716.48	1.86	27.5
8	943.21	0.02	929.34	0.94	1.47	752.12	1.95	20.26
9	980.99	0.02	960.13	1.15	2.13	746.07	1.79	23.95
10	1113.79	0.01	1067.57	0.63	4.15	735.62	1.31	33.95
11	1201.55	0.02	1189.82	1.33	0.98	1115.58	2.69	7.16
12	1088.35	0.03	1085.55	1.56	0.26	946.21	2.81	13.06
13	1137.8	0.03	1118.3	2.03	1.71	1075.91	2.84	5.44
14	1481.29	0.03	1477.8	2.21	0.24	1230.43	2.69	16.94
15	1408.05	0.07	1378.65	4.07	2.09	1090.68	6.11	22.54

Hybrid genetic algorithm achieves the minimum maximum completion time in all cases. From the perspective of computational efficiency, there is a significant difference in the average solution time among the three algorithms. The greedy algorithm is almost instantaneous, with an average solution time of only 0.003 seconds; The average solving time of the greedy local search algorithm is 0.20 seconds, which increases the computation time by about 67 times compared to the greedy algorithm, but still remains within the second range; The average solution time of the hybrid genetic algorithm is 3.21 seconds. Although it takes longer than the first two algorithms, considering its significant improvement in solution quality, this time cost is acceptable. Especially in small-scale examples (Examples 1-5), the average solution time of the hybrid genetic algorithm is only 0.59 seconds, while in large-scale examples (Examples 11-15), the average solution time is 6.81 seconds, reflecting the reasonable relationship between algorithm complexity and problem size.

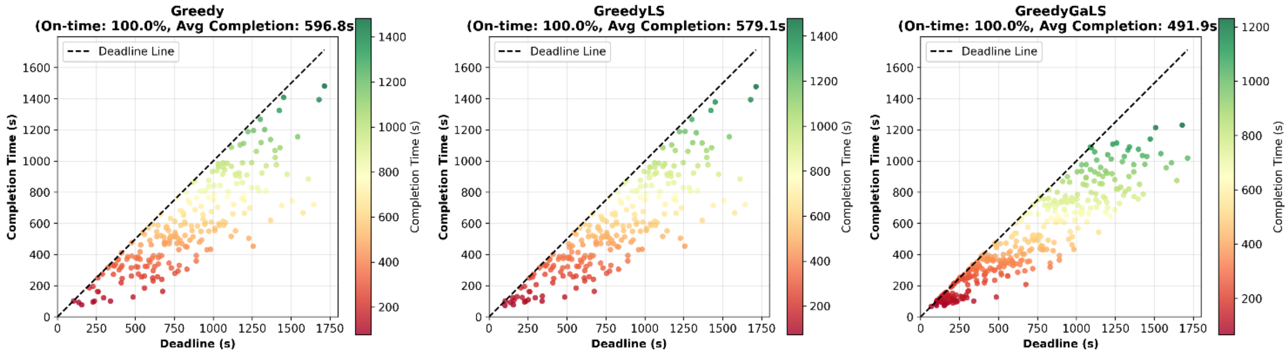

Fig. 10. Summary chart of order completion time analysis of the instances

Fig. 10 shows the order completion time analysis summary of all instances (grouped by algorithm). The results of the three algorithms are plotted in three subgraphs, and each subgraph shows the relationship between the completion time and the deadline of each order in all 15 instances. In the figure, the horizontal axis is the order deadline, the vertical axis is the actual completion time, the red dotted line is the deadline baseline (completion time=deadline), and the scattered color indicates the degree of completion ahead of schedule (green indicates significant advance, and red indicates approaching deadline). The three algorithms achieve 100% on-time rate in all cases, and all orders are completed before the deadline. However, the hybrid genetic algorithm makes the distribution of order completion time more compact and reasonable through better task scheduling, which verifies the advantages of hybrid genetic algorithm in optimizing the overall performance of the system from the order level.

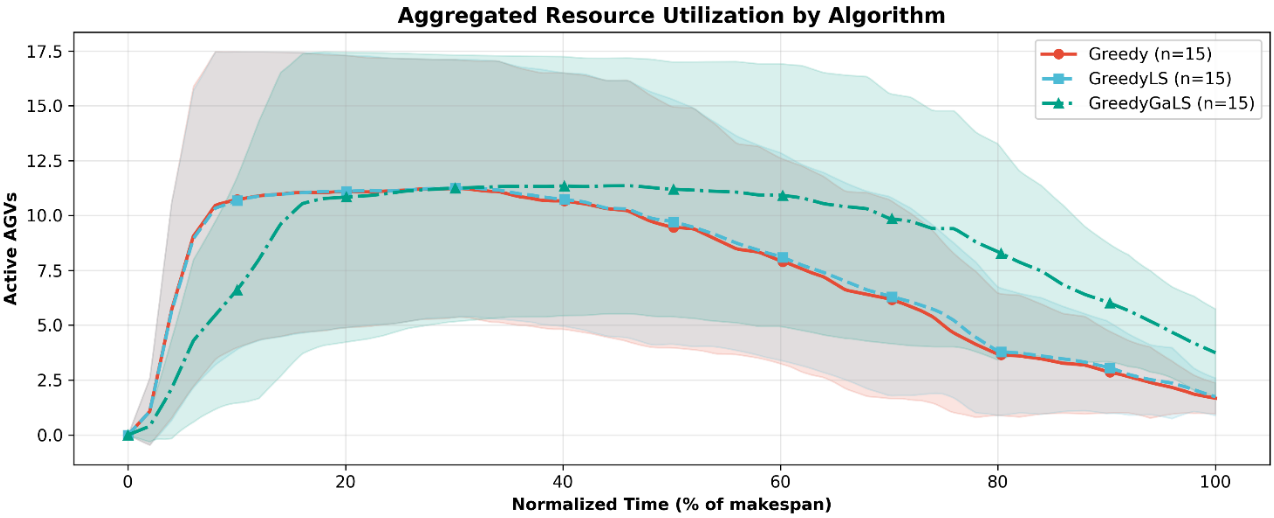

Fig. 11. Summary chart of order completion time analysis of the instances

Fig. 11 shows the summary analysis of AGV resource utilization of all examples (grouped by algorithm). In this figure, the average curve of AGV activity over time of the three algorithms on all 15 examples is drawn in the same figure for visual comparison. In the figure, the horizontal axis is the normalized time (percentage of the maximum completion time), the vertical axis is the number of AGVs working at the same time, the solid line represents the average value, and the shaded area represents the range of standard deviation. In the whole scheduling cycle of hybrid genetic algorithm, the number of active AGVs remains at a high level, and the average number of active AGVs is the largest. In order to more intuitively show the optimization effect of the algorithm, small-scale instance 1 is selected as a representative example for visual analysis. Fig. 12 is the AGV path direction of instance 1 under the hybrid genetic algorithm, and Fig. 13 is the Gantt chart of the running time of instance 1 AGV under the hybrid genetic algorithm.

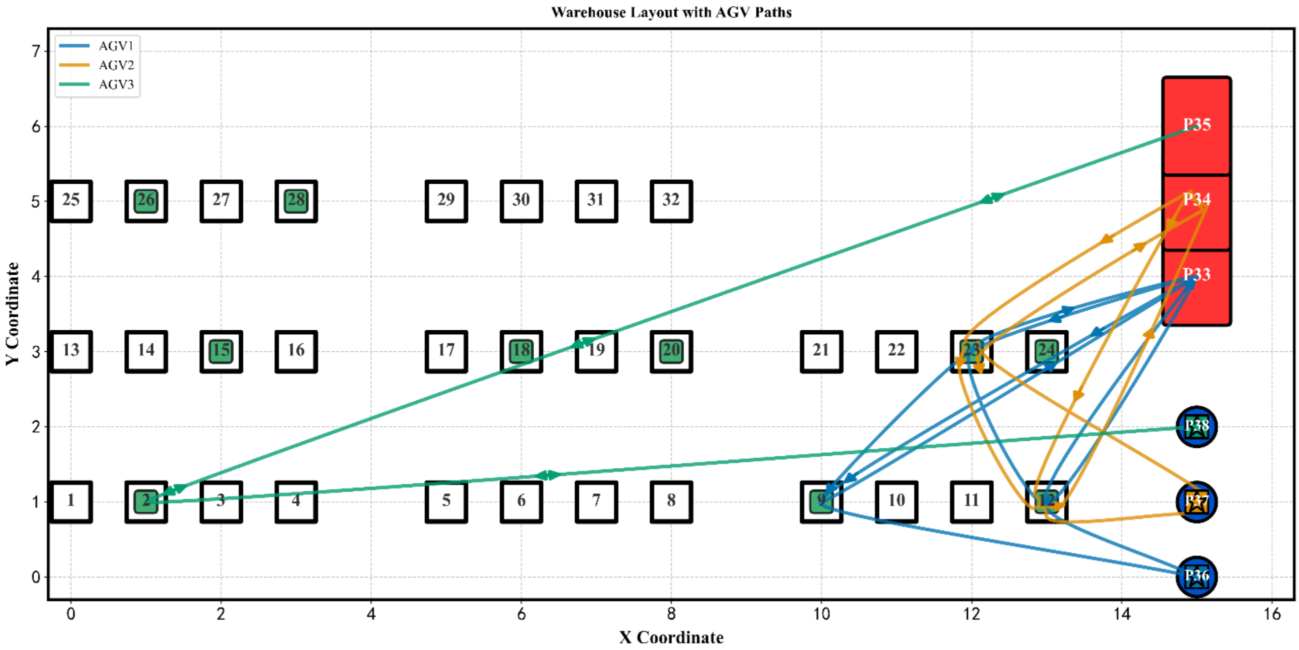

Fig. 12. instance 1 AGV path direction under hybrid genetic algorithm

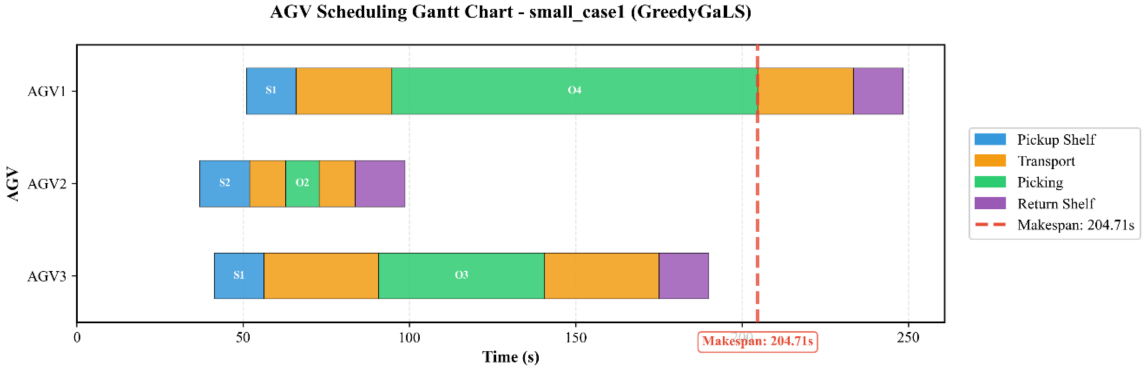

Fig. 13. Gantt chart of AGVs time under instance 1 hybrid genetic algorithm

Fig. 12 is the AGV path planning result of hybrid genetic algorithm on small-scale example 1. In order to avoid path overlap caused by repeated route tracks, a curved polyline with arrows between two points is used to represent the path, and the arrows represent the direction of motion. It can be seen from the figure that the hybrid genetic algorithm can effectively plan the operation path of AGV, avoid unnecessary detour, and realize efficient shelf handling. The task allocation of the three AGVs is relatively balanced, and the paths are less crossed, which reflects the good performance of the algorithm in path optimization and task allocation.

Fig. 13 is the Gantt chart of hybrid genetic algorithm on small scale example 1. The Gantt chart clearly shows the task execution of each AGV on the time axis, including picking shelves, moving to stations, picking, and putting shelves back. It can be seen from the figure that the hybrid genetic algorithm can effectively arrange the task sequence, reduce the idle time of AGV and improve the resource utilization. The maximum completion time is 204.71, which is 3.46% lower than 212.05 of greedy algorithm, which fully reflects the optimization effect of the algorithm.

6. Conclusions

This paper focuses on the integrated collaborative scheduling problem of three kinds of resources in the mobile shelf storage system, i.e. shelf, AGV and storage location. The modeling, algorithm and example analysis are carried out in the static order environment. Compared with splitting the decision into several stages or optimizing only for a single resource, this paper attempts to consider the decisions of order allocation, shelf access, station selection and AGV path under a unified framework, which provides a relatively complete description idea for analyzing the RMFs scheduling problem with multi resource coupling.

In terms of solving methods, aiming at the NP hard characteristics of this kind of problems, this paper designs a three-stage hybrid genetic algorithm framework of "probabilistic greedy construction local search optimization genetic evolution". The algorithm uses the chromosome coding method based on the task unit, represents a feasible solution by the task sequence of multiple AGVs, generates the initial population combined with the probabilistic greedy algorithm, and adds local search in the initial stage to improve the solution quality; In the evolution stage, multi granularity crossover operators such as order level, AGV level, task level and uniform crossover, as well as several neighborhood search operators around the task unit are designed to adjust the task allocation and order on the premise of maintaining feasibility, so as to achieve a balance between global search and local improvement.

In general, the main work of this paper includes: constructing an integrated scheduling model for shelf AGV storage space, and introducing the "pick place" task unit to describe the relationship between path and time in a unified way; A hybrid genetic algorithm combining probabilistic greedy, local search and genetic evolution is designed, and a coding method and a variety of crossover and neighborhood operators that fit the task unit structure are proposed; Through the comparative analysis of several groups of examples, the solution effect of the proposed method in the given test environment is verified. It should be noted that this paper is still based on the assumptions of static orders, deterministic parameters and non explicit modeling of AGV collision, and the optimization goal is mainly focused on the single index of maximum completion time. These settings simplify the problem complexity to a certain extent, and also constitute the applicable premise of the research conclusion.

Acknowledgments

This work was supported by the Natural Science Foundation of Zhejiang Province (Grant number LMS26E050028). All authors declare that he has no other conflict of interest.

References

Dai, M., Tang, D., Giret, A., & Salido, M. A. (2019). Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. *Robotics and Computer-Integrated Manufacturing, 59*, 143–157.

Gharehgozli, A., & Zaerpour, N. (2020). Robot scheduling for pod retrieval in a robotic mobile fulfillment system. *Transportation Research Part E: Logistics and Transportation Review, 142*, Article 102087.

Hashemi, S., & Ranjbar, M. (2024). Optimal resource allocation and routing in robotic mobile fulfillment systems. *Computers & Operations Research, 165*, Article 106571.

Jiang, J., Fu, M., Tang, X., & Wei, Q. (2025). Research on storage allocation strategy for mobile shelf storage system [Mobile shelf storage system storage allocation strategy research]. *Mechanical and Electrical Engineering Technology, 54*(16), 85–90.

Jiang, M., Leung, K. H., Lyu, Z., & Huang, G. Q. (2020). Picking-replenishment synchronization for robotic forward-reserve warehouses. *Transportation Research Part E: Logistics and Transportation Review, 144*, Article 102138.

Justkowiak, J.-E., & Pesch, E. (2023). A column generation driven heuristic for order-scheduling and rack-sequencing in robotic mobile fulfillment systems. *Omega, 120,* Article 102897.

Kim, H.-J., Pais, C., & Shen, Z.-J. M. (2020). Item assignment problem in a robotic mobile fulfillment system. *IEEE Transactions on Automation Science and Engineering, 17*(4), 1854–1867.

Li, K., Liu, T., & He, B. (2022). Research on AGV path planning and scheduling in the "goods to people" picking system ["Goods to People" picking system AGV path planning and scheduling research]. *Chinese Management Science, 30*(4), 240–251.

Li, S., Zhao, Z., Wang, D., & Wang, H. (2025). A reinforcement learning framework for efficient task allocation among AGVs in smart warehouse. *IEEE Internet of Things Journal, 12*(11), 16947–16961.

Liu, A., Qi, S., Li, R., & Guo, X. (2024). Collaborative optimization of storage allocation and path planning for multi carrier robot access system [Multi-carrier robot access system storage allocation and path planning collaborative optimization]. *System Engineering Theory and Practice*, 1–11.

Lu, J., Qian, H., Zhao, W., & Zhang, J. (2024). Storage allocation problem of RMFS order picking system for manufacturing enterprises based on real-time replenishment [RMFS order picking system storage allocation problem for manufacturing enterprises based on real-time replenishment]. *Computer Integrated Manufacturing Systems, 30*(7), 2526–2539.

Rahman, H. F., & Nielsen, I. (2019). Scheduling automated transport vehicles for material distribution systems. *Applied Soft Computing, 82*, Article 105552.

Shi, X., Deng, F., Guo, M., & Shen, L. (2024). A novel fulfillment-focused simultaneous assignment method for large-scale order picking optimization problem in RMFS. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 54*(2), 1226–1238.

Tang, H., Huang, J., Ren, C., & Wang, G. (2025). Integrated scheduling of multi-objective lot-streaming hybrid flowshop with AGV based on deep reinforcement learning. *International Journal of Production Research, 63*(4), 1275–1303.

Valle, C. A., & Beasley, J. E. (2021). Order allocation, rack allocation and rack sequencing for pickers in a mobile rack environment. *Computers & Operations Research, 125*, Article 105090.

Xie, L., & Otten, S. (2024). Introducing combi-stations in robotic mobile fulfilment systems: A queueing-theory-based efficiency analysis. [Journal information missing - please add journal name, volume, and page numbers if available].

Yang, X., Hua, G., Hu, L., & Zhang, L. (2021). Joint optimization of order sequencing and rack scheduling in the robotic mobile fulfillment system. *Computers & Operations Research, 135*, Article 105467.

Yang, X., Hua, G., Zhang, L., & Cheng, T. C. E. (2025). Joint optimization of order- and rack-scheduling in KIVA picking systems. *Omega, 135,* Article 103286.

Yuan, Z., & Gong, Y. Y. (2017). Bot-in-time delivery for robotic mobile fulfillment systems. *IEEE Transactions on Engineering Management, 64*(1), 83–93.

Zhang, G. H., Hu, Y. F., Sun, J. H., & Zhang, L. (2020). An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints. *Swarm and Evolutionary Computation, 54*, Article 100664.

Zhuang, Y., Zhou, Y., Hassini, E., & Yuan, X. (2024). Improving order picking efficiency through storage assignment optimization in robotic mobile fulfillment systems. *European Journal of Operational Research, 316*(2), 718–732.

Zou, B., Gong, Y., Xu, X., & Yuan, Z. (2017). Assignment rules in robotic mobile fulfilment systems for online retailers. *International Journal of Production Research, 55*(20), 6175–6192.