Robotics and Computer–Integrated Manufacturing 79 (2023) 102397

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Full length article

MLATSO: A method for task scheduling optimization in multi-load AGVs-based systems

Yishuai Lin [a,*], Yunlong Xu [a], Jiawei Zhu [b], Xuhua Wang [a], Liang Wang [c], Gang Hu [a]

[a] The School of Computer Science and Technology, Xidian University, Xi'an, 710071, China
[b] The School of Information Engineering, Chang'an University, Xi'an, 710064, China
[c] Suzhou Mingyi Intelligence Warehousing Information Technology Co., Ltd., SuZhou, 215000, China

ARTICLE INFO

Keywords:
Multi-load AGV
Multi-objective optimization
Task scheduling
Automated storage and retrieval system

ABSTRACT

In the context of increasingly competitive intelligent manufacturing, the multi-load Automated guided vehicles (AGVs) based Automated Storage and Retrieval System (AS/RS) has been of particular interest, as reductions in the number of AGVs required can significantly decrease potential congestions and increase the system effectiveness. In comparison with the single-load AGVs system, more difficult and critical issue of scheduling multi-load AGVs to automate storage/retrieval missions and to maximize economic benefits remains unresolved. Therefore, we propose a task scheduling optimization method for multi-load AGVs-based systems, with which, the objectives of least number of occupied AGVs, shortest travel time and minimum conflicts can be met simultaneously. The experiments are conducted in various scenarios, and verify that our work can use fewer AGVs to optimize tasks delivery, which enables the AS/RS stakeholders to reach win-win results for system performance and AGVs investment, thus maximizing economic benefit.

1. Introduction

In the highly competitive intelligent manufacturing industry, the Automated Guided Vehicles (AGVs) based automated storage and retrieval system (AS/RS) has emerged as a competitive, efficient and reliable solution [1,2], in which, AGVs are used to execute storage and retrieval missions due to their end-to-end capabilities, more efficient performance of storage and retrieval, reduced errors and labor costs, etc. [3,4]. Particularly, in the special period to fight an all-out global battle against COVID-19, using AGVs instead of people to transport items in semi-enclosed spaces, such as a factory, a workshop and a warehouse, means reducing the risk of viral infection caused by the transport of freight [5].

As the development of AGVs-based AS/RSs, two kinds of AGVs have been used in AS/RSs, as the single-load AGV and the multi-load AGV [6]. The single-load AGV can take only one item or SKU (Stock Keeping Unit) at a time, and the multi-load AGV is able to pick up multiple different items from one station or multiple stations. The performance comparison between the single-load AGVs system and the multi-load AGVs system has been discussed deeply, and with analyzing experimental results, it is obviously observed that the application of multi-load AGVs can significantly decrease necessary required quantity of AGVs and related congestion, as well as improve the effectiveness of the system [7]. Thus, due to the growing AS/RS scale and growing requirement of system transportation efficiency, the application of multi-load AGVs has become an essentially essential trend.

Meanwhile, in order to maximize the efficiency of AGVs in AS/RSs, there have been many studies on AGVs-based AS/RSs, in which, the optimization of AGVs task scheduling is a very significant problem for stakeholders of AGVs-based AS/RSs, because this problem can directly determine the performance indicators of a AS/RS, such as the delivery time, the utilization rate of AGVs, etc., and indirectly influence the economic benefit, the energy consumption, the maintenance of equipments and so on.

Based on a review of related literature, it is apparent that most of them are focused on scheduling optimization of single-load AGVs based AS/RSs, such as minimizing penalized earliness and tardiness [8], controlling deadlocks [9] and minimizing makespan while considering the AGVs' battery charge [10]. Nevertheless, some progress has been achieved in optimizing AS/RSs based on multi-load AGVs. However, scheduling multi-load AGVs is a much more complicated and impact question than scheduling single-load AGVs, since there are more variables that must be determined in a system with multi-load AGVs, such as the number of assigned AGVs, the specific AGV assigned to the specific task, the load capacity of each assigned AGV, and the routing of each assigned AGV. Accordingly, the purpose of this paper is to analyze

* Corresponding author.
E-mail address: yslin@xidian.edu.cn (Y. Lin).

https://doi.org/10.1016/j.rcim.2022.102397
Received 24 January 2022; Received in revised form 5 May 2022; Accepted 17 June 2022
Available online 18 July 2022

and optimize the task scheduling for multi-load AGVs-based AS/RSs. Our contributions are summarized as follows.

- We establish a model that supports conceptualizing the multi-load AGVs-based AS/RS and analyzing its key performance indicators.
- We propose a task scheduling optimization method for the multi-load AGVs-based AS/RS, with which, multiple optimized objectives can be achieved, including AGV quantity, routing time and conflicts of AGVs. Specially, if there are not enough AGVs available, this method may yield a suboptimal solution with fewer AGVs. It is the first time to answer the question of task scheduling optimization of multi-load AGVs in AS/RSs.
- We provide a series of experiments executed in different scenarios of a multi-load AGVs-based AS/SR, in which, we consider incremental task sizes, changeable maximum load of AGVs and the number of available AGVs. In analyzing the experimental results and comparing our work with the prior work, we demonstrate that our work can optimize tasks delivery time with fewer AGVs, allowing AS/RS stakeholders to determine the most beneficial number of multi-load AGVs to use, ultimately improving system performance and minimizing AGV investment costs, thus maximizing economic profits.

The organization of this paper is as follows. In Section 2, we state the literature contribution of both single-load AGVs and multi-load AGVs system scheduling and analyze the necessity of our work. Based on the literature review, we specify the multi-load AGVs-based AS/RS in Section 3 and provide our designed model for analyzing and optimizing multi-load AGVs-based AS/RSs in Section 4. With this model, in Section 5, a task scheduling optimization method with objectives of simultaneously optimizing the number of AGVs, traveling time and multiple AGV conflicts is proposed. Moreover, in Section 6, there are related experiments to verify the effectiveness of our work in the system with abundant AGVs and limited AGVs and the impact of applying different load capacity AGVs on the system. Finally, we draw the conclusion of the paper in Section 7.

2. Related work

As we know, scheduling optimization of AGVs in the AS/RS is an extremely relevant question. There has been a lot of literature about scheduling optimization since the AGVs-based AS/RS came into service, in which, on the one hand, it is discussed more deeply that the scheduling optimization of the single-load AGVs system from the perspective of optimizing the delivery time, route planning, path conflict, and quantity configuration of the AGVs, etc. On the other hand, research carrying out on the multi-load AGVs system scheduling is scarce, and the advantage and potential benefits due to multi-load AGVs is not explored totally. In this paper, we review relevant literature on both the single-load AGVs system and the multi-load AGVs system, as well as other related scenarios involving the similar issue. Based on that, we dedicate ourselves to scheduling the multi-load AGVs system.

2.1. Single-load AGVs

Several publications have addressed the scheduling optimization of single-load AGVs, which consider factors such as optimizing delivery time, route planning, path conflicts, and quantity configuration of AGVs. For example, Zijian He et al. [11] developed a policy called differentiated probabilistic queuing for a single-load AGVs-based system, to reduce the average overall latency of each customer order, as well as the average total processing time for all orders belonging to the same customer. O.T. Baruwa et al. [12] demonstrated a timed colored Petri net that was able to dispatch AGVs and machines and outperformed conventional search algorithms. In [13], Nitish Singh et al. investigated the problem of scheduling AGVs with battery constraints and proposed a mixed-integer linear programming model to minimize a weighted sum of the tardiness costs of transport requests and travel costs of AGVs. The work above mainly focused on minimizing tasks completion time, which was as most of literature considering [14,15].

Additionally, other related important factors were optimized, in which, a number of studies [16–18] were conducted to the process of routing and the conflicts can be obtained in scheduling, in order to decrease the conflict and deadlock issues for single-load AGVs. The research by Yang et al. [19] proposed an AGVs scheduling algorithm based on dynamic time estimation, which reduced the start-stop frequency and reduced road network congestion. However, experimental results revealed a local optimal problem. Hao et al. [20] studied the dispatching and routing problem for AGV systems and proposed an AGV dispatching method based on the hybrid genetic algorithm, but the method can only be applied in the conflict-free condition. Zou et al. [21] addressed a new multiple AGV dispatching problem from the material handling process in a matrix manufacturing workshop, which aimed to determine a solution with the objective of minimizing the transportation cost, including travel cost, penalty cost for violating time and AGVs cost. However, the optimization of AGV quantity and the avoidance of conflict between AGVs were not taken into account.

There was also a group of researchers who tried to determine the optimum minimal number of AGVs used in AGV-based systems, but the method was designed under the assumption of free-conflict. For example, Vivaldini et al. [22] and Mousavi et al. [23] respectively investigated the method of estimating the optimal number of AGVs for AGV-based warehouses, which can reduce the number of vehicles and energy used, but the assumption of no traffic congestion was far away from the actual operation of the system. Yuxin Li et al. [24] proposed a hybrid deep Q network to minimize the makespan and total energy consumption, which concerned the limited number of AGVs.

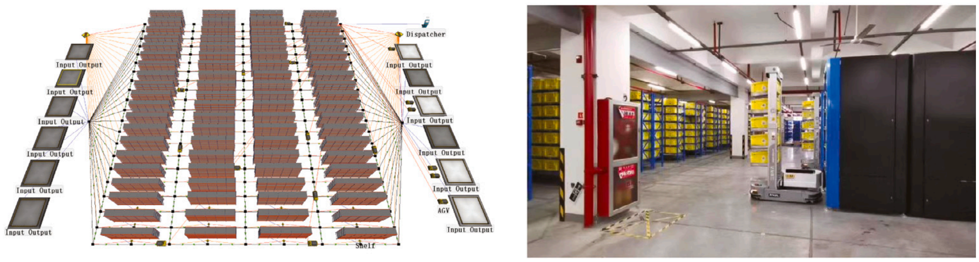
Reviewing this literature, we found that, because of the impact of path conflict and congestion, increasing the number of AGVs did not have a continuous positive effect on system performance. There was obviously exciting the influence of the marginal effect on the number of AGVs [25]. In other words, the transportation efficiency of the system will not continue to increase as the number of AGVs increases. Hence, it would be important to propose a scheduling method that can optimize the task completion time and the number of AGVs while considering path conflicts and congestions.

2.2. Multi-load AGVs

The use of multi-load AGVs in AS/RSs is relatively new compared with the use of single-load AGVs. Specifically, in order to understand how to control multiple multi-load AGVs to complete a task, the multi-load AGVs system has been simulated and analyzed as the first consideration [26,27].

Based on that, a method to determine an optimal load capacity of the multi-load AGV for a particular system was concerned in [28]. V.K. Chawla et al. [29] proposed an analytical model and a MMPSO (Modified Memetic Particle Swarm Optimization Algorithm) based method for the simultaneous scheduling of multi-load AGVs and their jobs with aims of minimizing the travel time and the waiting time of AGVs in the flexible manufacturing system, but every single route was planned separately in this method with no conflicts.

With reviewing literature above and other existing studies [22, 30,31], on the one hand, in the problem of scheduling optimization of multiple-load AGVs, a number of related elements were simplified or ignored, e.g. routes conflicts, the number of AGVs and the AGV load decisions, etc. On the other hand, compared with the single-load AGVs system, more elements need to be optimized in the multi-load AGVs system, and the optimal decision of these elements can have an extensive impact on the system's performance, energy consumption, or economic costs, etc.



(a) The simulation model (b) The actual enviorment

Fig. 1. A multi-load AGVs-based AS/RS.

In addition, there was some literature on vehicle routing problems, such as [32–34], which considered the routing time, the path length and other factors to schedule vehicles under particular scenarios, and provided useful references to our work.

As a result, in this paper, we firstly dictate to the model and the method for task scheduling optimization in the multi-load AGVs system with considering all related factors. With our work, for a given task, it can be intelligently decided that the minimal number of AGVs required, the optimal load and the route planned for each assigned AGV with estimated shortest delivery time and minimum conflicts. That is to say, our work can help system stakeholders complete tasks with the shortest possible delivery time using the most economically feasible resources.

3. System specification

An AS/RS using AGVs consists of several basic components, namely storage racks/shelves, input/output work centers, which are areas where AGVs can load and unload items, as well as charging stations for AGVs, and obstacles such as pillars and conveyor belts. As shown in Figs. 1(a) and 1(b), it is one AGV-based AS/RS with 20 rows and 20 columns of shelves. Precisely, there are 20 rows of shelves, and the width between each row can only pass through one AGV. In addition, each row contains 20 shelves, and 5 shelves are in a group, which are tightly packed without any gaps between them. There is a one-way channel between different groups of shelves for one AGV to pass. In addition, in this system, there are multiple AGVs distributed at random, as well as some pillars, conveyor belts, and charging stations. Furthermore, there are multi input/output points located on both sides of the AS/RS system, and these points could also be dynamically opened or closed as the business requirements change. In the experiment, we designed 4 output points located on the same side for working.

Moreover, we detail the process of AGVs executing outbound tasks, in order to explain how items can be delivered using multi-load AGVs in the AS/RS. Following the arrival of a batch of output tasks, the system requires the scheduling of particular AGVs to deliver items in accordance with the order and the path planning. Specifically, according to the results of the scheduling, one assigned AGV begins from its current location, travels to pick up every item from different shelves in the defined order and route, and finally delivers items to the output work center. After that, the AGV goes to an unoccupied location for waiting the next assignment. Furthermore, if the number of outbound tasks exceeds the maximum load capacity of all available AGVs, certain AGVs can be assigned to transport a portion of the items to the output work center, and then return to pick up the remainder. The process of AGVs executing inbound tasks is similar to that of outbound tasks; the only difference is that the items are transported from the work center to the shelf rather than from the shelf to the work center.

4. Model description

4.1. Assumptions and limitations

According to practical observations as well as academic studies for multi-load AGVs-based AS/RS, we make the following fundamental assumptions.

- The number of AGVs in the AS/RS is not limitless, so that the quantity of assigned AGVs cannot exceed current available AGVs.
- The decided load of each multi-load AGV cannot exceed its maximum load capacity.
- All AGVs are reliable and travel at a pre-determined constant average speed.
- All tasks have the same priority, that is to say, no task is required to be handled preferentially in the scheduling process.
- Every scheduled AGV departs from its current location at the same time and returns to a random free location after finishing its assigned tasks.

4.2. Model statement

Based on the system specification, we carry out mathematical modeling of tasks scheduling optimization, in order to demonstrate performance indicators and optimization objectives. Specifically, the locations of all components of the system can be expressed in two-dimensional coordinates, including the input/output point, the shelf, the AGV, the item to be transported, and other obstacles. S_i is the ith shelf, $S = \{S_i | i = 0, 1, \dots, X\}$ corresponds to all the shelves in the system. A_k is the kth AGV, $A = \{A_k | k = 1, 2, \dots, K\}$ is defined as the collection of multi-load AGVs. v refers to the speed of the AGV. L is the restraint load (maximum load) for the multi-load AGV in the system. W_j is the jth item required to be handled, $W = \{W_j | j = 1, 2, \dots, M\}$ is the collection of items required to be transported. K is the max available number of multi-load AGVs at the moment.

Furthermore, $N\left(A_k\right) = \{0, 1\}$, $k = 1, 2, \dots, K$ represents whether the AGV A_k is chosen to deliver items, the value of which is 1 means AGV A_k is selected, and the value of 0 is the opposite. $d_{W_i W_j}$ is the distance between the location of the item W_i and the location of the item W_j, which is the distance between the locations of two shelves where items are located, due to the fact that the item is placed on a shelf. Moreover, $C_{A_i A_j}$ means the number of conflicts between the AGV A_i and the AGV A_j during the transportation. $C_{A_i A_j t_u}$ refers to the conflicts between the AGV A_i and the AGV A_j at time t_u, and ts_{A_k} and te_{A_k} refers to the start time and the end time of AGV A_k. The $p_{W_i W_j A_k} = \{0, 1\}$ refers to whether the AGV A_k executes the item W_j after executing the item W_i (the value of which is 1 means the AGV A_k performs the item W_j, when the item W_i is executed, as well as 0 is the opposite). $q_{W_j A_k} = \{0, 1\}$ refers to whether the item W_j will be executed by the AGV A_k (the value of which is 1 means item W_j is executed by the AGV A_k, and the value of 0 is the opposite).

Additionally, $t_{W_iW_jA_k}$ is the time for the AGV A_k to deliver items from the location of the item W_i to the location of the item W_j, which is calculated as the Formula (1). G_{A_k} presents the time for the AGV A_k to deliver all items, which is calculated as Formula (2).

$$t_{W_iW_jA_k} = \frac{d_{W_iW_j}}{v} p_{W_iW_jA_k} \tag{1}$$

$$G_{A_k} = ts_{A_k} - te_{A_k} = \sum_{i=1}^{M} \sum_{j=1}^{M} t_{W_iW_jA_k} \tag{2}$$

4.2.1. Objective function

During the task scheduling optimization process, we aim to enhance the system's performance primarily in terms of minimizing the delivery time of tasks. Moreover, we aim to reduce the number of AGVs required, as well as to ensure the economic benefits of this system. Therefore, based on the defined notations, we model three optimization objectives and associated constraints as follows.

The three objectives that we designed are presented as F_1, F_2, F_3, in which, as Formula (3) shows, the aim of F_1 is to use as few AGVs as possible to deliver goods so that the costs brought by the AGVs are minimized, and the goals of F_2 and F_3 are to optimize system performance and efficiency.

$$F_1 = min \sum_{i=1}^{K} N(A_i) \tag{3}$$

As Formula (4) indicates, the objective function F_2 is to minimize the maximum routing time. In this case, the routing time is the pure traveling time for the AGV, without taking into account the time spent handling conflicts. This function F_2 traverses each routing time G_{A_k} for every AGV, and minimizes the maximum time.

$$F_2 = min(max(G_{A_k})), \quad k = 1, \ldots, K \tag{4}$$

It is the objective function of F_3 in Formula (5) to reduce the potential conflicts between all AGVs traveling along a particular route. The conflict can be defined as follows. If multiple AGVs follow their planned routes, there will be a conflict if they arrive at the same place (the same coordinate point on the map) at the same time.

$$F_3 = min \sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} C_{A_iA_j} \tag{5}$$

Particularly, it is usually possible to resolve conflicts by letting the AGV with the lower priority wait or detour. Consequently, although different methods may result in different amounts of time spent on conflict resolution, it must spend more than a unit of time (the amount of time taken by the AGV to move to the next point on the map) in order to solve a conflict. As a result, the final delivery time is the sum of the routing time and the time spent resolving conflicts. Both of these objectives are considered in our work in order to improve the final delivery time of the system.

4.2.2. Constraint

There are specific constraints that must be mentioned during the scheduling process of AGVs in the AS/RS. First of all, the AGV can only pick up one item at a time, which is as Formula (6).

$$\sum_{k=1}^{K} \sum_{j=1}^{M} p_{W_iW_jA_k} = 1, \quad i = 1, \ldots, M \tag{6}$$

Meanwhile, the number of each AGV delivering items does not exceed its maximum load, which is as Formula (7).

$$\sum_{j=1}^{K} q_{W_jA_k} \leq L, \quad k = 1, 2, \ldots, K \tag{7}$$

Additionally, each task can only be assigned once, as indicated by Formula (8).

$$\sum_{k=1}^{K} q_{W_jA_k} = 1, \quad j = 1, \ldots, M \tag{8}$$

Lastly, Formula (9) demonstrates that the value of the total number of tasks assigned must be greater than or equal to the number of tasks required to be completed, in order to ensure that all tasks are considered.

$$\sum_{j=1}^{M} \sum_{k=1}^{K} q_{W_jA_k} \geq M \tag{9}$$

5. A multi-load AGVs task scheduling optimization method

According to the model above, a multi-load AGVs task scheduling optimization (MLATSO) method based on NSGA-II improvement is proposed for the multi-load AGVs based AS/RS, where the optimization objectives consist of minimizing the AGV quantity, routing time, and the estimated conflicts.

Particularly, NSGA-II (the Non-dominated Sorting Genetic Algorithm II) [35] has been proposed by Deb et al. NSGA-II is one of the most efficient multi-objective algorithms that employs non-dominated sorting and crowding distance to rank and select the fronts of the population. In addition, we improve the chromosome design, the crossover strategy, and the mutation strategy based on the original NSGA-II algorithm, in order to meet our particular requirements.

Moreover, the number of non-dominated solutions (Pareto fronts) is quite large. Hence, in order to further rank non-dominated solutions, we apply the TOPSIS (Technique for Order Performance by Similarity to Ideal Solution) [36] approach in order to avoid any imprecision caused by the subjective nature of the decisions, which is a practical and useful method among many famous multiple criteria decision-making methods for ranking and selecting a number of possible alternatives.

5.1. General procedure of MLATSO

Algorithm 1 MLATSO Method

Input: AGV quantity, Maximum Load of per AGV, Items to be transported, Generation size($Size_G$), Population size($Size_P$)
Output: Best Individual

```
Initial population P_t, t = 0
while t < Size_G and low similarity between Individuals in P_t do
    Get the offspring generation Q_t by crossover and mutation
    R_t = P_t ∪ Q_t
    G = Fast-non-dominated-sort(R_t)
    P_{t+1} = ∅ and i = 1
    while i ≤ N do
        Crow_{I_j} =crowding-distance-assignment(I_j) in G_i
        Sort Individuals I_j by Crow_{I_j}
        P_{t+1} = P_{t+1} ∪ G_i[1 : Size_P − |P_{t+1}|]
        i = i + 1
    end while
    t = t + 1
end while
Get the set P_t, sort the P_t by TOPSIS method, and Best Individual is the top solution.

return Best Individual
```

As the pseudo-code presented in Algorithm 1, the general procedure of our proposed MLATSO method based upon the NSGA-II Algorithm and TOPSIS Method is as follows.

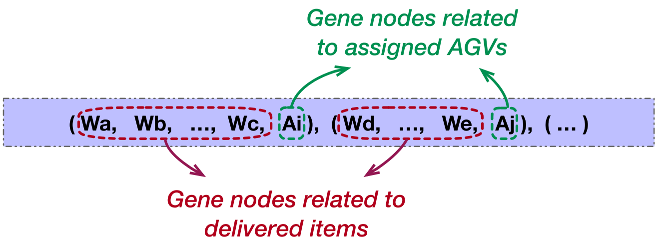


Fig. 2. Chromosome design.

- Step 1: Initially, a random parent population (P_t) of size $Size_P$ is created according to tasks required to execute, the current available AGVs and the load capacity of each AGV. ($Size_P$ is the number of chromosomes (Individuals) in the population (P_t)).
- Step 2: Generate offspring population (Q_t) from P_t by performing our designed crossover strategy and mutation strategy.
- Step 3: Combine parent population (P_t) and offspring population (Q_t) to give a $2{*}Size_P$ size combined population R_t; therefore $R_t = P_t \bigcup Q_t$.
- Step 4: Calculate fitness functions for every Individual in R_t.
- Step5: Perform non-dominated sorting to assign different fronts based upon the fitness functions. The result is expressed as collection G. $G = G_i(i = 1, 2, \ldots, N)$ refers to the different non-dominated fronts in G, and N is the number of the non-dominated fronts.
- Step 6: The new population of size $Size_P$ is filled by sequentially adding the individual in different non-dominated front sets(from first to last) until the size of the population exceeds $Size_P$. The last front solutions are selected based on crowding-distance to fill exactly $Size_P$ slots.
- Step 7: Copy the new population to the parent population and go to Step 2 until the maximum number of allowable generations ($Size_G$) is reached or high similarity between Individuals in the population.
- Step 8: The solutions of the problem are the first non-dominated front of the newly selected population, which contains the non-dominated solutions.
- Step 9: With further ranking the non-dominated solutions by applying the TOPSIS method, the top solution is decided as the best Individual.

5.2. Design of MLATSO

In the following section, we present our designed chromosomes, the crossover strategy and the mutation strategy, fitness functions, as well as how to select non-dominated solutions and how to sort non-dominated solutions using TOPSIS.

5.2.1. Chromosome design

Based on the special constraints of our model, we designed a new encoding chromosome based on real number coding, which is presented as Fig. 2. In this designed chromosome, $W_a, W_b, etc.$ represent the items that need to be delivered. $A_i, A_j, etc.$ represent the different assigned AGVs. Furthermore, the whole element $(W_a, W_b, \ldots, W_c, A_i)$ presents that the AGV A_i assigned to carry items $W_a, W_b, \ldots, W_c$, and the picking-up order is $W_a \rightarrow W_b \rightarrow \ldots \rightarrow W_c$.

5.2.2. Crossover and mutation

Considering the special constraints of the system, the traditional crossover method cannot be applied directly, due to the rate of convergence. Therefore, a three fragments crossover strategy is designed, in which, three fragments make up the daughter chromosome.

As Fig. 3 presenting, P1 and P2 are two parent chromosomes, and D is the daughter chromosome with the crossover strategy. Specifically, the three fragments of the daughter chromosome are Frag1, Frag2 and Frag3. Firstly, Frag2 is a gene selected randomly from one parent chromosome, Frag1 is made up of genes before the selected gene in the related parent, and the way to select items (W_i) and AGVs (A_i) to compose Frag3 is as follows. In detail, all genes related to items are selected from all W_i in the other parent and elements appearing in Frag1 are removed. All genes related to AGVs are selected from all available AGVs and elements appearing in Frag1 are removed. Meanwhile, the number of assigned AGVs in the daughter chromosome is a random number between the assigned AGVs quantity in two parent chromosomes. Based on that, the daughter chromosome is checked with all defined constraints as Formula (6), (7) and (8).

Considering the mutation, for the reason that each task can only be executed once, we design a mutation strategy in the form of a pair of gene nodes exchange, and this pair of genes can be randomly selected from genes related to delivered items or genes related to assigned AGVs on the parent chromosome. The daughter chromosome is then checked with all defined constraints.

Precisely, as Fig. 4 presenting, P1 is one parent chromosome, and D1, D2 and D3 are three examples of daughter chromosomes, which are obtained by mutation with three possible kinds of exchange, including two gene nodes related to delivered items exchange, two gene nodes related to assigned AGVs exchange, and one gene node related to delivered items and one gene node related to assigned AGVs exchange. Through our designed mutation strategy, we have reduced the number of poor daughter chromosomes caused by random mutations and improved the algorithm's performance.

5.2.3. Fitness function

Three aspects are considered to design the fitness functions of the MLATSO, which include the number of required AGVs, the pathing time, and the possible conflicts that may arise in the paths.

Precisely, the number of required AGVs can be calculated by the sum of A_i in our designed chromosome. Additionally, in order to obtain values of the pathing time (pure traveling time) and the possible conflicts, we build a public solution space to record the estimated conflicts appearing in possible paths, which is a way to ensure the efficiency of the algorithm. The solution space is represented by collection R, the $R_{S_iS_j}$ is the solution of the route from the shelf S_i to the shelf S_j, and the (x_{S_i}, y_{S_i}) is the coordinates of the shelf S_i, all related coordinates for conducting paths are obtained by a heuristic routing method in advance. With the solution space, if the AGV A_i and the A_j go through the same point at the same time, the objective function of AGVs conflicts (Formula (5)) can be calculated as Formula (12). Meanwhile, AGVs routing time (Formula (4)) can be calculated at the same time by the solution space, as Formula (10), (11).

$$R = \{R_{S_iS_j} | i, j = 1, 2, \ldots, M; i \neq j\} \tag{10}$$

$$R_{S_iS_j} = [(x_{S_i}, y_{S_i}), \ldots, (x_{S_j}, y_{S_j})] \tag{11}$$

$$C_{A_iA_j} = \sum_{t_u=min(ts_{A_i}, ts_{A_j})}^{max(te_{A_i}, te_{A_j})} C_{A_iA_jt_u} \tag{12}$$

5.2.4. Selecting non-dominated solutions

According to the basic NSGA-II Algorithm, with getting the combined population R_t, fast non-dominated sorting can be used to divide the combined population into different non-dominated fronts, and then, under consideration of the three different objectives, the individuals in the same non-dominated front can be sorted with the sum of the crowding-distance.

Firstly, for each fitness function F_k, $F_k(I_j)$ refers to the value of the fitness function F_k of Individual I_j, as well as $F_k(I_{j+1})$ and $F_k(I_{j-1})$ refer

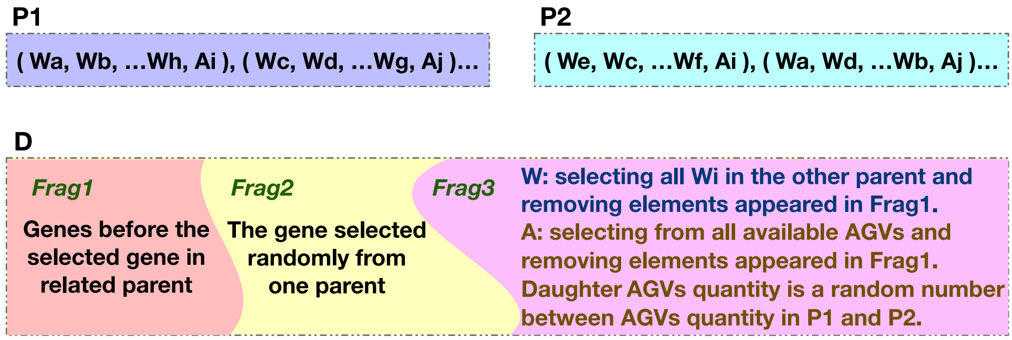


Fig. 3. Fragments crossover strategy.

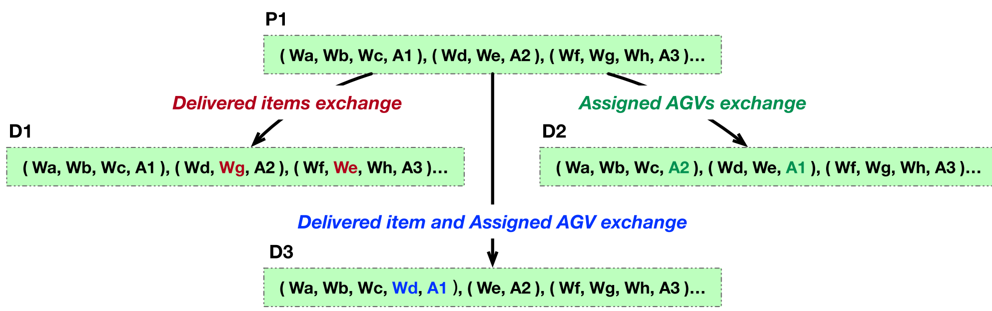


Fig. 4. Gene nodes exchange mutation strategy.

to the values of the fitness function F_k of two individuals closest to Individual I_j. $max(F_k(I_m))$ is the maximum value of the fitness function F_k of all individuals. Oppositely, $min(F_k(I_m))$ represents the minimum value.

In order to convert the values of three fitness functions (related to three objectives) measured in various units into common measurable units, for every fitness function, the normalization process is executed, in which, we calculate the ratio of the difference between $F_k(I_{j+1})$ and $F_k(I_{j-1})$ and the difference between $max(F_k(I_m))$ and $min(F_k(I_m))$.

Using the results, the three objectives are considered to be equally important, and the sum of the crowding distance for three fitness functions ($Crow_{I_j}$) can be obtained as Formula (13) demonstrating.

$$Crow_{I_j} = \sum_{k=1}^{3} \frac{|F_k(I_{j+1}) - F_k(I_{j-1})|}{max(F_k(I_m)) - min(F_k(I_m))}, \quad m = 1, 2, \ldots, Size_p \tag{13}$$

We believe that, for an Individual I_j, the larger the value of $Crow_{I_j}$, the better it is. In addition, $Crow_{I_j}$ will be set to infinity, when an individual is on the edge without two closest individuals, since its fitness function value is the maximum or the minimum.

5.2.5. TOPSIS for ranking non-dominated solutions

TOPSIS is a widely used and effective method for multi-objective decision analysis. In our algorithm, TOPSIS is used to rank the obtained non-dominated solutions based on consideration of three different objectives.

First of all, because the three defined objectives are measured in different units, the normalization process must be completed, in order to convert the values of three fitness functions (related to three objectives) into common measurable units. $z_{I_{jk}}$ represents the normalized result of Individual I_j on the fitness function F_k, which can be calculated as Formula (14) presenting. In particular, u is the number of individuals in non-dominated solutions.

$$z_{I_{jk}} = \frac{F_k(I_j)}{\sqrt{\sum_{j=1}^{u} F_k(I_j)^2}}, \quad k = 1, 2, 3 \tag{14}$$

Furthermore, as the Formula (15), a matrix Z is constructed for presenting the normalized results of all individuals in non-dominated solutions on three different fitness functions.

$$Z = \begin{bmatrix} z_{I_{11}} & z_{I_{12}} & z_{I_{13}} \\ z_{I_{21}} & z_{I_{22}} & z_{I_{23}} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ z_{I_{u1}} & z_{I_{u2}} & z_{I_{u3}} \end{bmatrix} \tag{15}$$

With the matrix Z, $z_k{}^+$ and $z_k{}^-$ can be obtained. Specially, $z_k{}^+$ is the one of $z_{I_{jk}}$, which is the largest normalized result on the fitness function F_k. It can be presented by Formula (16). On the contrary, $z_k{}^-$ is the smallest normalized result on the fitness function F_k, which is presented by Formula (17).

$$z_k{}^+ = max(z_{I_{1k}}, z_{I_{2k}}, \ldots, z_{I_{uk}}) \tag{16}$$

$$z_k{}^- = min(z_{I_{1k}}, z_{I_{2k}}, \ldots, z_{I_{uk}}) \tag{17}$$

In addition, we consider the importance of three fitness functions to be equal, and so the score (S_{I_j}) of Individual I_j can be calculated by Formula (18), which is used to rank all individuals within a non-dominated solution.

$$S_{I_j} = \frac{\sqrt{\sum_{k=1}^{3} (z_k{}^- - z_{I_{jk}})^2}}{\sqrt{\sum_{k=1}^{3} (z_k{}^- - z_{I_{jk}})^2} + \sqrt{\sum_{k=1}^{3} (z_k{}^+ - z_{I_{jk}})^2}} \tag{18}$$

6. Experimentation

In this section, a series of experiments are designed and analyzed. These experiments are intended to confirm our work with changeable task sizes, variable AGV quantities, different system layouts, and AGVs with different load capacities in the AS/RS.

Table 1
Experimental results of 5-Load AGVs system and 10-Load AGVs system with abundant AGVs under 20 rows and 20 columns of shelves layout.

TS	AAQ	MLATSO(5-L)			MMPSO(5-L)			MLATSO(10-L)			MMPSO(10-L)		
		RT	Con	OAQ	RT	Con	OAQ	RT	Con	OAQ	RT	Con	OAQ
10	5	38	0	2	50	11	5	51	0	1	55	4	5
20	10	49	8	5	64	15	9	61	0	3	66	14	9
30	15	54	19	8	67	33	13	66	5	5	68	30	13
40	20	64	35	10	71	44	17	73	13	6	75	38	17
50	25	68	48	13	72	66	22	81	25	8	79	60	21
60	30	73	77	14	82	79	26	85	29	10	81	77	26
70	35	75	101	19	88	105	32	91	38	11	84	95	31

TS: Task Size AAQ: Available AGVs Quantity Con: Conflict
RT: Routing Time (seconds) for finishing all tasks OAQ: Occupied AGVs Quantity

6.1. Experimental settings and parameters settings

The data from our experiment has been taken from an actual AGVs-based AS/RS in the city of Kunshan in China. Our algorithm generates 200 outbound tasks by randomly selecting outbound tasks per week.

We also model and simulate the actual environment with 20 rows and 20 columns of shelves, the detail layout of which is specified in Section 3. Meanwhile, in order to investigate the relationship between the effectiveness of reducing the number of AGVs in improving system performance and the layout of the system, we further construct an environment with 10 rows and 10 columns of shelves. Similar to the system with 20 rows and 20 columns of shelves, considering 10 rows of shelves, the width between each row can only pass through one AGV. Each row contains 10 shelves, and 5 shelves are in a group, which are tightly packed without any gaps between them. There is a one-way channel between different groups of shelves for one AGV to pass.

Furthermore, the AGV's speed is set to 1.0 m/s. Moreover, there are two AGVs considered in the experiments, the 5-load AGV, the maximum load capacity of which is 5, and the 10-load AGV, the maximum load capacity of which is 10. Lastly, all the results of the experiments are calculated with 100 sample populations, a crossover rate of 0.9 and a mutation rate of 0.2. The algorithm will stop when either the maximum generation (1000) has been reached or individuals within a population are very similar.

Moreover, we chose MMPSO (Modified Memetic Particle Swarm Optimization Algorithm) method [29] as a comparison method in our designed experiments, which is a method for scheduling multi-load AGVs used in a FMS (Flexible Manufacturing System), so that MMPSO and our proposed method are aimed at solving the problem of scheduling multi-load AGVs to optimize performance indicators in two relatively similar application scenarios. Specifically, for the FMS scenario, in the MMPSO method, the travel time (the time for all assigned AGVs to complete all tasks), and the waiting time (the total delayed service time for tasks) are considered as optimized targets. Conflicts of AGVs are not considered in the MMPSO method since it is assumed that the system controller will adjust the routing to handle congestions. In addition, the number of AGVs is not taken into account either. Accordingly, as a multi-load AGVs scheduling method for the manufacturing scenario in the literature, we derived the major idea of the MMPSO method, implemented it, and applied it to our AS/RS system scenario.

6.2. Experiments with abundant AGVs

6.2.1. Experiments of task scheduling methods

The first group experiments consider changeable task sizes (from 10 items to 70 items), two layouts (20 rows and 20 columns of shelves, 10 rows and 10 columns of shelves), two kinds of AGVs (5-load and 10-load), as well as the increasing number of AGVs (from 5 to 35) to test our proposed method for diverse scenarios. In addition, we measured MMPSO method [29] under the same conditions for comparison. The related results of 5-load AGVs and 10-load AGVs under two layouts systems are listed in Table 1 and Table 2, respectively.

In Table 1 and Table 2, we present the data in one row to provide an explanation of the results, and the shorthand for the result data is also applicable to other related tables. Specifically, the first two columns display the input parameters provided by the system, including the current task size (TS) and the number of available AGVs (AAQ) in use. Moreover, the other part presents system indicators obtained by our proposed MLATSO method and the MMPSO method respectively, including routing time for finishing all tasks (RT), the occupied quantity of AGVs (OAQ), and the possible conflicts in the routing paths of all AGVs (Con). Especially, the 5-L and 10-L labels refer to results of 5-load and 10-load AGVs, respectively.

Based on the results, as a first step, we compare and analyze our proposed MLATSO method with the existing MMPSO method. Generally, the results of the two different layouts exhibited similar characteristics. To perform the detailed analysis, we used the results of the layout system with 20 rows and 20 columns of shelves (the results in Table 1) to draw graphs, in order to illustrate the change in assigned AGVs quantity (Y direction in Fig. 5(a)), the routing time (Y direction in Fig. 5(b)), and estimated conflicts (Y direction in Fig. 5(c)) as task sizes increase (X direction in three sub- Figs. 5(a)–5(c)). In these Figures, the yellow full curve and the blue full curve demonstrate respectively the related indicators of the 5-load AGVs and the 10-load AGVs optimized by our proposed method. The yellow dotted curve and blue dotted curve respectively illustrate the related indicators of the 5-load and the 10-load AGVs optimized using the compared MMPSO method.

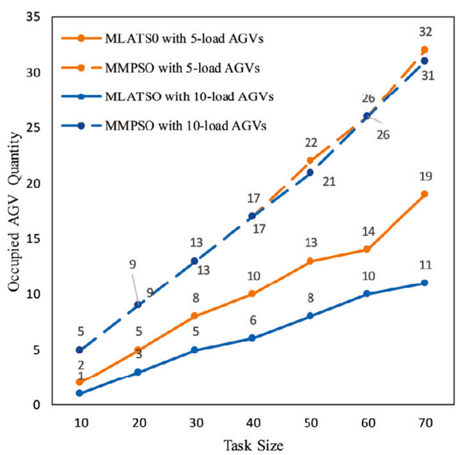
Specifically, the solid yellow line and solid blue line in Fig. 5(a) are lower than the dashed yellow line and dashed blue line respectively, which indicates that, with the same task sizes, the number of occupied AGVs in our proposed MLATSO method is singularly smaller than the number in MMPSO method. As shown in Fig. 5(b) and Fig. 5(c), yellow and blue full curves are below the yellow and blue dotted lines in terms of the overall trend, which indicates that MLATSO can provide better optimization than MMPSO as task sizes increase. According to the statement of the model, the routing time is defined as the time spent by the AGV in its travel, excluding the time spent on handling conflicts, and the final delivery time is defined as the sum of the routing time and the time spent on handling conflicts. With respect to the final delivery time, our proposed method has a better chance of achieving optimal results than MMPSO method.

Additionally, observing results from the same number of occupied AGVs, it is commonly observed that with our proposed method optimization, using the same amount of AGVs, more tasks are able to be accomplished with similar routing times and conflicts. For example, by using the same number of 5-Load AGVs (5 AGVs) and spending the similar delivery time, 20 tasks are completed according to our proposed MLATSO method, but a mere 10 tasks are completed according to the MMPSO method. Consequently, we propose our MLATSO method to provide a task scheduling scheme that engages fewer AGVs and reduces task delivery time by optimizing AGV quantity, routing time, and potential path conflicts.

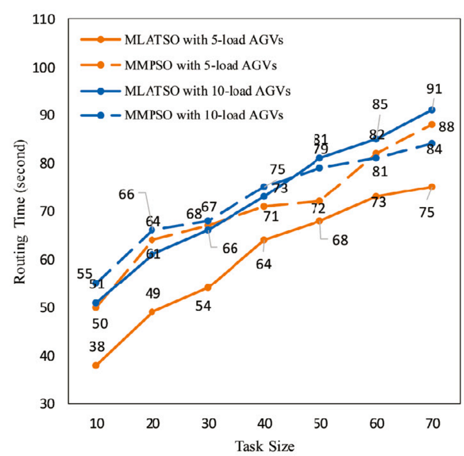
Table 2
Experimental results of 5-Load AGVs system and 10-Load AGVs system with abundant AGVs under 10 rows and 10 columns of shelves layout.

TS	AAQ	MLATSO(5-L)			MMPSO(5-L)			MLATSO(10-L)			MMPSO(10-L)		
		RT	Con	OAQ	RT	Con	OAQ	RT	Con	OAQ	RT	Con	OAQ
10	5	24	0	2	41	4	5	30	0	1	37	1	5
20	10	33	2	5	45	16	9	45	0	2	45	14	9
30	15	38	11	7	50	39	13	46	2	4	50	22	13
40	20	39	21	10	50	69	17	57	7	5	51	45	18
50	25	45	31	11	49	94	22	63	6	6	52	56	23
60	30	43	59	16	53	110	26	67	23	8	55	69	27
70	35	49	74	18	54	131	31	65	50	13	59	104	31

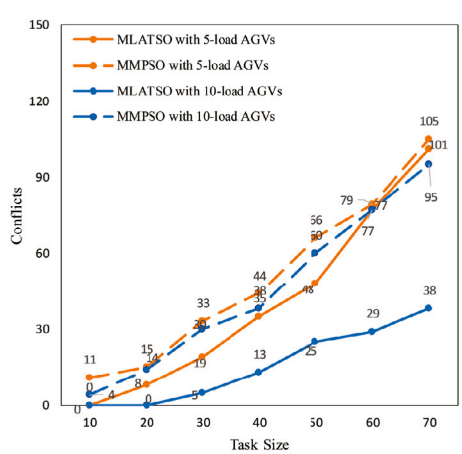
TS: Task Size AAQ: Available AGVs Quantity Con: Conflict
RT: Routing Time (seconds) for finishing all tasks OAQ: Occupied AGVs Quantity



(a) AGVs quantity



(b) Routing time



(c) Conflicts

Fig. 5. Comparison between MLATSO method and MMPSO method under 20 rows and 20 columns of shelves layout.

Additionally, focusing on the routing time change presented by Fig. 5(b), it is found that when the task size is large, the routing time optimized by MMPSO method is somewhat lower than that optimized by our MLATSO method, due to the approximately double number of AGVs occupied. To investigate optimization in different scenarios, we design another group experiment with large task sizes and limited AGVs.

6.2.2. Experiments of load capacity analysis

AGVs with different capacities are also considered in relation to system indicators. Specifically, in the yellow and blue full lines of the three sub- Figs. 5(a)–5(c), it appears that, for the same given task size, the system needs more 5-load AGVs than 10-load AGVs, while the routing time is opposite. The reason is that, AGVs with larger load capacity take longer time to deliver, while conflicts decrease with the fewer number of AGVs on the delivery route. However, by using AGVs with smaller load capacity, the number of AGVs assigned must increase in order to reduce routing time, and therefore more conflicts arise. Thus, for the AS/RS stakeholders, the decision to apply different load AGV types is a comprehensive decision that must take into consideration the equipment budget, the AS/RS space, and the system effectiveness requirements.

Table 3
Conflict differences between the results of MMPSO the results of MLATSO with two load AGVs and in two layouts.

TS	AAQ	Layout20 (5-Load AGVs)	Layout10 (5-Load AGVs)	Layout20 (10-Load AGVs)	Layout10 (10-Load AGVs)
10	5	11	4	4	1
20	10	7	14	10	14
30	15	12	28	25	20
40	20	11	48	26	38
50	25	18	63	35	50
60	30	2	51	48	46
70	35	4	57	47	54

TS: Task Size AAQ: Available AGVs Quantity
Layout20: the conflict differences in the layout with 20 rows and 20 columns of shelves
Layout10: the conflict differences in the layout with 10 rows and 10 columns of shelves

6.2.3. Experiments of system layout analysis

Concern has been raised about the relationship between the effectiveness of reducing the number of AGVs in improving system performance and the layout of the system. Because the task sizes and the AGV settings in the layout with 10 rows and 10 columns of shelves are consistent with the layout with 20 rows and 20 columns of shelves, as well as the channel mode for the AGV passing. That is, in terms of the ratio of AGV quantity per unit space, the ratio of AGV quantity per unit space in the 10 rows and 10 columns of shelves system is greater.

In addition, with the results of Table 1 and Table 2, we are able to determine the differences of the number of conflicts between the optimization without considering AGV quantity (as the results of MMPSO method) and the optimization with considering AGV quantity (as the results of MLATSO method) under these two different layout systems. Precisely, the conflict difference can be calculated as follows. In each layout, for the particular task size and the number of available AGVs, the conflict difference is calculated as the number of conflicts in the scheduling result of MMPSO method minus the number of conflicts in the scheduling result of MLATSO Method. The calculated results are shown in Table 3.

In light of the calculated differences listed in Table 3, on the overall trend, it is generally evident that the differences of conflicts in the layout with 10 rows and 10 columns of shelves (as data in columns Layout10) are greater than the differences in the layout with 20 rows and 20 columns of shelves (as data in columns Layout20). This implies that the effect on conflict optimization is more pronounced in the system with the layout of 10 rows and 10 columns of shelves. Therefore, in a system with a greater number of AGVs per unit space (for instance, a large number of AGVs distributed in a small scale system), the optimization effect of conflicts is more evident, so AGV number optimization is more crucial.

6.3. Experiments of limited AGVs

In the experiments above, we assume that the number of available AGVs is always sufficient. It means that there are as many AGVs as the system requires. However, the number of available AGVs is limited for most actual systems. In response, we conduct an additional experiment using unabundant AGVs under 20 rows and 20 columns of shelves layout system. The purpose of this experiment is to determine whether the continued growth of dedicated AGVs is associated with the continued improvement of overall system performance.

As a first step, we perform numerous experiments within the same constant task size (30) while increasing the number of AGVs available (from 2 to 10). In line with the data found in Table 4, there is a limit to the number of AGVs (8 AGVs) that can be added to the system, beyond which it will no longer be possible to improve efficiency by adding more AGVs.

Additionally, we perform other scenarios, where the task size is increased from 50 to 70, but the number of corresponding AGVs is insufficient, as 10, 15 and 20. Precisely, the related experimental results

Table 4
Experimental results of 5-Load AGVs system with a constant task size and increasing number of AGVs under 10 rows and 10 columns of shelves layout.

TS	AAQ	MLATSO			MMPSO		
		RT	Con	OAQ	RT	Con	OAQ
	2	198	0	2	218	0	2
	3	118	1	3	135	1	3
	4	81	5	4	102	10	4
30	5	81	5	4	86	13	5
	6	68	15	6	86	13	5
	7	59	19	7	69	24	7
	8	54	19	8	56	23	8
	9	54	19	8	54	27	9
	10	54	19	8	53	55	10

TS:Task Size AAQ:Available AGVs Quantity OAQ:Occupied AGVs Quantity
RT:Routing Time (seconds) for finishing all tasks Con:Conflict

are listed in Table 5. As we analyze these results, it is apparent that due to the fact that the optimization of AGV quantity is not taken into account in MMPSO method, all available AGVs are assigned in this environment. Nonetheless, our proposed method optimizes the number of AGVs, the routing time of AGVs, and possible conflicts simultaneously, so that fewer AGVs are assigned to related tasks, and both the routing time and conflicts are better than the MMPSO method.

Thus, based on a series of experiments executed in scenarios of incremental task sizes, changeable load capacity AGV types, and the number of available AGVs, we conclude that our work is capable of optimizing task delivery time by using a minimal number of required AGVs, thus assisting the stakeholders of AS/RS in determining the most suitable quantity of multi-load AGVs, and ultimately improves both the system performance and the AGV investment, thus maximizing economic benefits.

On the other hand, we examine the computation time taken by our proposed MLATSO method and that of the compared MMPSO method. In all of the scenarios mentioned above, we recorded the computation times for both methods. As the number of tasks increases from 10 to 70, the MLATSO method takes 10 s to 3 min to produce scheduling results, while the MMPSO method takes 10 s to 2 min. Generally, the results demonstrate that our proposed method requires more computation time to achieve scheduling results than the MMPSO method. The main reason is that our proposed method emphasizes more optimization objectives, which may result in longer computation times. Hence, one of our future endeavors will be to improve the mutation mechanism, in order to reduce the computation time required by our method.

7. Conclusion

In this paper, we propose our work dedicated to analyze and optimize the task scheduling in the multi-load AGVs-based AS/RS. Precisely, we provided a model to support the specialization of the retrieval process using multi-load AGVs in the AS/RS, and analyzing key performance indicators. Based on that, we designed a multi-load

Table 5
Experimental results of 5-Load AGVs system and 10-Load AGVs system with different task sizes and limited AGVs under 20 rows and 20 columns of shelves layout.

TS	AAQ	MLATSO(5-L)			MMPSO(5-L)			MLATSO(10-L)			MMPSO(10-L)		
		RT	Con	OAQ	RT	Con	OAQ	RT	Con	OAQ	RT	Con	OAQ
50	20	68	43	13	89	55	20	81	26	8	78	46	20
	15	68	43	13	92	49	15	81	26	8	90	47	15
	10	131	53	8	129	66	10	81	26	8	135	55	10
60	20	68	78	14	95	77	20	81	35	9	89	68	20
	15	68	78	14	122	109	15	81	35	9	99	61	15
	10	131	107	10	139	74	10	81	35	9	141	65	10
70	20	73	109	18	132	121	20	83	41	10	108	75	20
	15	137	153	15	127	158	15	83	41	10	129	93	15
	10	176	157	10	188	165	10	88	51	10	155	110	10

TS:Task Size AAQ:Available AGVs Quantity OAQ:Occupied AGVs Quantity
RT:Routing Time (seconds) for finishing all tasks Con:Conflict

AGVs task scheduling optimization (MLATSO) method, with which, for a given task, decisions related to which particular AGVs to assign, the loads selected for each assigned AGV, and the routing path could be intelligently determined, and the objectives could be achieved simultaneously in terms of the fewest number of occupied AGVs, the shortest travel time, and the fewest conflicts between AGVs. Finally, the experiments were conducted considering various scenarios, which include increasing task sizes, the number of AGVs, and two load capacity AGV types in an AS/RS system. Analyzing the experimental results, we have demonstrated that our work maximizes system capability by using fewer AGVs. As a result, AS/RS stakeholders can benefit from both the system performance and the AGV investment, thus maximizing economic benefits.

Future research will focus on three directions. The first is to improve the mutation mechanism to reduce the computational time of our task scheduling method. Furthermore, the task scheduling method with the ability to reschedule dynamically is necessary to be concerned, in order to handle the situation where AGVs experience unexpected faults. The other is considering task scheduling adapted to dynamic scheduling requirements (e.g. dynamically changing throughput and delivery time, as well as temporary dynamic tasks), so as to meet the collaborative scheduling optimization of the AGV-based system and other links (e.g. automatic sorting, packaging and palletizing) in complex cluster intelligent manufacturing scenarios.

corollary

CRediT authorship contribution statement

Yishuai Lin: Conceptualization, Methodology, Resources, Supervision, Writing – review & editing. **Yunlong Xu:** Data curation, Software, Writing – original draft. **Jiawei Zhu:** Validation, Investigation. **Xuhua Wang:** Investigation, Visualization. **Liang Wang:** Resources, Validation. **Gang Hu:** Software, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge technical assistant from Mr. Qingshan LI and Mr. Tonghao AN, and support by the Basic and Applied Basic Research Program of Guangdong Province (2021A1515110518), the National Natural Science Foundation of China (61902039) and the Key Research and Development Program of Shaanxi Province (2020ZDLGY09-04).

References

[1] E.H. Grosse, C.H. Glock, W.P. Neumann, Human factors in order picking: a content analysis of the literature, Int. J. Prod. Res. 55 (5) (2017) 1260–1276.
[2] T. Zhou, D. Tang, H. Zhu, Z. Zhang, Multi-agent reinforcement learning for online scheduling in smart factories, Robot. Comput.-Integr. Manuf. 72 (2) (2021) 102202.
[3] H. Yoshitake, R. Kamoshida, Y. Nagashima, New automated guided vehicle system using real-time Holonic scheduling for warehouse picking, IEEE Robot. Autom. Lett. 4 (2) (2019) 1045–1052.
[4] D. Culler, J. Long, A prototype smart materials warehouse application implemented using custom mobile robots and open source vision technology developed using EmguCV, Proc. Manuf. 5 (2016) 1092–1106, 44th North American Manufacturing Research Conference, NAMRC 44, June 27-July 1, 2016, Blacksburg, Virginia, United States.
[5] M. Cardona, A. Palma, J. Manzanares, COVID-19 pandemic impact on mobile robotics market, in: 2020 IEEE ANDESCON, 2020, pp. 1–4.
[6] T. Miyamoto, K. Inoue, Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems, Comput. Ind. Eng. 91 (2016) 1–9.
[7] K. Murakami, Time-space network model and MILP formulation of the conflict-free routing problem of a capacitated AGV system, Comput. Ind. Eng. 141 (2020) 106270.
[8] H. Fazlollahtabar, M. Saidi-Mehrabad, J. Balakrishnan, Mathematical optimization for earliness/tardiness minimization in a multiple automated guided vehicle manufacturing system via integrated heuristic algorithms, Robot. Auton. Syst. 72 (2015) 131–138.
[9] Z. Li, N. Wu, M. Zhou, Deadlock control of automated manufacturing systems based on Petri nets—A literature review, IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) 42 (4) (2012) 437–462.
[10] M. Mousavi, H.J. Yap, S.N. Musa, F. Tahriri, S. Dawal, Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization, PLOS ONE 12 (2017).
[11] Z. He, V. Aggarwal, S.Y. Nof, Differentiated service policy in smart warehouse automation, Int. J. Prod. Res. 56 (22) (2018) 6956–6970.
[12] O.T. Baruwa, M.A. Piera, A coloured Petri net-based hybrid heuristic search approach to simultaneous scheduling of machines and automated guided vehicles, Int. J. Prod. Res. 54 (16) (2016) 4773–4792.
[13] N. Singh, Q.-V. Dang, A. Akcay, I. Adan, T. Martagan, A matheuristic for AGV scheduling with battery constraints, European J. Oper. Res. 298 (3) (2022) 855–873.
[14] J. Yu, Y. Li, Research on AGV scheduling based on point-to-point mode, in: 2020 5th International Conference on Mechanical, Control and Computer Engineering, ICMCCE, 2020, pp. 1943–1947.
[15] B. Bao, Z. Duan, W. Chen, Mission scheduling of multi-agv system with dynamic simulation, in: 2020 International Symposium on Autonomous Systems (ISAS), 2020, pp. 115–120.
[16] J. Zhang, Z. Li, L. Li, Y. Li, H. Dong, A bi-level cooperative operation approach for AGV based automated valet parking, Transp. Res. C 128 (2021) 103140.
[17] M. De Ryck, D. Pissoort, T. Holvoet, E. Demeester, Decentral task allocation for industrial AGV-systems with routing constraints, J. Manuf. Syst. 62 (2022) 135–144.
[18] J. Xin, R.R. Negenborn, F. Corman, G. Lodewijks, Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance, Transp. Res. C 60 (2015) 377–396.
[19] Z. Yang, C. Li, Q. Zhao, Dynamic time estimation based AGV dispatching algorithm in automated container terminal, in: Proceeding of the 37th Chinese Control Conference, 2018, pp. 7868–7873.
[20] J. Hao, C. Wang, M. Yang, B. Wang, Hybrid genetic algorithm based dispatch and conflict-free routing method of AGV systems in unmanned underground parking lots, in: 2020 IEEE International Conference on Real-Time Computing and Robotics, RCAR, 2020, pp. 475–480.

[21] W.-Q. Zou, Q.-K. Pan, T. Meng, L. Gao, Y.-L. Wang, An effective discrete artificial bee colony algorithm for multi-AGVs dispatching problem in a matrix manufacturing workshop, Expert Syst. Appl. 161 (2020) 113675.
[22] K. Vivaldini, L. Rocha, N. Martarelli, M. Becker, A. Moreira, Integrated tasks assignment and routing for the estimation of the optimal number of AGVS, Int. J. Adv. Manuf. Technol. 82 (1–4) (2016) 719–736.
[23] M. Mousavi, H.J. Yap, S.N. Musa, A fuzzy hybrid GA-PSO algorithm for multi-objective AGV scheduling in FMS, Int. J. Simul. Model. (IJSIMM) 16 (1) (2017) 58–71.
[24] Y. Li, W. Gu, M. Yuan, Y. Tang, Real-time data-driven dynamic scheduling for flexible job shop with insufficient transportation resources using hybrid deep Q network, Robot. Comput.-Integr. Manuf. 74 (2022) 102283.
[25] E.C. Norton, B.E. Dowd, M.L. Maciejewski, Marginal effects—Quantifying the effect of changes in risk factors in logistic regression models, JAMA 321 (13) (2019) 1304–1305.
[26] Y.-C. Ho, S.-H. Chien, A simulation study on the performance of task-determination rules and delivery-dispatching rules for multiple-load AGVs, Int. J. Prod. Res. 44 (20) (2006) 4193–4222.
[27] Z. Yao, S. Fujimura, Simulation of multi-load AGV system based on JIT production environment, in: Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, Vol. 2, ICCSIT 2010, 2010.
[28] R. Yan, L. Jackson, S. Dunnett, A study for further exploring the advantages of using multi-load automated guided vehicles, J. Manuf. Syst. 57 (2020) 19–30.
[29] V.K. Chawla, A.K. Chanda, S. Angra, Scheduling of multi load AGVs in FMS by modified memetic particle swarm optimization algorithm, J. Project Manag. (2018) 39–54.
[30] X. Ma, Y. Bian, F. Gao, An improved shuffled frog leaping algorithm for multiload AGV dispatching in automated container terminals, Math. Probl. Eng. 2020 (2020) 1–13.
[31] Y. Kang, G. Feng, Z. Wang, J. Sun, Real-time Task Assignment Method of Two-load AGV under Dynamic Change of Goods Urgency in Logistics Warehouse, Vol. 1576, IOP Publishing, 2020, p. 012055.
[32] K. Braekers, K. Ramaekers, I. Van Nieuwenhuyse, The vehicle routing problem: State of the art classification and review, Comput. Ind. Eng. 99 (2016) 300–313.
[33] G.-S. Wang, Y.-X. Yu, An improved ant colony algorithm for VRP problem, in: 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, 2010, pp. 129–133.
[34] P.T. Daely, Y.J. Aruan, J.M. Lee, D.-S. Kim, Dynamic VRP optimization using discrete PSO in edge computing environment, in: 2021 International Conference on Information and Communication Technology Convergence, ICTC, 2021, pp. 654–656.
[35] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature PPSN VI, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 849–858.
[36] H.-S. Shih, H.-J. Shyur, E.S. Lee, An extension of TOPSIS for group decision making, Math. Comput. Modelling 45 (7) (2007) 801–813.