vehicles

MDPI

Article

Multi-AGV Scheduling and Path Planning Based on an Improved Ant Colony Algorithm

Yang Xu *, Wei Liu and Hao Yuan

Yancheng Institute of Technology, School of Automotive Engineering, Hope Avenue Block, Yancheng 224001, China; liuwei@ycit.edu.cn (W.L.); yuanhao0131@outlook.com (H.Y.)
* Correspondence: xuyang0266@outlook.com

Abstract

In current intelligent manufacturing workshops, multi-automated guided vehicle (AGV) systems often face issues such as uneven task allocation, path conflicts, and idle travel, which significantly affect scheduling efficiency. To address these problems, this paper proposes an improved ant colony algorithm that collaboratively optimizes task allocation and path planning by integrating path costs and AGV task execution capabilities. The algorithm utilizes shortest-path planning results to optimize task allocation priorities, achieving synchronized optimization of task scheduling and path planning. Based on this, a multi-objective scheduling model is constructed with the goal of minimizing task completion time, idle travel distance, and total travel distance. The results show that the method effectively shortens task completion time and significantly improves scheduling efficiency, verifying its feasibility for application in intelligent manufacturing workshops.

Keywords: collaborative scheduling of multi-AGVs; conflict-avoidance path planning; intelligent manufacturing workshop; improved ant colony algorithm

check for updates

Academic Editor: Yongmin Zhong

Received: 4 June 2025
Revised: 30 June 2025
Accepted: 3 July 2025
Published: 17 September 2025

Citation: Xu, Y.; Liu, W.; Yuan, H. Multi-AGV Scheduling and Path Planning Based on an Improved Ant Colony Algorithm. *Vehicles* **2025**, *7*, 102. https://doi.org/10.3390/vehicles7030102

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The multi-AGV (automated guided vehicle) system has found widespread applications in intelligent manufacturing, logistics distribution, and flexible production systems. With the increasing complexity of production environments, optimizing task scheduling and path planning for multi-AGV systems to enhance overall system efficiency has become a core issue in current research.

Wang et al. [1] developed a novel multi-state AGV scheduling algorithm that dynamically optimizes task scheduling by comprehensively considering vehicle status, travel distance, trajectory characteristics, and the impact of multi-AGV collisions on travel time. Lin et al. [2] proposed a system task scheduling optimization method based on multi-load AGVs, with optimization goals including minimizing the number of AGVs, travel time, and conflicts. This approach achieved optimal AGV utilization and minimized operational costs. In terms of energy efficiency, Zou et al. [3] established a mathematical model aimed at minimizing energy consumption for a matrix manufacturing workshop and designed a hybrid constructive heuristic algorithm to improve scheduling energy efficiency optimization. Li et al. [4] studied the impact of AGV charging on multi-AGV flexible manufacturing cell scheduling, analyzing the relationships between scheduling tasks, charging thresholds, and power consumption, and solved the scheduling model using a Genetic Algorithm (GA), improving the intelligence of energy management. Hu et al. [5] focused on dynamic scheduling issues, establishing a Mixed-Integer Linear Programming (MILP) model and

using a hierarchical planning approach to decompose the problem into upper-level task allocation and lower-level path planning subproblems, which were solved using a hybrid discrete state transition algorithm (HDSTA) based on elite solution sets and taboo lists.

Regarding cost optimization, Zou et al. [6] proposed an MILP model combined with an adaptive iterative greedy (SAIG) algorithm to optimize multi-AGV scheduling costs in matrix manufacturing workshops. To address the issue of AGV scheduling systems lacking adaptability to unforeseen events, Li et al. [7] developed a dynamic AGV scheduling model, employing a non-periodic departure strategy and a real-time task list updating method to significantly improve dynamic responsiveness. Zou et al. [8] focused on the multi-compartment AGV scheduling problem in matrix manufacturing workshops (MC-AGVS), establishing an MILP model aimed at minimizing total costs and optimizing it using an iterative greedy (IG) algorithm. Maoudj et al. [9] studied a novel multi-AGV scheduling problem with conflicting products (CMASPCP), considering both AGV capacity constraints and the impact of conflicting products on scheduling, and proposed an optimization model based on a Multi-Agent System (MAS). Liu et al. [10] addressed the material replenishment AGV scheduling problem (AGVDP) by proposing a mixed-integer optimization model with the objectives of minimizing transportation costs and delivery time deviations, and developed a knowledge-guided distribution estimation algorithm to solve the model, significantly improving scheduling efficiency.

Path planning is a critical component of multi-AGV systems. Wang et al. [11] proposed a time reuse strategy that addressed the mismatch between AGV planning server computation time and actual movement time. By managing the path length of each planning instance, the synchronization of computation time and AGV movement time was achieved, improving the real-time nature of path planning and avoiding a decrease in path optimality. In dynamic environments, Bai et al. [12] introduced a two-level path planning method (GA-KL) that combined global and local path planning algorithms, effectively enhancing the path optimization capability of multi-AGV systems in complex environments. Shen et al. [13] studied the path coordination issue in multi-AGV systems, proposing a multi-agent A3C algorithm (MAA3C) with an attention mechanism to optimize path distribution in warehouse environments, improving AGV coordination. Regarding path optimization algorithms, Qiuyun et al. [14] proposed an Improved Particle Swarm Optimization (IPSO) algorithm and established a mathematical model with the goal of minimizing transport time, achieving a coordinated optimization of path planning and task scheduling. Yang et al. [15] introduced a time-slice-based rotating deadlock prevention algorithm, optimizing the traffic control mechanism of multi-AGV path planning based on system-level scheduling. Lin et al. [16] studied the robustness of multi-AGV path planning, proposing a two-layer strategy that combined local path optimization and global path planning, enhancing the stability of path planning.

Additionally, Shan et al. [17] addressed the dynamic path planning problem in multi-AGV systems by proposing a time-window-based collision-free path optimization algorithm and developing a time window conflict model, which improved the reliability of path planning. Liu et al. [18] proposed an improved A* algorithm that integrates the Artificial Potential Field (APF) method, introducing a dynamic potential field function to solve deadlock issues and improving path smoothness by 23.6%. Farooq et al. [19] studied a multi-AGV path optimization method based on an improved Genetic Algorithm (GA), establishing an optimization model with time-independent and time-dependent variables, which enhanced the flexibility of path planning. Jiang et al. [20] proposed a two-stage optimization method, where the first stage used the A* algorithm to calculate the shortest obstacle-free path in a warehouse, and the second stage used a Genetic Algorithm (GA) for task assignment optimization, minimizing task completion time and AGV energy consumption. Zhang

et al. [21] designed an algorithm combining adjacency and reward matrices for multi-AGV path planning in grid-based warehouse environments, reducing the number of corners in the path and improving path planning efficiency. Liu et al. [22] proposed an improved Model Predictive Control (MPC) algorithm combined with a Gradient Projection Method (GPM) to optimize the calculation of projection directions, improving trajectory tracking accuracy and reducing computational iterations. Bahwini T et al. [23] developed an optimal path planning method for needle insertion in soft tissue, coupling bioheat transfer with tissue deformation models. This approach leverages steady-state temperature gradient potential fields—solved via finite element method with variable thermal conductivity—to optimize insertion trajectories. Zhong Y et al. [24] proposed a neurodynamic path planning approach. This method constructs targets and obstacles as activity landscape peaks and valleys, respectively, enabling real-time generation of smooth collision-free optimal paths in dynamic environments. Zhou Q et al. [25] developed a crossover-recombined Global-best Brain Storm Optimization (GBSO) algorithm incorporating cubic B-spline representation. This method solves 3D UAV path planning under continuous curvature constraints by optimizing a comprehensive cost function integrating safety, economy, and flyability requirements, with simulations demonstrating superior performance over GBSO, SHADE, and benchmark algorithms. Hills J et al. [26] proposed a transient heat conduction-based robotic path planning method that treats the target as a constant heat source and obstacles with Dirichlet boundary conditions. Using cellular neural networks for real-time temperature field computation, this approach generates collision-free optimal paths in dynamic environments via iterative heat flux algorithms without global search or prior environmental knowledge.

The remainder of this paper is organized as follows: Section 2 describes the AGV scheduling problem and modeling approach. Section 3 presents the improved ant colony algorithm. Section 4 conducts comparative experiments of the algorithm. Finally, Section 5 summarizes the conclusions of this study.

2. Problem Description and Model Building

2.1. Problem Description

The production workshop includes several areas: the production work zone, the parts assembly area, aisles, the warehouse, and the AGV transport zone; the specific layout is shown in Figure 1. Raw materials, finished products, and AGVs are temporarily stored in the warehouse. The production lines are located in the loading area, where parts are produced, and the AGVs transport the finished parts from the end of the production line to the assembly area through the transport zone. At the start of the planning period, all AGVs queue at the starting point in the warehouse, awaiting tasks. The areas at the end of the production line and the front of the assembly line serve as parts storage zones. AGVs move products from the production line to the storage area via conveyors, then load and transport the parts. After completing the production task, the AGVs deliver the products to the designated assembly area and transfer them to the assembly production line via conveyors. The workshop contains A production lines, each associated with a product storage area and a loading point, represented as set $X = \{X_1, X_2, \ldots, X_A\}$. There are also B assembly lines, each corresponding to an unloading point, represented as set $Y = \{Y_1, Y_2, \ldots, Y_B\}$. The AGVs have a load capacity of Z_c, allowing them to handle transportation tasks from multiple loading points simultaneously, as long as the total load does not exceed the maximum capacity. This paper treats the transportation control problem as a constrained multi-objective optimization problem, with the goals of minimizing task completion time, total travel distance, and idle travel distance, to improve overall scheduling efficiency.

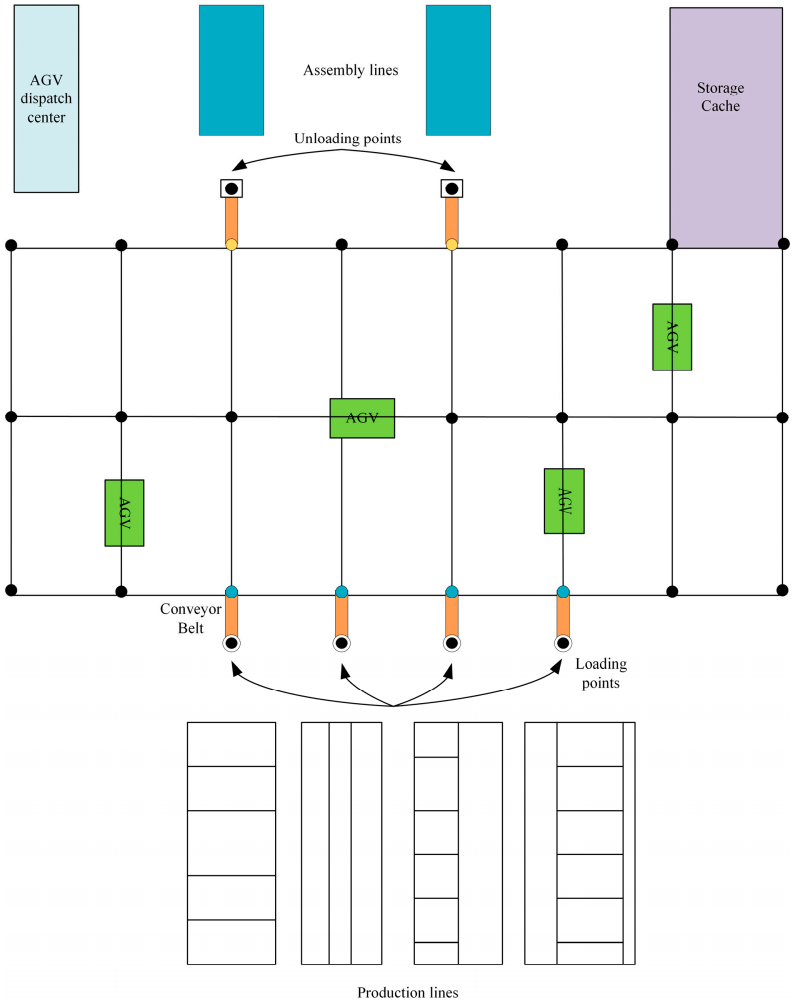


Figure 1. Map of the production line environment.

2.2. Model Building

A mathematical model is developed with optimization objectives focused on task completion time, idle travel distance, and total travel distance. The key settings and assumptions are as follows:

Each task can only be assigned to one AGV, and tasks are continuous. Once a transportation task is assigned to an AGV, that AGV is responsible for transporting the product to its destination until the task is completed. Each AGV can only perform the current task and will not accept another until the current task is finished. All AGVs travel at the same uniform speed, with no significant difference in speed between loaded and unloaded conditions, effectively avoiding conflicts caused by overtaking. When an AGV reaches a node on the path, it must wait for the AGV already occupying that node to depart before continuing its journey. The loading and unloading time for each AGV per task is fixed at t, with no significant differences in time during startup or stoppage. Products are conveyed along the path from the product placement area to the adjacent nodes via a conveyor belt, and AGV loading and unloading points are set within the network. After completing the task, the AGV will return to its starting station.

2.2.1. Minimum Task Completion Time

$$Min f_1(X) = max_{m \in AGV}(T_{i,end}) \quad (1)$$

2.2.2. Minimum Idle Travel Distance

$$minf_2(X) = \sum_{m \in AGV} \sum_{j=1}^{N_i - 1} D_{empty}(i,j) \tag{2}$$

2.2.3. Minimum Total Travel Distance

$$minf_3(X) = \sum_{i \in AGV} \sum_{j=1}^{N_i} D_{total}(i,j) \tag{3}$$

Constraints:

$$\sum_{K=1}^{m} \sum_{i=0}^{n} x_{ijk} = 1, \forall j \in C \tag{4}$$

$$\sum_{K=1}^{m} \sum_{j=0}^{n} x_{ijk} = 1, \forall i \in C \tag{5}$$

$$\sum_{J=1}^{n} x_{0jk} \leq 1, \forall k \in K \tag{6}$$

$$\sum_{I=0}^{n} x_{ijk} = \sum_{i=0}^{n} x_{jik}, \forall j \in C, \forall k \in K \tag{7}$$

$$\sum_{(I,j) \in S \times S} x_{ijk} \leq |S| - 1, \forall S \subseteq V,\ 1 \leq |S| \leq n, \forall k \in K \tag{8}$$

$$Q_{ik} + q_j + \Psi\left(1 - x_{ijk}\right) \leq Q_{jk}, \forall (I,j) \in B, \forall k \in K \tag{9}$$

$$T_j^c \leq y_{jk} T_{jk}^a \leq T_j^d, \forall j \in C, \forall k \in K \tag{10}$$

$$Y_{jk} Q_{jk} \leq Q, \forall j \in C, \forall k \in K \tag{11}$$

$$T_{ik}^a = T_k^r, i = 0, \forall k \in K \tag{12}$$

In the proposed model, Equations (1)–(3) define the three optimization objectives. Constraints (4) and (5) ensure that each customer is served by exactly one AGV. Constraint (6) limits each AGV to visiting at most one customer after leaving the warehouse. Constraint (7) requires each AGV to travel from the current task location to the next task location upon completing its current task. Constraint (8) stipulates that the number of AGVs deployed must not exceed the predefined total number mm. Constraint (9) establishes the relationship between the arrival times of two consecutive customers. Constraint (10) defines the time window requirements for task execution. Constraint (11) ensures that the load of each AGV does not exceed its maximum capacity while serving customers. Finally, Constraint (12) guarantees that the release times of all AGVs comply with the scheduling requirements.

3. Proposed Algorithm

3.1. Traditional Ant Colony Optimization

As shown in the Figure 2, the Ant Colony Optimization (ACO) algorithm achieves collaboration through ants releasing and sensing pheromones. In the initial phase, ants randomly select paths to explore. As the search progresses, the frequency of ants traveling along shorter paths gradually increases, enhancing the pheromone concentration on these paths. This, in turn, attracts more ants to choose these routes, creating a positive feedback mechanism. Eventually, the pheromone accumulates on the optimal path, completing the path optimization process.

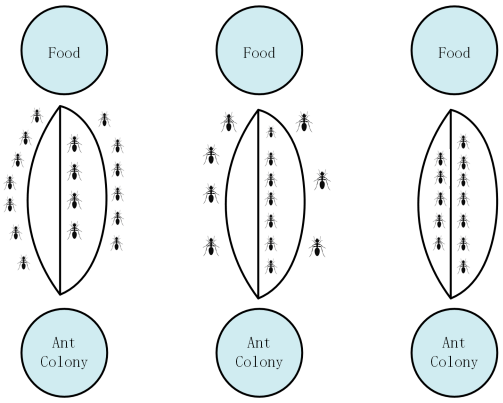


Figure 2. Ant colony algorithm illustration.

In this paper, the Ant Colony Optimization (ACO) algorithm is used to solve the multi-AGV task assignment problem. Ants select the next set of tasks from the current task pool, continuing this process until all tasks are assigned. Ultimately, the algorithm computes the path costs for completing all tasks and selects the task assignment with the shortest path cost from all the ants. The task assignment process based on ACO is illustrated in the Figure 3, which mainly involves initializing path cost information, initializing pheromone information, and iterating through the search. The specific steps are as follows:

1. **Task Assignment**: Tasks are assigned to specific AGVs based on pheromone concentrations.
2. **Path Cost Calculation**: During each iteration, the path taken by all ants is calculated. Task assignments are made based on the shortest paths traveled by the ants.
3. **Pheromone Update**: After each iteration, pheromone information is updated to guide ants towards converging on the optimal path.

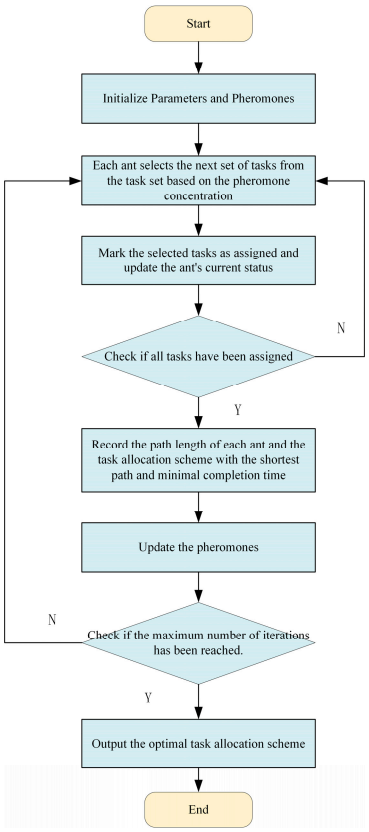


Figure 3. Traditional Ant Colony Optimization algorithm.

3.2. Improved Ant Colony Algorithm

In traditional ant colony algorithms, only the task of transporting all parts to the assembly line by AGVs is considered, while the priority of transportation tasks is neglected. This results in low task allocation efficiency. Such an approach may lead to delays for a single AGV while executing tasks, leaving other AGVs idle due to task dependency constraints. Delays for certain AGVs can further prolong the overall operation cycle (i.e., the time required to complete all tasks). To address this issue, this paper introduces an improved ant colony algorithm. is illustrated in the Figure 4 During pheromone updates, the transportation capacity of AGVs is considered, and task allocation integrates both path cost and task execution capability.

The task allocation process of the improved ant colony algorithm is illustrated in the flowchart, and the specific steps are as follows:

(1) Initialize Parameters: Set initial values, including the number of ants, pheromone importance, pheromone evaporation rate, pheromone release quantity, and the maximum number of iterations.

(2) Calculate Distances: Compute the distances between each AGV and task point, i.e., the distance from the current position of each AGV to the task point.

$$D(i,j) = \sqrt{\sum \left(AGV_{current}(i,:) - Task_{j,:}\right)^2} \tag{13}$$

In the formula, $AGV_{current}(i,:)$ represents the current coordinate position of the i-th AGV, $Task_j$ denotes the coordinate position of task point j, and $D(i,j)$ indicates the path cost between the current position of the i-th AGV and the task point j.

(3) Define the maximum transportation distance function for each AGV, and initialize the function using the corresponding formula.

$$AGV_{max_{distance(i)}} = maxD \tag{14}$$

In the formula, $AGV_max_distance(i)$ represents the maximum transportation distance of the iith AGV, and $maxD$ represents the maximum transportation distance set by the system for the AGV.

(4) Each ant selects the next set of tasks based on the AGV task sequence number. Then, each ant applies the formula.

$$AGV_ability_k(i) = AGV_ability_{k-1}(i) - Task_k \tag{15}$$

$AGV_ability_k(i)$ represents the remaining transportation capacity of the iith AGV in the kkth iteration. $AGV_ability_{k-1}(i)$ represents the remaining transportation capacity of the iith AGV in the $(k-1)(k-1)$th iteration. $Task_k$ is the transportation capacity required for the tasks currently assigned to the iith AGV.

(5) Determine whether all tasks have been assigned. If all tasks are assigned, calculate the total path length for each ant's allocation scheme and record the task allocation scheme with the minimum path cost. Otherwise, return to Step (16).

$$AGV_{path(i,j)} = 1 \tag{16}$$

$$all_{AGV_{paths_k}} = \left[all_{AGV_{paths_{k-1}}}; AGV_{path}\right] \tag{17}$$

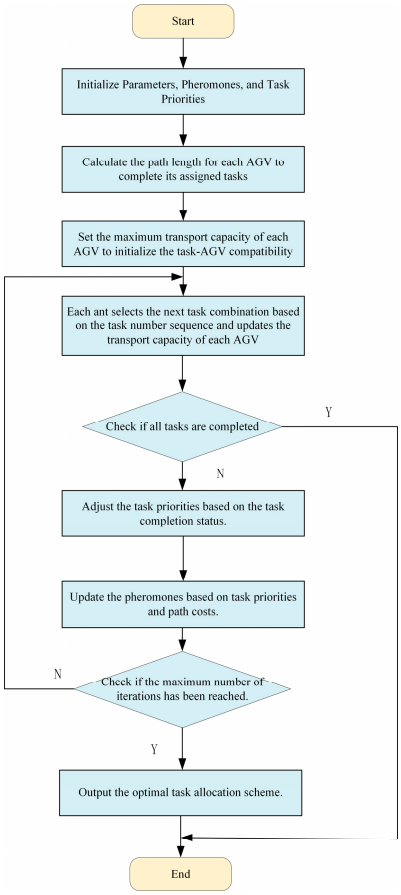


Figure 4. Flowchart of the improved Ant Colony Optimization algorithm.

4. Simulation Experiment and Analysis

This section employs three different algorithms to compare their convergence speeds and solution performance. The algorithms include the Improved Ant Colony Optimization (IACO), Genetic Algorithm (GA), and the standard Ant Colony Optimization (ACO). Computational simulations are conducted to verify the effectiveness of the improved hybrid algorithm. The experiments are carried out with eight AGVs, and the parameter settings are provided in Table 1.

The convergence curves of GA, ACO, and IACO are compared as shown in the figure below. The maximum completion time is illustrated in another figure, while the total travel distance of AGV operations and the empty travel distance are presented in subsequent figures.

From the optimization results in Figure 5, it can be observed that the Improved Ant Colony Optimization (IACO) algorithm outperforms both Ant Colony Optimization (ACO) and Genetic Algorithm (GA) in terms of convergence speed and solution quality. During the initial iteration phase, IACO demonstrates a significantly faster convergence rate, substantially reducing empty travel distance compared to ACO and GA. In contrast, ACO exhibits a slower convergence speed and is prone to incomplete convergence. While GA also shows a relatively fast convergence rate in the early stage, its final optimization performance falls short of that achieved by IACO.

In Figure 6, GA initially performs slightly better than IACO. However, as iterations progress, IACO's optimization capability becomes more evident, ultimately yielding a shorter total travel distance compared to both GA and ACO. ACO, on the other hand, demonstrates weaker optimization performance in this metric, characterized by slow convergence and incomplete optimization.

From the optimization performance in Figure 7, IACO again exhibits superior optimization efficiency. During the early solution process, GA and IACO perform similarly. However, in the middle and later iterations, IACO continues to improve task completion time, ultimately achieving better solutions than both GA and ACO. In contrast, ACO lags behind, showing slower optimization and failing to reach the optimal performance levels of GA and IACO.

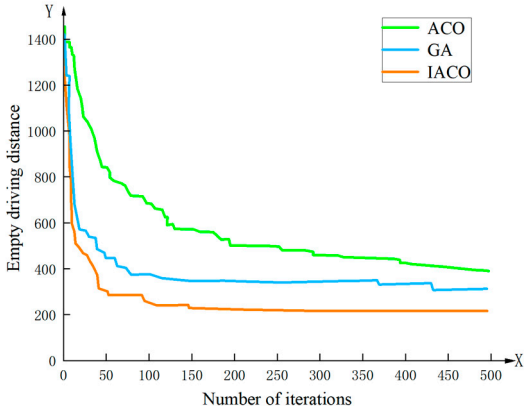


Figure 5. Empty travel distance comparison chart.

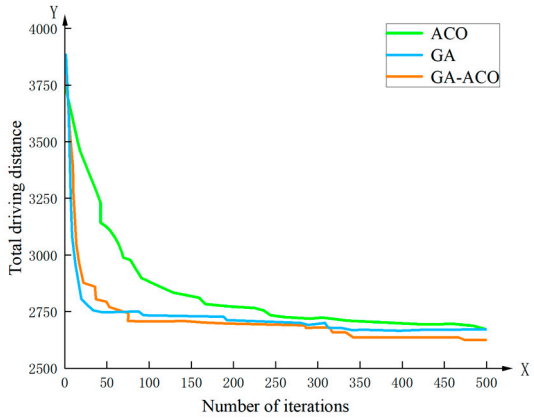


Figure 6. Total driving distance comparison chart.

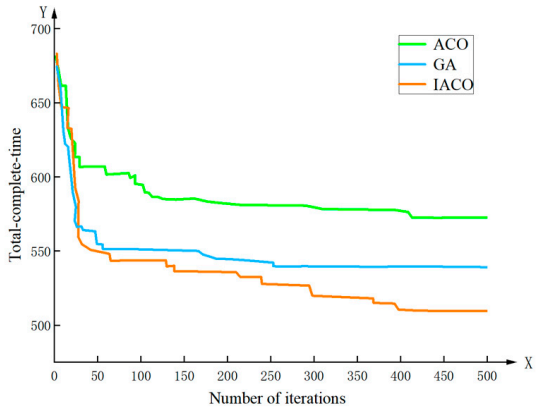


Figure 7. Total completion time.

Overall, the experimental results indicate that IACO outperforms GA and ACO across all three key metrics: empty travel distance, task completion time, and total travel distance.

To comprehensively evaluate the performance of the three algorithms, we utilized real-world data from a company. Tasks ranging from 5 to 20 were classified as small-scale

problems, with 2 to 7 AGVs assigned for scheduling. Tasks exceeding 100 were defined as large-scale problems, with 6 to 15 AGVs involved. For both small- and large-scale cases, six to seven task groups were randomly generated, with each group executed 10 times to compare task completion time and computation time. The results, presented in Tables 2 and 3, provide a clear illustration of AGV coordination efficiency, with computation time serving as a critical indicator of the scheduling system's computational performance.

From Table 2, when the number of tasks is 15 and four AGVs are deployed, IACO (Improved Ant Colony Optimization) achieves a task completion time of 390.30 s and a computation time of 6.50 s. In comparison, the GA and ACO algorithms result in task completion times of 408.50 s and 400.20 s, with computation times of 7.10 s and 6.80 s, respectively. Analysis of the small-scale cases shows that IACO reduces task completion time by an average of 4.46% and computation time by 8.45% compared to GA. Similarly, compared to ACO, IACO reduces task completion time by an average of 2.47% and computation time by 4.41%. While the differences in task completion times among the three algorithms are relatively minor for small-scale problems, GA demonstrates slightly better computation efficiency than the other two methods.

From Table 3, for large-scale cases where the number of tasks is 150 and 11 AGVs are assigned, IACO achieves a task completion time of 34,100 s and a computation time of 1060 s. In contrast, the GA and ACO algorithms produce task completion times of 36,900 s and 36,100 s, with computation times of 1150 s and 1200 s, respectively. Analysis of the large-scale results reveals that IACO reduces task completion time by an average of 7.58% and computation time by 7.83% compared to GA. When compared to ACO, IACO achieves an average reduction of 5.54% in task completion time and 11.67% in computation time. For large-scale problems, IACO demonstrates the ability to generate higher-quality scheduling solutions within a reasonable computation time, whereas ACO struggles to achieve globally optimal solutions due to its limited local search capability.

The comparative analysis highlights that IACO outperforms both GA and ACO in scheduling efficiency and computation time for both small- and large-scale problems. By incorporating task prioritization mechanisms, dynamically adjusting pheromone evaporation rates, and enhancing heuristic information, IACO significantly improves global search and local optimization capabilities. This leads to notable reductions in task completion time while maintaining reasonable computation times, making it an effective and efficient approach for AGV scheduling.

Parameter Tuning and Sensitivity Analysis

(1) Tuning methodology: Grid search ($\alpha \in \{1, 2, 3\}$, $\beta \in \{3, 5, 7\}$, $\rho \in \{0.5, 0.7, 0.9\}$) across 10 random obstacle scenarios in 20 × 20 maps determined the optimal combination ($\alpha = 2$, $\beta = 5$, $\rho = 0.7$), reducing average path length by 12.3% compared to untuned parameters.

(2) Sensitivity analysis—coupling effect of α and β: When $\beta = 5$ and $\alpha < 2$, the algorithm over-relies on heuristics, causing path length fluctuations of $\pm$15%; when $\alpha = 2$ and $\beta < 5$, pheromone dominance slows convergence by ~40%.

(3) Critical effect of ρ—$\rho = 0.7$ optimizes pheromone updating: $\rho < 0.5$ causes 35% of scenarios to loop due to rapid pheromone loss; $\rho > 0.7$ reduces new path exploration, decreasing optimal solution discovery by 28%.

(4) Trade-off between Nant and Q: Nant = 35 and Q = 500 balance computation and search precision—every 10 ants increases runtime by 25%, but Q = 500 compensates with pheromone concentration to reduce invalid searches.

Table 1. Parameters.

Parameter	Symbol	Value	Rationale
Number of Ants	N_{ant}	35	35 ants cover 20 × 20 maps efficiently, reducing computational overhead by 15%
Pheromone Importance Weight	α	2	α = 2 enhances pheromone guidance over α = 1/3, and balances with β = 5 to avoid local optima
Heuristic Information Weight	β	5	β = 5 strengthens goal orientation, shortening path length by ~8% vs. β = 3/7, requiring α = 2 for balance
Pheromone Evaporation Coefficient	ρ	0.7	Moderate evaporation reduces repeated exploration by 22% vs. ρ = 0.5/0.9 in 20 × 20 maps
Pheromone Release Amount	Q	500	Q = 500 creates clearer pheromone gradients, improving convergence speed by 30%
Maximum Number of Iterations	n	500	500 iterations ensure convergence in 20 × 20 maps, with <5% performance gain beyond n = 300

Table 2. Comparative analysis of small-scale cases.

Serial Number	Number of Tasks	Number of AGVs	IACO		GA		ACO	
			Computation Time (s)	Task Completion Time (s)	Computation Time (s)	Task Completion Time (s)	Computation Time (s)	Task Completion Time (s)
1	10	2	4.20	305.10	4.50	318.50	4.60	312.20
2	12	3	5.10	340.20	5.50	355.80	5.30	348.40
3	15	4	6.50	390.30	7.10	408.50	6.80	400.20
4	18	3	7.30	435.50	7.90	455.60	7.60	445.40
5	20	5	8.40	480.70	9.20	505.40	8.90	495.30
6	25	4	9.80	525.90	10.50	555.20	10.10	540.60
7	30	5	11.20	570.40	12.00	600.10	11.80	590.20
8	35	6	12.70	615.20	13.50	645.30	13.20	630.40
9	40	7	14.10	660.50	15.00	695.80	14.60	680.40
10	50	6	15.80	705.30	16.50	750.20	16.20	730.50

Table 3. Comparative analysis of large-scale cases.

Serial Number	Number of Tasks	Number of AGVs	IACO		GA		ACO	
			Computation Time (s)	Task Completion Time (s)	Computation Time (s)	Task Completion Time (s)	Computation Time (s)	Task Completion Time (s)
1	100	6	620	21,600	690	23,100	740	22,800
2	110	7	710	23,800	780	25,400	820	24,900
3	120	8	800	26,400	870	28,800	920	27,700
4	130	9	880	28,700	960	31,300	1010	30,500
5	140	10	970	31,200	1050	33,800	1100	33,200
6	150	11	1060	34,100	1150	36,900	1200	36,100
7	160	12	1150	36,800	1250	40,000	1300	38,900
8	170	13	1240	39,500	1350	43,100	1400	41,800
9	180	14	1340	42,600	1470	46,400	1520	44,900
10	200	15	1470	46,200	1600	50,300	1650	48,700

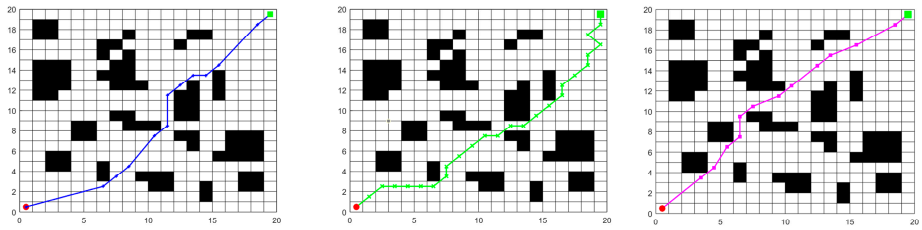
Simulation of Multi-AGV Path Planning

Although the first part of this study focused on optimizing the scheduling problem for multi-AGV systems using different algorithms, in practical applications, the performance of a scheduling system often relies heavily on the support of path planning strategies. To ensure AGVs can efficiently execute scheduling tasks, the selection and optimization of path planning algorithms are critical. To further validate the effectiveness of the proposed algorithm, we conducted detailed simulations and experiments on multi-AGV path

planning, with a focus on evaluating the reliability and efficiency of the algorithm during task execution.

To verify the feasibility of the proposed approach, Matlab 2024b was used to simulate the path planning for multi-AGV transportation tasks. The simulation was conducted on a computer running the Windows 11 operating system, equipped with an Intel Core i9 processor (2 GHz) and 8 GB of RAM.

The A* algorithm is a pathfinding algorithm used in graph space to find the shortest path from a start point to a goal point among multiple nodes. It combines the completeness of Dijkstra's algorithm with the efficiency of greedy best-first search, and guides the search direction by introducing a heuristic function, thus enabling efficient finding of the optimal path in many scenarios. A 20 × 20 grid map was constructed to simulate the layout of a production workshop, with obstacles added to the map. Red dots represent the start points, green dots represent the goal points, and the black areas represent obstacles, as shown in Figure 8.



(a) A* (b) ACO (c) IACO

Figure 8. Comparison of the performances of different algorithms.

In a 20 × 20 grid environment, the pathfinding performances of three algorithms are as follows: The Improved Ant Colony Optimization (IACO) demonstrates the best performance, with a path length of 27.56 and a running time of only 0.028 s. Benefiting from optimized pheromone update strategies and heuristic guidance, IACO balances path optimality and computational efficiency. The A* alg orithm, with a path length of 27.87, approaches IACO's optimization level but requires a longer running time (0.063 s), as its heuristic search traverses more nodes to ensure path optimality. The original Ant Colony Optimization (ACO) exhibits the longest path (28.31) and poorest optimization; although its running time (0.048 s) is faster than A*, its early-stage random exploration leads to circuitous paths. Overall, by integrating heuristic information, IACO breaks through the limitations of traditional ant colony algorithms, achieving a balance between path quality and operational efficiency in static scenarios, which provides a foundation for further testing in subsequent research, as shown in Table 4.

Table 4. Comparison of the performances of different algorithms.

Map Specification	Algorithm	Path Length	Running Time/s
20 × 20	A* Algorithm	27.87	0.063
	ACO	28.31	0.048
	IACO	27.56	0.028

A 21 × 21 grid map was constructed to simulate the layout of a production workshop, with obstacles added to the map. Taking the example of two AGVs simultaneously performing transportation tasks, the starting and destination points for each AGV were predefined. The global path planning results for the multi-AGV system are shown in Figures 9 and 10.

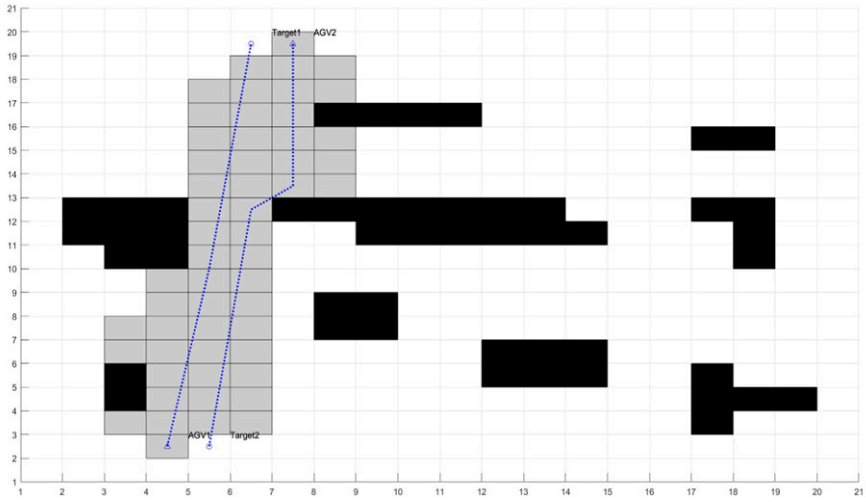


Figure 9. Path planning map.

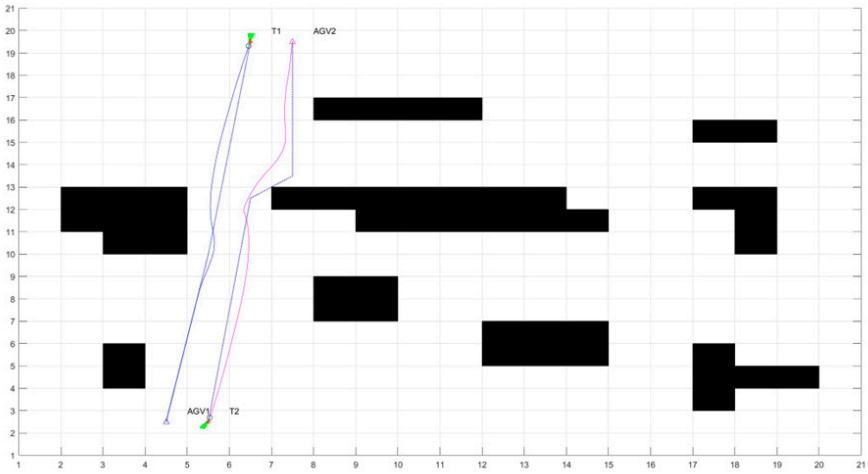


Figure 10. Actual path map.

The "AGV-Target path" in the figure represents the globally planned path for the AGV, with the black grids indicating unknown obstacles. After running for 105.36 s, both AGVs reach their target points, with the specific paths shown in the figure. As shown in Table 5, due to the influence of obstacles, the actual travel distance for AGV1 and AGV2 is slightly greater than the planned distance. From the above experimental results, it can be concluded that when the environment is relatively complex and contains various obstacles, the proposed improved fusion algorithm is capable of performing path planning for multiple AGVs. Based on global optimization, it allows the moving AGVs to quickly reach the designated target points.

Table 5. Multi-AGV path planning experiment in a physical environment.

Path Planning AGV	Path Planning Distance (m)	Motion Path Distance (m)
AGV1	16.46	25.38
AGV2	17.13	30.62

Due to terrain constraints, a 4 × 4 physical experimental platform was constructed. Obstacles were replaced with chairs and placed within a square grid. The copper-colored lines represent the boundaries of the obstacles, as shown in the figure. The starting point is located at the bottom-left corner, while the endpoint is at the top-right corner. The grid

sequence numbers range from 1 to 16, proceeding from left to right and from bottom to top. The algorithm mentioned above will be used to validate the feasibility of the experiment. A total of 10 physical experiments were conducted. Figure 11 illustrates the platform for the multi-AGV scheduling experiment. The figure also shows the experimental setup for the mobile robots.

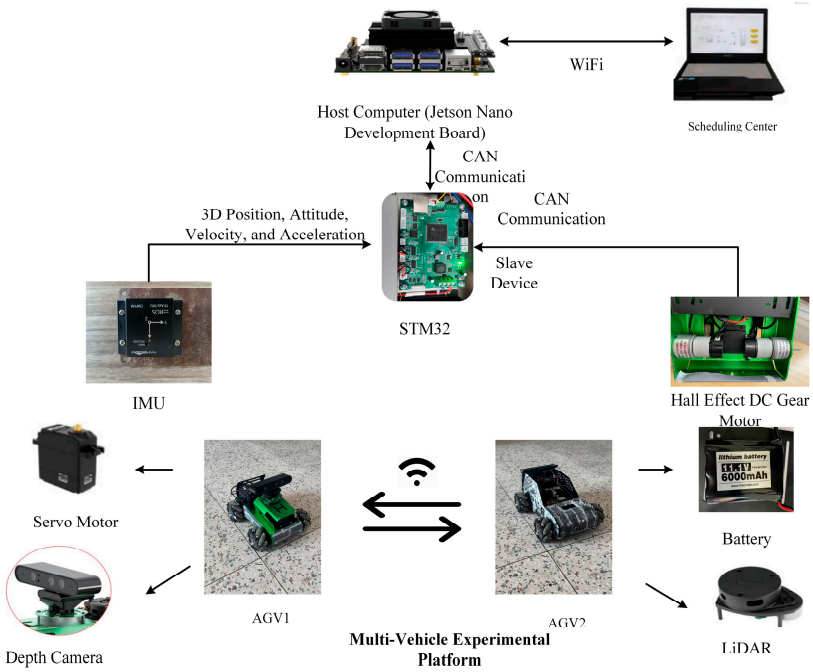


Figure 11. Multi-vehicle experimental platform.

To verify the path planning capability and coordination efficiency of the multi-AGV system in a dynamic environment, a simulated experimental scenario was constructed, and a map containing obstacles was designed, as shown in Figure 12. In the experiment, two AGVs started from different initial points and completed their respective tasks through path planning and obstacle-avoidance strategies. The experimental results are shown in Table 6.

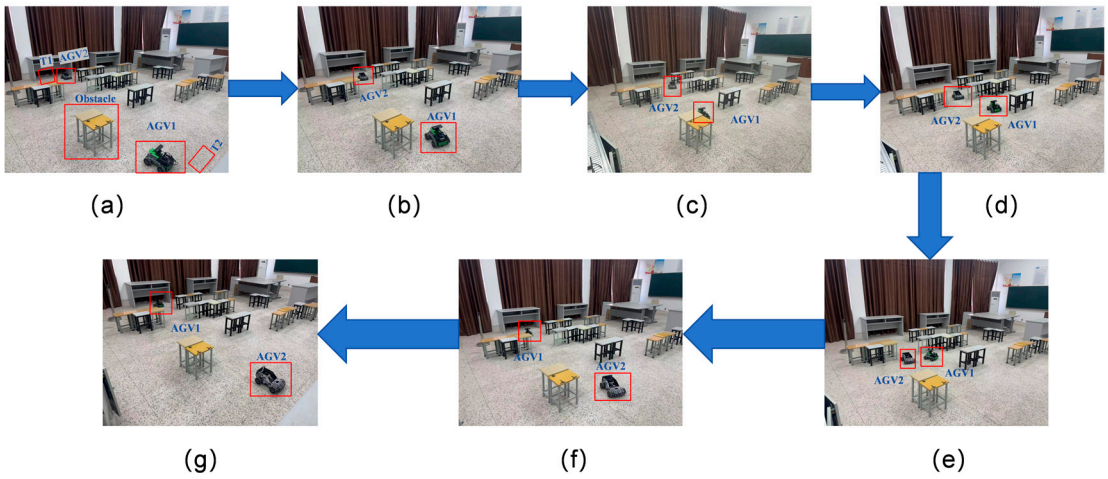


Figure 12. Multi-AGV path planning experimental setup comparison chart.

Table 6. Experimental data table.

Path Planning AGV	Path Planning Distance (m)	Motion Path Distance (m)
AGV1	3.82	4.52
AGV2	4.32	5.18

From the path planning results, it can be observed that the path planning algorithm used by the multi-AGV system exhibits good adaptability in avoiding both static and dynamic obstacles. In the experiment, when AGV1 approached the target area, it adjusted its original path upon encountering a dynamic obstacle. The real-time obstacle-avoidance strategy successfully mitigated potential collision risks. Similarly, AGV2 maintained a safe distance from AGV1 at the path intersection and replanned its path to avoid direct conflict. This demonstrates that the multi-AGV system is capable of achieving optimal path selection through information sharing and dynamic adjustment.

5. Conclusions

This paper proposes an integrated solution for multi-AGV scheduling in intelligent workshops. Unlike most existing studies and solutions that address AGV task allocation and path planning as separate problems, our approach integrates task assignment and scheduling into a unified framework. We establish a comprehensive optimization model aimed at minimizing total travel distance, idle travel distance, and overall task completion time. To solve this model, we develop an improved Ant Colony Optimization (IACO) algorithm with task prioritization capabilities. The algorithm adaptively adjusts task execution priorities by considering AGV transportation capability and path costs, and improves the pheromone update rules. A series of numerical experiments are conducted to evaluate the performance of the IACO algorithm, demonstrating its superior convergence speed. Additionally, we test the IACO algorithm under varying numbers of AGVs. Experiments show that the proposed model is practically applicable to existing automated terminals and significantly enhances scheduling efficiency.

Author Contributions: Conceptualization, Y.X. and W.L.; methodology, Y.X.; software, Y.X.; validation, Y.X., W.L. and H.Y.; formal analysis, Y.X.; investigation, Y.X.; resources, Y.X.; data curation, Y.X.; writing—original draft preparation, Y.X.; writing—review and editing, Y.X.; visualization, Y.X.; supervision, Y.X.; project administration, Y.X.; funding acquisition, Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data cannot be made publicly available upon publication because they contain sensitive personal information. The data that support the finding of this study are available upon reasonable request from the authors.

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

References

1. Wang, X.; Wu, W.; Xing, Z.; Chen, X.; Zhang, T.; Niu, H. A neural network based multi-state scheduling algorithm for multi-AGV system in FMS. *J. Manuf. Syst.* **2022**, *64*, 344–355. [CrossRef]
2. Lin, Y.; Xu, Y.; Zhu, J.; Wang, X.; Wang, L.; Hu, G. MLATSO: A method for task scheduling optimization in multi-load AGVs-based systems. *Robot. Comput.-Integr. Manuf.* **2023**, *79*, 102397. [CrossRef]
3. Zou, W.Q.; Pan, Q.K.; Wang, L.; Miao, Z.H.; Peng, C. Efficient multiobjective optimization for an AGV energy-efficient scheduling problem with release time. *Knowl.-Based Syst.* **2022**, *242*, 108334. [CrossRef]
4. Li, J.; Cheng, W.; Lai, K.K.; Ram, B. Multi-AGV flexible manufacturing cell scheduling considering charging. *Mathematics* **2022**, *10*, 3417. [CrossRef]

5. Hu, E.; He, J.; Shen, S. A dynamic integrated scheduling method based on hierarchical planning for heterogeneous AGV fleets in warehouses. *Front. Neurorobotics* **2023**, *16*, 1053067. [CrossRef]
6. Zou, W.-Q.; Pan, Q.-K.; Meng, L.-L.; Sang, H.-Y.; Han, Y.-Y.; Li, J.-Q. An effective self-adaptive iterated greedy algorithm for a multi-AGVs scheduling problem with charging and maintenance. *Expert Syst. Appl.* **2023**, *216*, 119512. [CrossRef]
7. Li, Z.; Sang, H.; Pan, Q.; Gao, K.; Han, Y.; Li, J. Dynamic AGV scheduling model with special cases in matrix production workshop. *IEEE Trans. Ind. Inform.* **2022**, *19*, 7762–7770. [CrossRef]
8. Zou, W.Q.; Pan, Q.K.; Tasgetiren, M.F. An effective iterated greedy algorithm for solving a multi-compartment AGV scheduling problem in a matrix manufacturing workshop. *Appl. Soft Comput.* **2021**, *99*, 106945. [CrossRef]
9. Maoudj, A.; Kouider, A.; Christensen, A.L. The capacitated multi-AGV scheduling problem with conflicting products: Model and a decentralized multi-agent approach. *Robot. Comput.-Integr. Manuf.* **2023**, *81*, 102514. [CrossRef]
10. Liu, L.; Qu, T.; Thürer, M.; Ma, L.; Zhang, Z.; Yuan, M. A new knowledge-guided multi-objective optimisation for the multi-AGV dispatching problem in dynamic production environments. *Int. J. Prod. Res.* **2023**, *61*, 6030–6051. [CrossRef]
11. Wang, K.; Liang, W.; Shi, H.; Zhang, J.; Wang, Q. Optimal time reuse strategy-based dynamic multi-AGV path planning method. *Complex Intell. Syst.* **2024**, *10*, 7089–7108. [CrossRef]
12. Bai, Y.; Ding, X.; Hu, D.; Jiang, Y. Research on dynamic path planning of multi-AGVs based on reinforcement learning. *Appl. Sci.* **2022**, *12*, 8166. [CrossRef]
13. Shen, G.; Ma, R.; Tang, Z.; Chang, L. A deep reinforcement learning algorithm for warehousing multi-agv path planning. In Proceedings of the 2021 International Conference on Networking, Communications and Information Technology (NetCIT), Manchester, UK, 26–27 December 2021; IEEE: New York, NY, USA, 2021; pp. 421–429.
14. Qiuyun, T.; Hongyan, S.; Hengwei, G.; Ping, W. Improved particle swarm optimization algorithm for AGV path planning. *IEEE Access* **2021**, *9*, 33522–33531. [CrossRef]
15. Yang, M.; Bian, Y.; Ma, L.; Liu, G.; Zhang, H. Research on traffic control algorithm based on multi-AGV path planning. In Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; IEEE: New York, NY, USA, 2021; pp. 697–702.
16. Lin, S.; Liu, A.; Wang, J. A Dual-Layer Weight-Leader-Vicsek Model for Multi-AGV Path Planning in Warehouse. *Biomimetics* **2023**, *8*, 549. [CrossRef]
17. Shan, H.; Wang, C.; Zou, C.; Qin, M. Research on pull-type multi-AGV system dynamic path optimization based on time window. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2021**, *235*, 1944–1955. [CrossRef]
18. Liu, W.; Chen, L.; Wang, R.; Wan, Y. Trajectory planning for AGV based on the improved artificial potential field-A* algorithm. *Meas. Sci. Technol.* **2024**, *35*, 096312. [CrossRef]
19. Farooq, B.; Bao, J.; Raza, H.; Sun, Y.; Ma, Q. Flow-shop path planning for multi-automated guided vehicles in intelligent textile spinning cyber-physical production systems dynamic environment. *J. Manuf. Syst.* **2021**, *59*, 98–116. [CrossRef]
20. Jiang, Z.; Zhang, X.; Wang, P. Grid-Map-Based Path Planning and Task Assignment for Multi-Type AGVs in a Distribution Warehouse. *Mathematics* **2023**, *11*, 2802. [CrossRef]
21. Zhang, Z.; Chen, J.; Zhao, W. Multi-AGV route planning in automated warehouse system based on shortest-time Q-learning algorithm. *Asian J. Control* **2024**, *26*, 683–702. [CrossRef]
22. Liu, W.; Wang, R.; Chen, L.; Wan, Y. Gradient projection-based trajectory tracking control for automatic guided vehicle. *J. Control. Decis.* **2024**, 1–13. [CrossRef]
23. Bahwini, T.; Zhong, Y.; Gu, C. Path planning in the presence of soft tissue deformation. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2019**, *13*, 1603–1616. [CrossRef]
24. Zhong, Y.; Shirinzadeh, B.; Yuan, X. Optimal robot path planning with cellular neural network. In *Advanced Engineering and Computational Methodologies for Intelligent Mechatronics and Robotics*; IGI Global: Hershey, PA, USA, 2013; pp. 19–38.
25. Zhou, Q.; Gao, S.; Qu, B.; Gao, X.; Zhong, Y. Crossover recombination-based global-best brain storm optimization algorithm for uav path planning. *Proc. Rom. Acad. Ser. A-Math. Phys. Tech. Sci. Inf. Sci.* **2022**, *23*, 207–216.
26. Hills, J.; Zhong, Y. Cellular neural network-based thermal modelling for real-time robotic path planning. *Int. J. Agil. Syst. Manag.* **2014**, *7*, 261–281. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.