International Core Journal of Engineering Volume 11 Issue 1, 2025
ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

Research on Multi AGV Routing Problem in Electrode Sheet Production Plant Considering Dynamic Unlocking

Wei Yang[a], Tongtong Chen[b], Zihan Zhang[c], and Xiaonan Zhang[d]

Shaanxi University of Science & Technology, Xi'an 710021, China

[a]yangwei@sust.edu.cn, [b]sust_ctt@163.com, [c]916298544@qq.com, [d]WLxn_2010@126.com

Abstract

This study delves into the problem of path planning for Automated Guided Vehicles (AGVs) in the electrode sheet production plant environment. Firstly, leveraging the grid method, a workshop environment model is constructed. An enhanced A* algorithm is devised, introducing a turning cost and a variable coefficient, effectively addressing issues of excessive path inflection points and lengthy computation times when dealing with large-scale environmental maps.For dynamic conflicts and deadlock issues within the local road network, two algorithms are introduced: the Dynamic Detection Algorithm based on Segment Reservation Policy (DD-SRP) and the Deadlock Resolution Algorithm based on Breadth-First Search (DR-BFS). A simulation experiment is designed and conducted. The experimental results underscore the superiority of the enhanced A* algorithm in terms of solution time and outcomes. Moreover, the proposed dynamic conflict and deadlock detection and resolution methods effectively achieve dynamic detection and deadlock resolution.

Keywords

Automated Guided Vehicle; Improved A* Algorithm; Routing Problem; Deadlock; Electrode Sheet Workshop.

1. Introduction

Automated Guided Vehicles (AGVs) are increasingly central to intelligent manufacturing systems, especially in Industry 4.0 settings. Their deployment improves efficiency and flexibility in logistics processes, particularly in challenging environments like electrode sheet workshops. However, AGVs face several operational challenges, including routing optimization, deadlock avoidance, and adaptive traffic management[1, 2, 3].

Routing optimization for AGVs often involves solving complex path-planning problems, where traditional algorithms like A* are widely applied. To overcome its limitations in dynamic environments, improved versions of A* have been developed, integrating features like dynamic cost functions and hybrid techniques. For example, combining A* with reinforcement learning has proven effective for adaptive and efficient routing[3, 4]. Advanced methodologies, such as time-space network formulations and metaheuristic approaches, enable conflict-free and energy-efficient paths in multi-AGV systems[2, 5].

Deadlock prevention remains a critical challenge, particularly in high-density and collaborative environments. Parallel algorithms and bi-level cooperative strategies have demonstrated success in resolving such conflicts by dynamically adjusting AGV paths and task schedules[1, 4, 7]. Additionally, intelligent traffic control models incorporating behavior trees and machine learning enhance AGV coordination and scalability, reducing downtime in busy production lines[3, 4].

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

These advancements collectively address the complex requirements of electrode sheet workshops, ensuring optimized material flow, higher throughput, and sustainable operations. Continued research in hybrid optimization methods and intelligent control algorithms will further refine AGV systems, driving innovation in smart manufacturing[7, 8].

2. Raster Method of Environmental Modelling

The raster method is used to establish the environmental map of the workshop, each grid can be indexed by a two-dimensional array, and at the same time, the attributes of each pixel can be expressed binarially, which makes the whole map easy to maintain and modify, 1 means that the grid AGVs can be accessed, and 0 is not accessed. The model assumptions are as follows:

1) The grid adopts four-way search, i.e., the AGV can only travel along the four directions of up, down, left and right;

2) At the same time, there can not be more than one AGV in a cell grid, and an AGV can not occupy more than two cell grids at the same time during travelling;

(3) When the AGV receives a task, the starting point and end point are known by default;

(4) By default, all location information in the workshop scene is known and reflected in the environment map;

5) All AGV road networks in the workshop are two-way single lane.

Based on the above assumptions, an environment model of the electrode wafer production workshop is constructed, including production equipment, loading and unloading buffer, obstacles, AGV road network and charging station. Taking the upper left corner of the workshop as the coordinate origin, a right-angle coordinate system is established, and the arrows on the road network identify the reachable directions of AGVs travelling on the road network. Due to the large size of the original workshop map, only part of the workshop map with a scale of 22*15 is shown here, which contains three processes of gravure, coating and cold pressing, as shown in Figure 1.

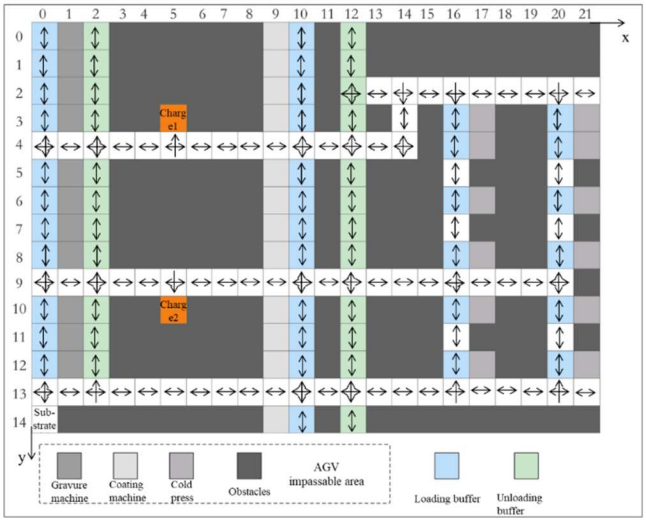


Figure 1. Hybrid flow shop environment modeling

3. Algorithm Design

3.1 Improvement of the A* Algorithm

The global static path planning problem for an electrode sheet production shop refers to planning an optimal path for an AGV that avoids static obstacles in a shop where all environmental elements are known.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

3.1.1 Reduction of Inflection Points in Mandate Implementation Routes

The basic A* algorithm is improved so that the solution minimises the number of turns on the route while ensuring the shortest distance travelled. The core of the control A* algorithm pathfinding is the function F(n) for Eq. It represents the path cost of the AGV travelling, where G(n) refers to the actual travelling distance between the starting point and the current position, and H(n) refers to the predicted travelling distance between the current position and the end point, which is calculated using the Manhattan distance. The heuristic function F(n) of A* is changed to add a new turn cost C. The value of C is summed with the value of G and updated with iterations to Gnew, i.e., Gnew= G(n)+C.

$$F(n) = G_{new} + H(n) \tag{1}$$

The calculation method of C-value is given. As shown in equation (4), the search is carried out according to the basic A* algorithm flow, assuming that the point with the smallest current F-value is obtained after one round of searching from the point (x, y), noted as (xmin, ymin), the point (x, y) is noted as the parent node, and the four-way search is continued on (xmin, ymin) for the searched child node, noted as (xcurrent, ycurrent).

$$a = \left|\left(x_{current} - x_{min}\right) - \left(x_{min} - x_{father}\right)\right| \tag{2}$$

$$b = \left|\left(y_{current} - y_{min}\right) - \left(y_{min} - y_{father}\right)\right| \tag{3}$$

$$C = \begin{cases} \lambda \cdot (a+b), (x\,min, y\,min)\, is\ not\ starting\ point \\ 0 \qquad\quad (x\,min, y\,min)\, is\ starting\ point \end{cases} \tag{4}$$

Where λ is the turn penalty factor, the value is too large to affect the optimality of the path, and the value is too small to play a punitive role, through the debugging of the actual scene map of the electrode production workshop, the final value of λ is obtained as [1, 1.5].

3.1.2 Dynamically Changing the Extent of the Search Space in Pathfinding

In order to increase the search speed of the A* algorithm without decreasing the search quality, a variable coefficient is added to the predicted distance H(n) in the heuristic function F(n). This coefficient will change dynamically with the distance from the current point to the end point, and the larger the distance, the smaller the coefficient.

Let the variable coefficient be α, which is a function of the predicted distance H, i.e. α=f(H). According to the actual map of the electrode sheet production plant, the distance distribution of loading and unloading between processes is obtained to be between 10 and 28 cell grids, so the design function f(H) is considered to be a segmented function, as shown in equation (5).

$$f(H) = \begin{cases} 1.5 & H \le 10 \\ 1 & 10 < H < 28 \\ 0.5 & H \ge 28 \end{cases} \tag{5}$$

In summary, the improved A* algorithm is to change the heuristic function on the basis of the basic A* algorithm in order to realise the reduction of path inflection points, dynamically change the search

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

range in the AGV pathfinding process, and improve the search speed of the A* algorithm without reducing the search quality.

3.2 Conflict Deadlock Resolution Algorithm

After planning the globally optimal path for each AGV, multiple AGVs may conflict or even deadlock in a certain area, especially under the layout of a two-way single-vehicle road network in a workshop, so the Conflict Deadlock CDRA algorithm is designed to avoid a long waiting time for AGVs, or even a deadlock situation.

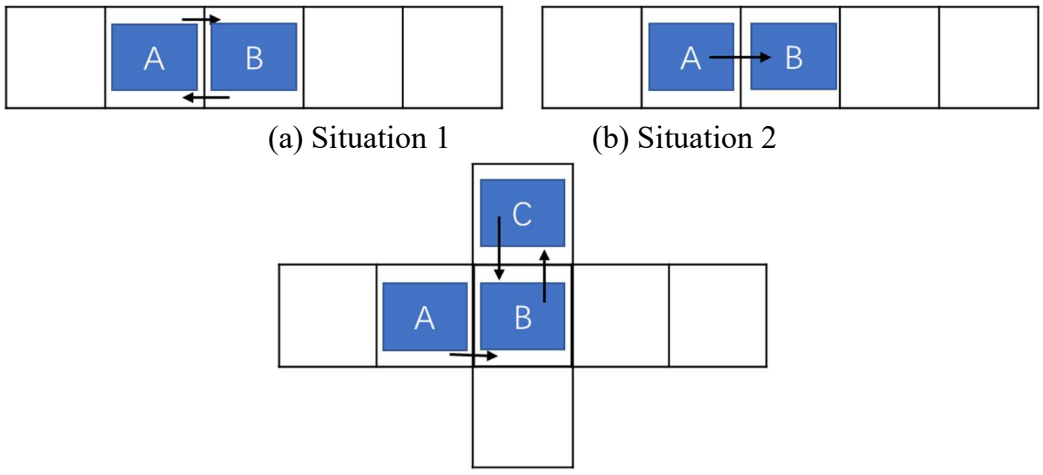
3.2.1 Dynamic Detection based on Segment Reservation Policy

Deadlocks often occur in the system in the case of cyclic waiting for resources, the following list of cases.

Scenario 1: Straight forward deadlock and both cars A and B have tasks. As shown in Figure 2(a), two cars A and B are travelling in opposite directions on a straight road, requesting the road network resources occupied by each other at the same moment, car A is blocked by car B, car B is blocked by car A, and cars A and B form a loop.

Scenario 2: straight deadlock and vehicle A has a task and vehicle B is stationary. there are three AGV states, AGV has a task in motion; AGV has no task and is stationary waiting for assignment; AGV is stationary but cannot move when loading/unloading or charging. As shown in Figure 2(b).

Scenario 3: Intersection, A, B and C vehicles all have tasks and two by two form a ring. As shown in Figure 2(c), vehicles B and C are deadlocked with each other, vehicle A is travelling to the right, and vehicle B can be regarded as a stationary vehicle relative to vehicle A. Therefore, vehicles A and B are deadlocked with each other again.



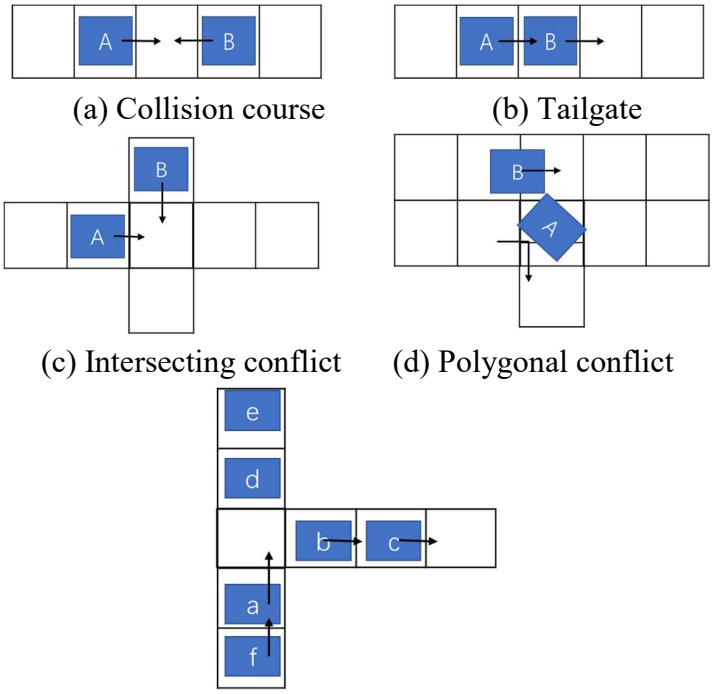
(a) Situation 1 (b) Situation 2

(c) Situation 3

Figure 2. AGV deadlock

Figure 3 shows five types of conflicts that may occur between AGVs, including phase-to-pace conflict, tailgating conflict, crossing conflict, polygonal conflict, and chain conflict. Since it is assumed that the AGV travels at a constant speed, the tailing conflict in Figure 3(b) will not occur, and the size of the grid cell in the electrode sheet production workshop is 1.5m*1.5m, while the length and width of the AGV is 1.00m*0.86m, respectively, and the radius of the circle covered by the AGV in the course of its rotation has been calculated to be 0.65m, so there will be no polygonal conflict, and therefore the situation in Figure 3(d) will not occur.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014



(a) Collision course (b) Tailgate

(c) Intersecting conflict (d) Polygonal conflict

(e) Chain of conflicts

Figure 3. AGV conflict

A dynamic deadlock detection algorithm based on segment reservation policy (DD-SRP) is proposed for deadlocks and conflicts among multiple AGVs.

Suppose a set of tasks T={td:1≤d≤R} are executed by an AGV system V={Ve:1≤e≤N}, the AGV road network can be partitioned into a number of non-overlapping zones (each cell grid in the map) Z={zi:1≤i≤M}, and the paths planned by the Improved A* algorithm for each AGV are referred to as guided paths GP=z1,z2,…,zi,zi+1,…,zs, and introduces a function OC:Z→V ∪ {0}, which indicates that the region zi is occupied by AGV ve if OC(zi)=ve, and indicates that the region zi is idle if OC(zi)=0. The guided path GP of each AGV is divided into different segments, denoted as SSl, the segments are based on the points with out-degree 3 in the route, and the segment capacity refers to the number of regions that make up each logical segment, denoted as |SSl|.

As shown in Figure 4, the guided path of this AGV is divided into two logical segments by a point with an out-degree of 3. The maximum segment capacity of the first segment is 9, and that of the second segment is 3. Segment reservation means that the AGV locks the free area in the logical segment from the current position, and the reserved area cannot be occupied by other AGVs, which is expressed as PR:V×{1,2,…}→{0,1}, the segment reservation cannot exceed the maximum capacity of this logical segment, i.e. $\sum_{ve=1}^{N} PR(v_e, l) \le |SS_l|$.

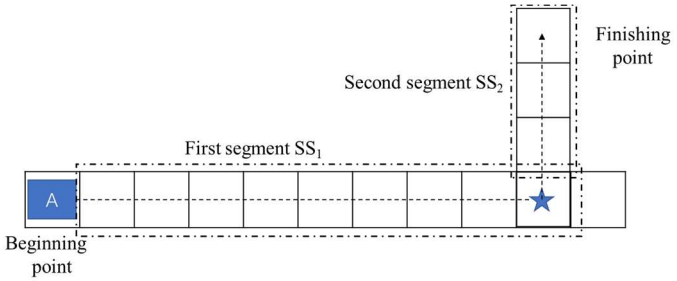


Figure 4. Division of sections

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

Each AGV in the system maintains a reservation table of its own, and the travelling process of each AGV is split into different one-step commands, and when all the AGVs in the system have completed one-step commands, they will update the reservation table to lock the new area and release the already travelled area from the reservation table. Based on this, the conflicts in Figure 3(a) and (c) can be avoided. Then for the conflict in Figure 3(e) it is additionally necessary to calculate the remaining capacity of the logical segment as in Equation (6).

$$R\left(SS_l\right)=C\left(SS_l\right)-I\left(SS_l\right)=Floor\frac{d_{ssl}}{L_{AGV}+S_F}-I\left(SS_l\right) \tag{6}$$

where R(SSl) denotes the remaining capacity of the logical segment SSl, C(SSl) denotes the maximum number of AGVs that can be travelled on the segment SSl, I(SSl) represents the number of AGVs on the current logical segment SSl, Floor() is a downward rounding function, LAGV denotes the length of an AGV, dSSl denotes the length of the logical segment, and SF denotes the minimum safe distance between two AGVs .

When all AGVs have finished updating the reservation table, the triggering system starts to detect deadlocks. Iterating through the current first region in all the reserved tables, it is found that two AGVs, A and B, request each other's resources at the same time, or the status of one of the AGVs is stationary, and the other is requesting resources, then it can be determined that the two AGVs have become a ring deadlock. The system records the content and number of loop detection, and triggers a loop detection whenever all AGVs complete a step of instruction.

3.2.2 Deadlock Resolution based on Breadth First Search

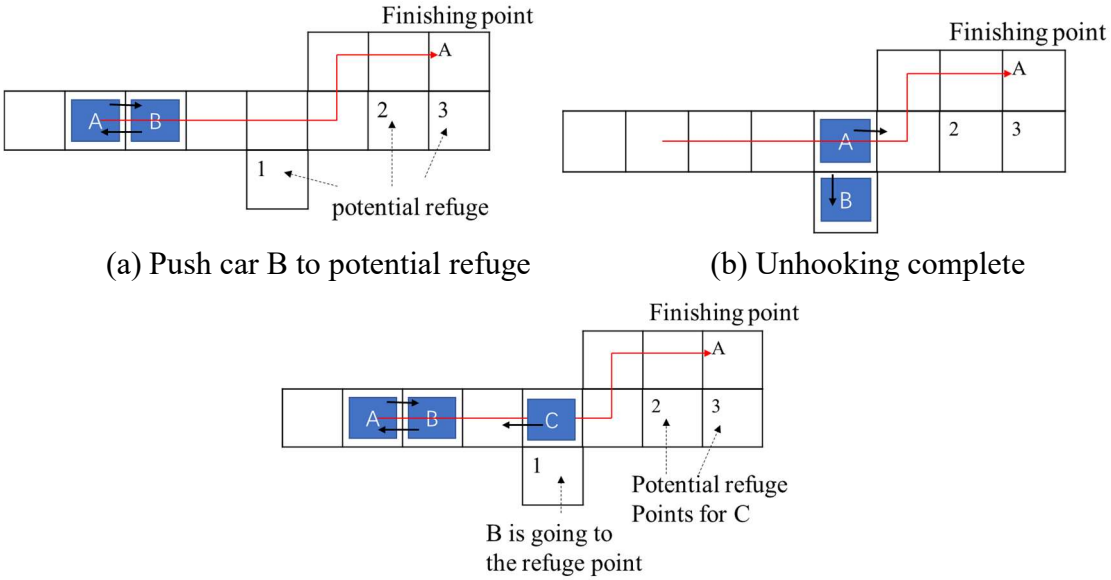
The idea of decoupling is considered as it is not limited by the size of the solution and the solution is faster. A deadlock solving algorithm (DR-BFS) based on graph theory search idea breadth first search is designed.

There are R AGVs in an H*W raster map G. Define R≤|V|-2, V is the number of cell grids in this raster map, and |V|=H×W, i.e., the method solves the problem of having at most n-2 AGVs in a graph with n cell grids. A task T, T:[1,n], let ∀i, j∈[1,n], T[i] ∈V. The start of the task is labelled S and the end goal is denoted E. The goal of the algorithm is to compute an optimal path π*={S,⋯,E} from S to E. The algorithm consists of a ‘push’ algorithm, which is a ‘push’ algorithm. The algorithm consists of two strategies, ‘push’ and ‘swap’: push means that when the system detects that two AGVs are deadlocked, one of them will be pushed out of the optimal path already planned by the other, and use the BFS to find an avoidance point that is closest to its current position, and then use the BFS to find a path that is closer to its current position than its current position. the nearest avoidance point and re-plan the path at that avoidance point. Once the AGV is unable to achieve its goal through the push operation, an exchange strategy is used to allow the two AGVs to exchange their positions along the shortest path.

Figure 5 shows the ‘push’ operation. A, B two vehicles deadlock, and B is the passive vehicle, then the use of ‘push’ operation will be pushed out of the A has been planned path, the completion of the loop. Call to improve the A * algorithm for the B vehicle to re-plan the path, B to the avoidance point of conflict with C, if the push operation is successful, then along the shortest path all AGVs will be pushed to the avoidance point outside the path.

Figure 6 shows the ‘swap’ operation, which consists of four steps: pushing multiple AGVs at the same time, clearing points, swapping, and reversing the mission path. In Figure 6, AGV B is a vehicle that has already reached the end point, but since A needs to go to the end point and the ‘push’ operation fails, it is necessary to exchange the positions of A and B with the exchange operation. After the swap is performed, the sequence of actions π computed during the simultaneous push and clear point operations must be reversed to ensure that the AGV in set U, i.e., AGV B, is able to return to its original end position at the end of the swap.

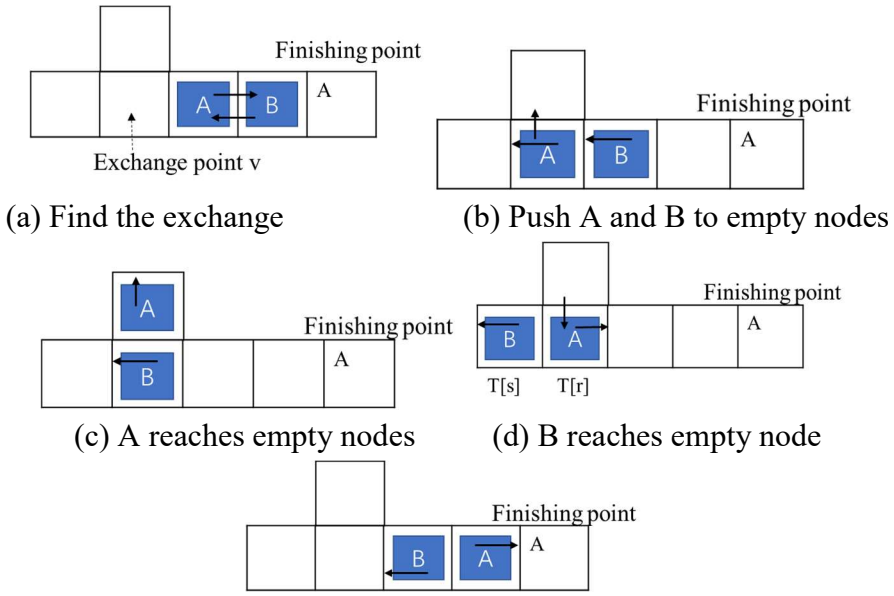
ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014



(a) Push car B to potential refuge (b) Unhooking complete

(c) Avoid one by one

Figure 5. "Push" operation



(a) Find the exchange (b) Push A and B to empty nodes

(c) A reaches empty nodes (d) B reaches empty node

(e) A and B exchange completed

Figure 6. "Swap" operation

The overall architecture of the DR-BFS algorithm is as follows: first set the currently assigned task T as the starting assignment S and insert it as a sequence of tasks into the optimal path π*. Set the set of AGVs that have reached the goal E to U, and initialise it to the empty set. When the system detects the existence of an AGV ring deadlock, it first searches for the passive vehicle r among the two AGVs through the push operation, the passive vehicle refers to the AGV among the two AGVs that has a longer remaining travelling distance from the end point and pushes it out of the optimal path that has been planned by the active vehicle for avoidance, and initiates the exchange operation if the push operation of the rth AGV fails, and the deadlock is damned if the exchange operation also fails cannot be resolved. If the rth AGV does not reach the goal point, then the algorithm applies the push and swap operations cyclically until r reaches the end point E[r], which will be inserted into the set U . Eventually, the algorithm returns the solution π*.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

4. Simulation Experiment

The experiments were all written in Python 3.9, running on macOS Monterey version 12.1, Apple M1 chip, and 16GB of RAM.

4.1 Experiments on Improving the A* Algorithm

Taking the raster map of electrode sheet production workshop as a scene, the basic A* algorithm, IA* algorithm, ACO algorithm and Dijkstra algorithm are compared in terms of the number of turns, time consumed to run 100 times, and the probability of finding the optimal path (the first and second columns are obtained from the map environment of Figure 1 as well as from randomly generating raster maps of 50 different environments, respectively), and the results are shown in Table 1.

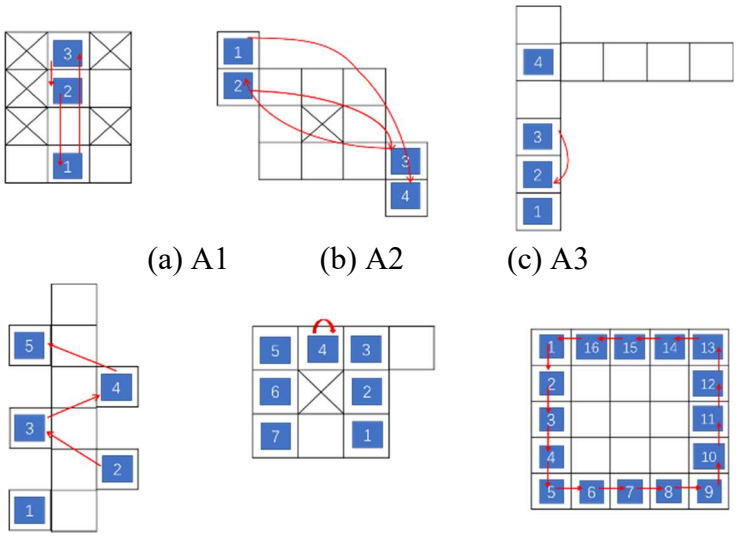
Table 1. Solution time and solution length of benchmark

Algorithm	Number of turns	Running time/s for 100 runs	Finding the optimal path probability	
A*	3	31	100%	100%
IA*	2	29	100%	100%
Dijkstra	3	40	100%	99.8%
ACO	7	303	100%	99.0%

From Table 1, it can be seen that the proposed IA* algorithm has the least number of turns and the computational time is slightly smaller than the A* algorithm and Dijkstra algorithm, and significantly smaller than the ACO algorithm, and the probability of finding the optimal path is smaller than that of the A* and IA* algorithms for the Dijkstra algorithm and the ACO algorithm. As a result, the designed IA* algorithm can effectively reduce the number of turns and improve the search speed of the A* algorithm without reducing the search quality.

4.2 Conflict Deadlock Resolution Algorithm Experiments

The feasibility of the CDRA algorithm is verified on benchmark arithmetic problems. Six benchmark arithmetic problems with multi-AGV cooperative wayfinding are used as shown in Figure 7, which are characterised by small map sizes and large deadlock task solution challenges. The results of the proposed deadlock dynamic detection and resolution algorithm are compared with two typical deadlock solving methods (coupled A* algorithm, decoupled WHCA* algorithm).



(a) A1 (b) A2 (c) A3

(d) A4 (e) A5 (f) A6

Figure 7. Benchmark numerical example

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

Table 2 shows the calculation results of the three methods, in which T represents the solution time, L represents the path length, and ∞ indicates that the optimal solution cannot be found within one hour. The proposed conflict deadlock solving algorithm has similar path lengths compared to the coupled A* algorithm, and in the solution time, except for the solution time of A1, which is slightly slower than the coupled A* algorithm, the other solution times are all better than that of the coupled A* algorithm, and when the number of AGVs is increased to 7, the coupled A* algorithm cannot give the solution result anymore, but the deadlock can still be solved in a shorter time. When the number of AGVs increases to 7, the coupled A* algorithm is no longer able to give solution results, but the conflict deadlock solving algorithm is still able to solve the deadlocks in a shorter time.

Table 2. Solution time and solution length of benchmark

Problem	Coupled A* algorithm		Decoupled WHCA* algorithm		CDRA algorithm	
	T/s	L/m	T/s	L/m	T/s	L/m
A1	2.9	15	0.7	34	3.7	18
A2	3590	45	0.6	40	12.7	46
A3	41.7	53	∞	—	16.7	38
A4	208.5	20	1.1	36	18.1	40
A5	∞	—	∞	—	32.2	45
A6	∞	—	310.0	18	60.2	15

Although the WHCA* algorithm takes less time than the CRDA algorithm, the WHCA* algorithm is less generalisable, as both A3 and A5 are characterised by a large number of AGVs but few drivable areas in the map, i.e., there is a limited amount of space to unlock the loops, which leads to the failure of the WHCA* algorithm to unlock the loops.The CRDA algorithm solves the different cases, and the quality of the solution is determined by the number of 'swapping' operations that have to be performed in order to solve the problem. The quality of the solution for different cases solved by the CRDA algorithm depends on the number of 'swapping' operations that must be performed to solve the problem; the more swapping operations that need to be performed, the longer the path.

4.3 Simulation Verification of Electrode Sheet Production Workshop Scenarios

In order to verify the application of the proposed improved A* algorithm followed by the conflict deadlock resolution algorithm in practical scenarios, simulation experiments are conducted using the actual map shown in Figure 2 of the electrode wafer production plant.

4.3.1 Verification of Algorithmic Superiority

Firstly, the number of AGVs is fixed to 10, and the average travelling speed of AGVs is 1.5m/s. Changing the number of tasks, the tasks are evenly distributed to each AGV, and the results of the comparison of the three different strategies are shown in Table 3 and

Table **4**. M represents the number of collisions, and T_{total} represents the time of completion of all task transports, in which GAP1, and GAP2 denote the percentage difference of the task completion times using IA*+CDRA with respect to IA*, IA*+DCS the percentage difference in task completion time. ∞ indicates that the deadlock between AGVs cannot be unlocked and the completion time is infinite.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

Table 3. Comparison of IA* and IA*+CDRA simulation experiments

Number of tasks	IA* time/s	IA*+CDRA		GAP1(%)
	T_{total}/s	M/times	T_{total}/s	
40	220	26	226.5	2.95
60	343	48	360.0	4.96
80	434	85	457.5	5.41
100	574	109	601.5	4.79
120	763	143	811.5	6.36
180	1132	204	1207.5	6.67
240	1895	266	2002.5	5.67
300	2505	310	2715.5	8.40

Table 4. Comparison of IA*+CDRA and IA*+DCS simulation experiments

Number of tasks	IA*+DCS		IA*+CDRA		GAP2(%)
	M/times	T_{total}/s	M/times	T_{total}/s	
40	31	241.0	26	226.5	-6.02
60	54	385.5	48	360.0	-6.61
80	90	∞	85	457.5	—
100	118	627.0	109	601.5	-4.07
120	148	839.5	143	811.5	-3.34
180	208	1219.0	204	1207.5	-0.94
240	240	∞	266	2002.5	—
300	301	∞	310	2715.5	—

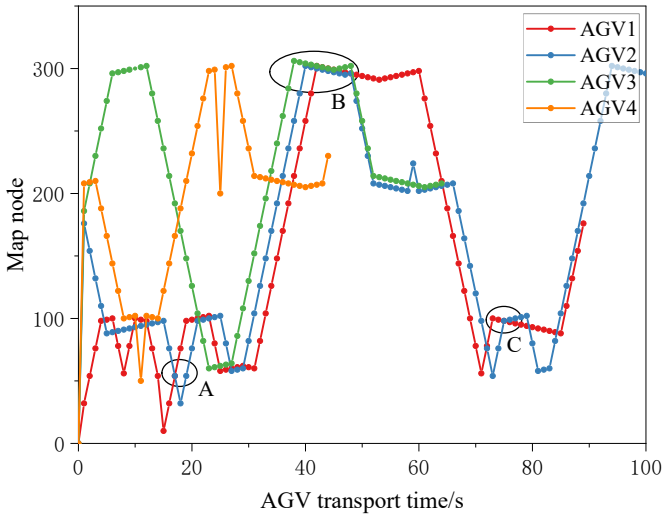
Analysis of GAP1 and GAP2 reveals that under different task sizes, the average difference in completion time between the IA*+CDRA method and the IA* algorithm is 6%, and the IA*+CDRA method reduces the task completion time by about 3.73% compared with the IA*+DCS method, and finds a better path scheme under different sizes, which indicates that the IA*+CDRA method is more universal in unlocking deadlocks and demonstrates the superiority of the algorithm.

4.3.2 Verifying the Effectiveness of Algorithmic Conflict Deadlock Resolution

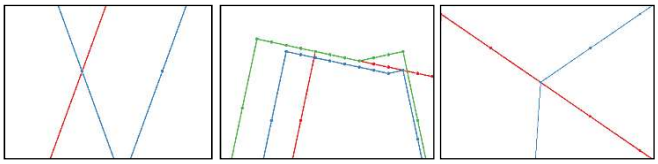
With 4 AGVs, 24 tasks and 6 tasks assigned to each AGV, Figure 8 shows global path planning using only the IA* algorithm, without considering conflict and deadlock resolution among AGVs in the local road network. It is found that the AGVs are travelling with a total of 4 conflicts, one at A, two at B, and one at C.

The conflict-free deadlocked multi-AGV travelling path planned using the IA*+CDRA algorithm is shown in Figure 9. It can be seen that there is no more conflict and deadlock between the 4 AGVs, but additional segment reservation and deadlock detection are required, and their task completion time is slightly increased compared to Figure 9. However, it is possible to plan an effective conflict-free deadlock path planning scheme in the studied electrode sheet production plant scenario.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014



(a) 4 AGV travelling paths



(b) A (c) B (d) C

Figure 8. AGV path without considering conflict and deadlock

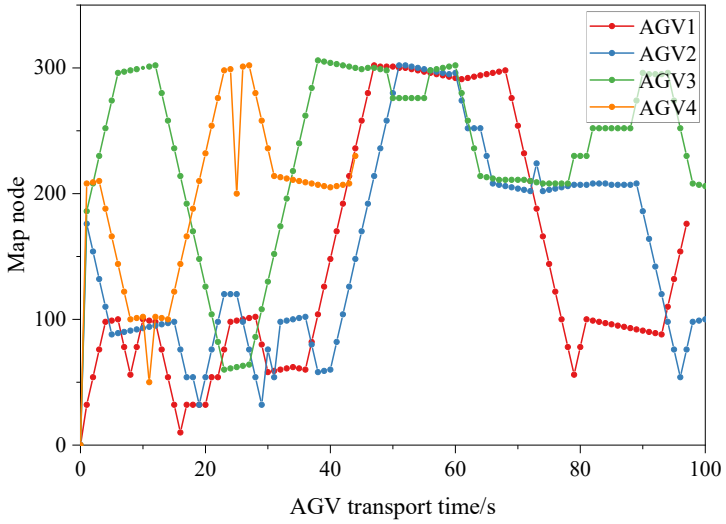


Figure 9. AGV path considering conflict and deadlock

4.3.3 Sensitivity Analysis

Sensitivity analysis of the number of AGVs is carried out in a 22*15 electrode sheet production plant. Figure 10 and Figure 11 represent the plots of task completion time and deadlock number with the number of AGVs, respectively. Under different task sizes, when the number of AGVs increases from 2 to 8, the task completion time decreases significantly and the number of occurrence of ring-forming deadlock increases slowly; when the number of AGVs increases from 8 to 18, the task completion time increases slowly and the number of occurrence of ring-forming deadlock increases significantly; when the number of AGVs increases from 18 to 20, both the task completion time and the number of occurrence of ring-forming deadlock increase rapidly.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

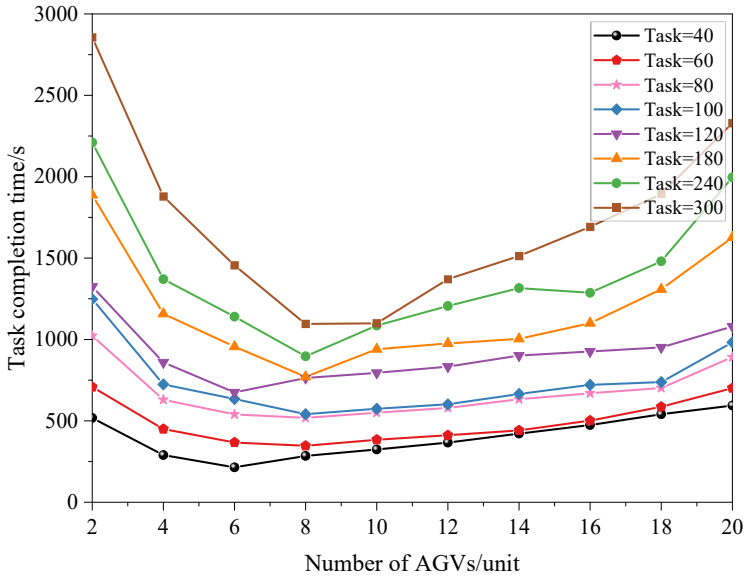


Figure 10. Change of task completion time with AGV number

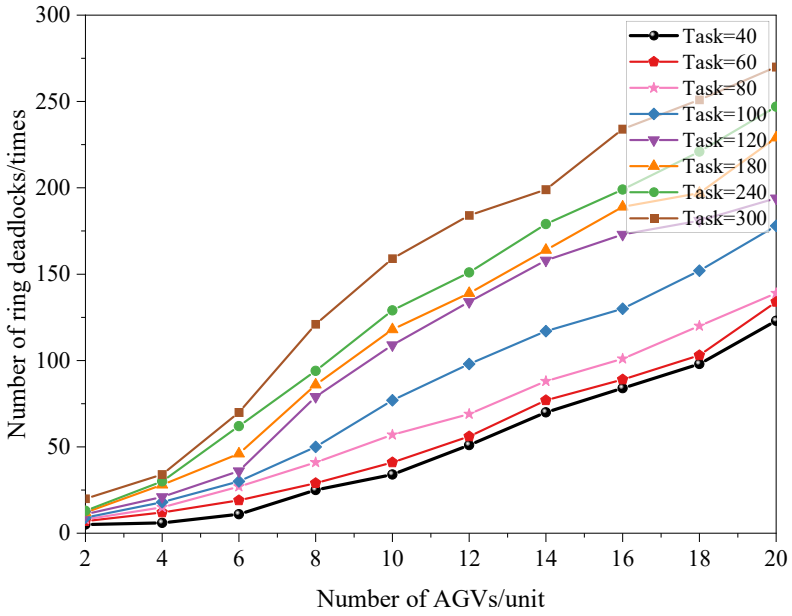


Figure 11. Change of deadlock times with AGV number

In the case of a certain scale of workshop tasks, increasing the number of AGVs, although each AGV performs fewer tasks, but the interference between AGVs is serious, and it takes extra time to solve the deadlock, so when the number of AGVs in the workshop to a certain extent, the bottleneck of the system is no longer the AGVs, and then increase the input of the AGVs has little effect on reducing the material distribution time and improving the efficiency of the system.

Figure 12 plots the average task completion time of AGVs (denoted as AT) versus the marginal time benefit of AGVs (reduction in completion time per AGV invested, denoted as MT) as an example with the number of tasks equal to 300. AT is equal to the system completion time divided by the number of AGVs, with a smaller AT indicating higher productivity.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

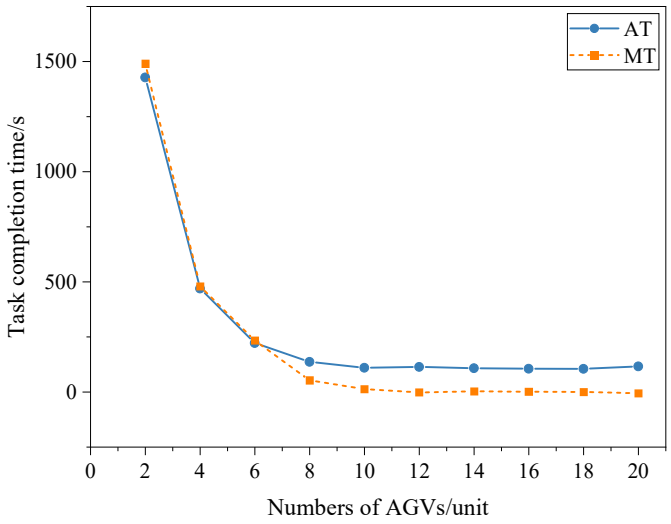


Figure 12. AGV optimal input quantity analysis (task number = 300)

From Figure 12, it can be seen that when AGV<6 vehicles, the folding line declines steeply, indicating that increasing the number of AGVs before then can effectively improve the productivity of the workshop, although MT is also declining rapidly in the process, MT is always greater than AT.

When 8<AGV<10, the AT folding line decreases slowly, MT<AT, at this time investing in AGVs brings only a small benefit.

When AGV>10 vehicles, AT line is slowly rising trend, MT<AT, this time and then put into the AGV not only can not bring benefits, but also caused by the workshop production speed slows down.

When the number of AGVs is between 6 and 8, the marginal time benefit of AGVs reaches the critical value, and the intersection of the two folding lines occurs, and the number of AGVs corresponding to this point is the optimal number of inputs, and since the number of AGVs must be taken as an integer, the optimal number of configurations is 6, which can be used as a reference for the decision-making of the enterprise.

5. Conclusion

In the path planning problem of multiple AGVs in the electrode sheet production workshop, the dynamic detection and resolution of conflict deadlocks that have already appeared in the road network are studied, for which the introduction of the turn cost C with variable coefficients is proposed to improve the A* algorithm, the IA*+CRDA algorithm based on the segment reservation strategy and based on the breadth-first search, which is capable of avoiding conflicts, and the dynamic deadlock detection and unlocking. Compared with the popular coupled A* algorithm, WHCA* algorithm, and delayed collision avoidance strategy, the solution problem has wider applicability and faster deadlock release. Finally, simulation experiments are designed to verify the effectiveness, universality and superiority of the improved A* algorithm and the proposed algorithms for solving conflicts and deadlocks, and to analyse the impact of the number of AGVs on the efficiency of the system's task execution under different task sizes, which provides ideas for the efficient utilisation of the AGV system.

However, due to the complexity of the problem, the subsequent research work should consider for multi-AGV conflict-free and deadlock path planning in The acceleration and deceleration changes of AGVs may bring new types of conflicts between AGVs, and how to solve conflicts and deadlocks between AGVs when considering AGV motion velocity profiles.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202501_11(1).0014

Acknowledgments

This research was supported by the National Key Research and Development Programme of the 14th Five-Year Plan (Project No. 2022YFD2100601) and the General Project of Humanities and Social Sciences Research of the Ministry of Education (Project No. 23XJCZH019). Their funding and support were essential in the completion of this work.

References

[1] D. D. Yu, X. L. Hu, K. W. Liang, et al. A parallel algorithm for multi-AGV systems, Journal of Ambient Intelligence and Humanized Computing, Vol. 13 (2022) No. 4, p. 2309-2323.

[2] J. B. Xin, L. Q. Wei, A. D'ARIANO, et al. Flexible time-space network formulation and hybrid metaheuristic for conflict-free and energy-efficient path planning of automated guided vehicles, Journal of Cleaner Production, Vol. 398 (2023) No. 2, p. 1-24.

[3] H. Hu, X. Jia, K. Liu, et al. Self-adaptive traffic control model with behavior trees and reinforcement learning for AGV in Industry 4.0, IEEE Transactions on Industrial Informatics, Vol. 17 (2021) No. 12, p. 7968-7979.

[4] B. Wu, Z. Xu, L. Chen, et al. Dynamic path planning for forklift AGV based on smoothing A* and improved DWA hybrid algorithm, Sensors, Vol. 22 (2022) No. 18, p. 7079.

[5] Y. Zhang, H. Luo. Hybrid routing and scheduling for multi-AGV systems in smart manufacturing, Computers & Industrial Engineering, Vol. 178 (2023), p. 108999.

[6] J. Zhang, Z. Li, L. Li, et al. A bi-level cooperative operation approach for AGV based automated valet parking, Transportation Research Part C: Emerging Technologies, Vol. 128 (2021) No. 6, p. 103140.

[7] S. Kim, C. Park, K. Lee. Bi-level path planning algorithm for multi-AGV routing problem, Electronics, Vol. 10 (2021) No. 9, p. 1123.

[8] D. M. RYCK, M. VERSTEYHE, F. DEBROUWERE. Automated guided vehicle systems, state-of-the-art control algorithms and techniques, Journal of Manufacturing Systems, Vol. 54 (2020), p. 152-173.