Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 2239342, 14 pages
https://doi.org/10.1155/2022/2239342

Research Article

Research on Optimization of the AGV Shortest-Path Model and Obstacle Avoidance Planning in Dynamic Environments

Ruixi Liu

Chongqing Police College, No. 666 Jing Zheng Road, Chongqing, China

Correspondence should be addressed to Ruixi Liu; dianqi206@outlook.com

Received 17 May 2022; Revised 8 June 2022; Accepted 20 June 2022; Published 19 July 2022

Academic Editor: Xuefeng Shao

This paper proposes a support vector machine (SVM)-based AGV scheduling strategy that enhances the scheduling efficiency of automated guided vehicles (AGVs) in intelligent factories. The developed scheme optimizes the task area division process to endow the AGVs with the ability to avoid obstacles in complex dynamic environments. Specifically, given the two AGV motion cases, i.e., towards a single target point and multiple target points, the optimal path was determined utilizing the exhaustive and the Q-learning methods, while path optimization was realized by utilizing different schemes. Based on the shortest path obtained, a nonlinear programming model with the shortest time as the objective was built, and the AGV's turning path was proved to be optimal by the non-dominated sorting genetic algorithm (NSGA-II). Several simulation tests and calculation results validated the proposed method's effectiveness, highlighting that the developed scheme is a rational solution to the obstacle congestion and deadlock problems. Moreover, the experimental results demonstrated the proposed method's superiority in path planning accuracy and its ability to respond well in complex dynamic environments. Overall, this research provides a reference for developing and applying AGV cluster scheduling in real operational scenarios.

1. Introduction

Traditionally, warehouses have been mostly managed manually. With the rise of e-commerce [1], s (AGVs), shuttles, and Delta sorting robots [2] are playing an essential role in automated warehouse logistics systems. Therefore, to ensure proper automated warehouse logistics system management, achieving reasonable obstacle avoidance of AGVs in complex smart warehouses is necessary to reach optimal scheduling. Some scholars have tried solving the scheduling of AGVs using path planning algorithms, such as Dijkstra's algorithm [3], A $*$ algorithm [4], and ant colony algorithm [5]. However, as the task dimensionality increases, the solution takes more time and becomes more complicated. Moreover, these algorithms have drawbacks, such as slow convergence and a tendency to fall into locally optimal solutions. Besides, the methods only focus on avoiding obstacles and do not consider the impact of local obstacle avoidance planning on subsequent operations. Consequently, the trajectory should be adjusted after obstacle avoidance to bring the AGVs back to the global path [6], reducing operational efficiency.

AGV assignment in scheduling problems has been studied by some researchers. To minimize the makespan and intercellular motions of components, Azadeh et al. [7] developed a nonlinear CFP which included intra-cell scheduling and material handling using AGVs. However, on the other hand, nonlinear CFP is only suited for small-scale AGV systems since it cannot adjust dynamically to the transportation environment. Chu et al. [8] used an adaptive memetic differential search method to tackle the problems, which included cross-training with a learning/forgetting impact that improves the flexibility of routing. However, the method cannot quickly determine the optimal solutions of multiple objective functions, limiting instantaneity and global optimality in large-scale transportation. To address the work assignment issue of AGVs, Radhia et al. [9] presented a hybrid method based on the Dijkstra algorithm, genetic algorithm, and heuristic algorithm, which can ensure conflict-free control of a large fleet on any layout, and that permits optimized routing for all AGVs' schedules. For AGV collision and deadlock complications, Malopolski [10] devised a novel approach to determine one-way, two-way, or multilane flow fields, which can adapt AGV control techniques in real time to the mobility environment. However, this method cannot motivate AGVs to arrive at destinations as rapidly as possible. In the meantime, it does not completely overcome the restrictions of the disadvantages in AGV control. Through the framework of a time-windows graph, Kim and Jin [11] used Dijkstra's shortest-path method to design AGV's course. The vehicle agent optimizes the distribution of transportation for AGVs and improves efficiency. A multi-AGV A $*$ algorithm based on a collision-free dynamic route planning approach was described by Chunbao Wang et al. [12]. The method categorized probable conflicts in order to find the shortest route that is conflict-free. The method classified potential conflicts in order to find the shortest conflict-free route. Similarly, Tai et al. [13] introduced a priority route planning method and achieved coordinated management of multi-AGVs based on time frames, which contributes to the conflict-free routing and shorter completion time, which contributes to the conflict-free routing and shorter completion time.

Considering the characteristics of existing AGV systems, this paper develops an AGV scheduling strategy based on the nonlinear programming model and the non-dominated sorting genetic algorithm NSGA-II [14]. Furthermore, obstacle avoidance simulations were conducted to minimize the total AGV moving path by optimizing the model so that the sorting stations reached a reasonable balance. Our trials concluded that the AGV path was optimal for the minimum radius and the circle center located at the obstacle's vertex (the circular obstacle was located at the circle's center), thereby avoiding obstacle congestion and deadlock in the current AGV scheduling [3].

The main work of this paper are as follows:

(1) Proposing an SVM-based AGV scheduling strategy that enhances the scheduling efficiency of AGVs in intelligent factories,

(2) Optimizing the division process to endow the AGVs with the ability to avoid obstacles,

(3) Determining the optimal path by utilizing the exhaustive and the Q-learning methods,

(4) Proving the optimal path by the non-dominated sorting genetic algorithm (NSGA-II),

(5) Providing a reference for developing and applying AGV cluster scheduling in real operational scenarios.

The structure of the remaining sections is as follows. Section 2 includes the problem statement. Section 3 describes the considered models and computational results of the study. Section 4 provides a discussion of the results and validation analyses. The concluding remarks and further research directions are provided in Section 5.

2. Problem Description and Hypotheses

2.1. Warehouse Information and Problem Description. The goal of scheduling is to assign handling tasks to s (AGVs) in different locations so that the total handling time is minimized without collision or deadlock. The unmanned warehouse discussed in this paper is a 32×22 rectangular area, simplified in Figure 1.

The node types are as follows.

(1) Path node (gray): AGV can pass freely.

(2) Storage node (green): Place pallets or ordinary shelves.

(3) Reserved node (yellow): Reserved position.

(4) Column node (black): Obstacle.

(5) Work station node (blue): AGVs pack the goods at the work station and exit from the conveyor belt.

(6) Replenishment node (purple): the placement point of replenished goods.

(7) Empty pallet recovery node (red): empty pallet recovery place.

The map used in Figure 1 is simplified to a dotted map as follows (Figure 2).

It is assumed that the AGV can only move within the limits of this plane scene. The region representation is shown in Table 1.

2.2. Condition Hypotheses. To simplify the calculations, the following reasonable assumptions are given.

(1) The AGV can turn accurately along a circular arc.

(2) The initial speed of the AGV is 5 units per second.

(3) The speed of the AGV will not be affected when it cuts from a straight line to an arc.

(4) AGVs do not stop accidently.

(5) Ignore the factors that affect the non-minimum turning radius and minimum safety distance of AGV travel.

2.3. Illustration of Symbols. The symbols are specified as follows (Table 2).

Idling: A continuous process in which an AGV stops moving but runs at its lowest possible speed.

Acceleration: A continuous process in which the acceleration of an AGV is greater than $0.1\,\mathrm{ms}^{-2}$.

Deceleration: A continuous process in which the acceleration of an AGV is less than $-0.1\,\mathrm{ms}^{-2}$.

Constant speed: A continuous process in which the absolute value of the acceleration of an AGV is less than $0.1\,\mathrm{ms}^{-2}$, a nonidling speed.

Average speed: The arithmetic mean of the speed of an AGV over a period.

Average driving speed: The arithmetic mean of the speed of an AGV when it is driven, excluding idling.

Idling time ratio: The percentage of total idling time in the total running time of an AGV.

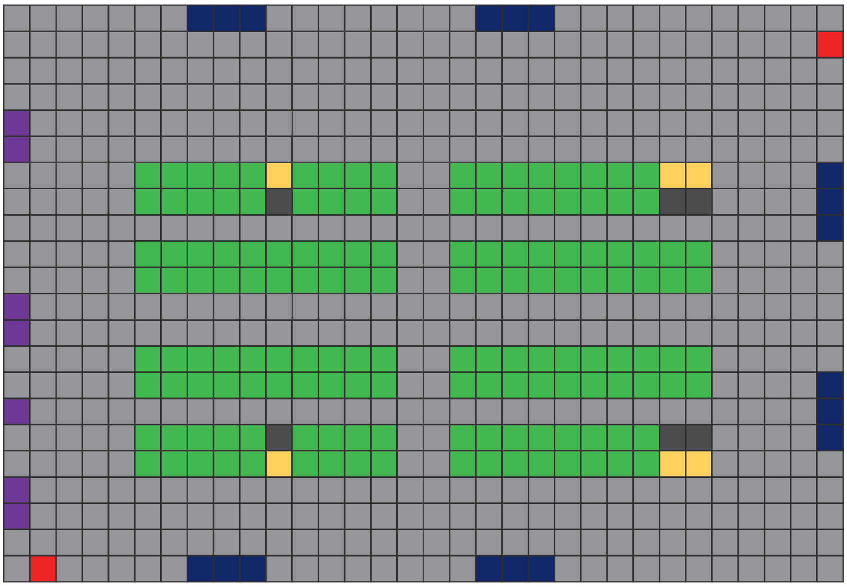

Figure 1: Unmanned warehouse map.

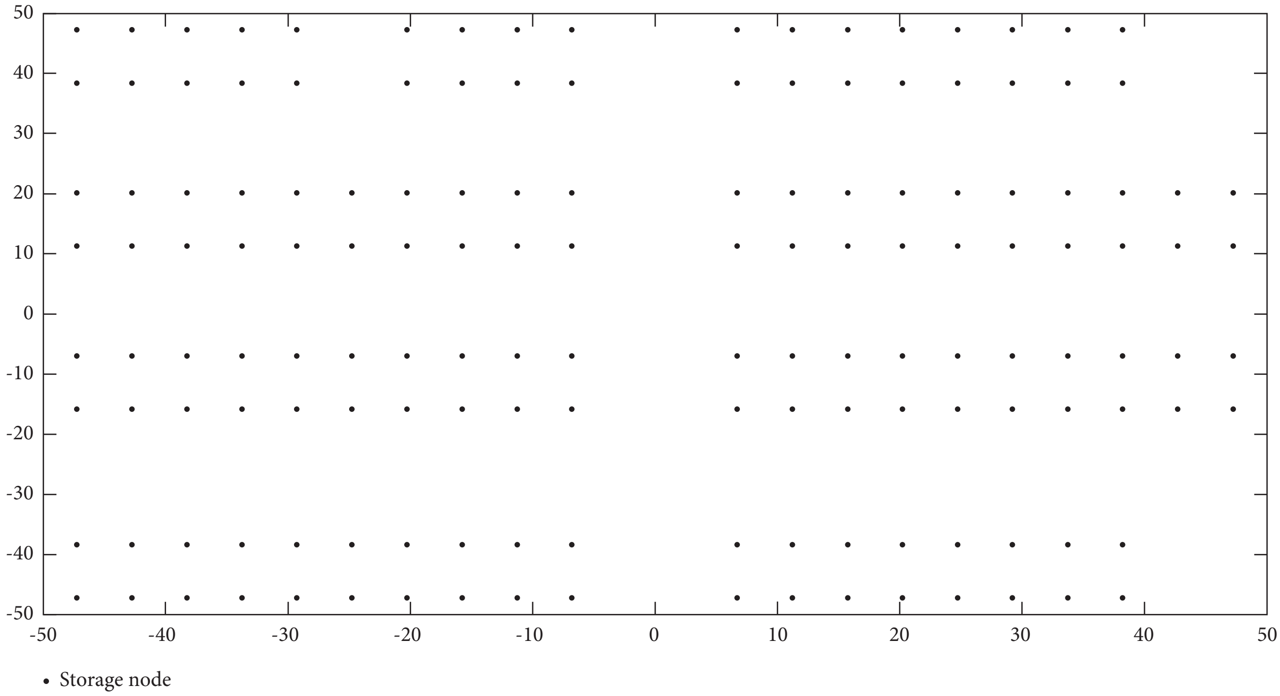

Figure 2: Simplified dot plot of unmanned warehouse distribution.

Average acceleration: The arithmetic mean of acceleration per unit time (second) of an accelerating AGV.

Average deceleration: The arithmetic mean of deceleration per unit time (second) of a decelerating AGV.

Acceleration time ratio: The percentage of accumulated time in acceleration in the total time of a period.

Deceleration time ratio: The percentage of accumulated time in deceleration in the total time of a period.

Speed standard deviation: The standard deviation of the speed of an AGV over a period, including idling.

Acceleration standard deviation: The standard deviation of the acceleration of an AGV that is accelerating over a period.

3. Modeling and Solution Finding

3.1. Minimization of Walking Path. Given the ignorance of possible collisions caused by AGVs during task implementation, we designed appropriate for an unmanned warehouse scenario. The possible shortest path from the starting point to the target

Table 1: Region representation.

Number	Region	Upper left vertex coordinate	Lower right vertex coordinate
1	Region 1	(5, 16)	(9, 15)
2	Region 2	(11, 16)	(14, 15)
3	Region 3	(17, 16)	(24, 15)
4	Region 4	(5, 13)	(14, 12)
5	Region 5	(17, 13)	(26, 12)
6	Region 6	(5, 9)	(14, 8)
7	Region 7	(17, 9)	(26, 8)
8	Region 8	(5, 6)	(9, 5)
9	Region 9	(11, 6)	(14, 5)
10	Region 10	(17, 6)	(24, 5)

Table 2: Meaning of symbols.

Symbol	Implication
d_m	Path length of m line
u_m	Path length of n arc
S	Minimum path length
T	Minimum time
v_0	Straight-line travel speed
v_p	Turning speed

point was determined, and the optimal path was found through the exhaustive and Q-learning methods [15]. Moreover, two AGV motion scenarios were considered, i.e., towards a single target point and multiple target points. In the former situation, the basic line-circle structure was first constructed as a solution for a single turn. Then, we developed a computational model for the different positional relationships between the straight-line and circular paths for the multiple-turn scenario. In the latter situation, straight-line turning was impossible for the AGV, which should turn in advance when passing the target point so that the intermediate target point was on the turning arc. Hence, we developed an optimization model to obtain the center of the turning arc at the intermediate target point, which was then considered an "obstacle" to transform the problem into a single-target point problem. The shortest path length was 2,812.52 units, and the traveling time was 585.6712 s. Thus, the objective function was the minimum value of the total handling AGVs' path when each AGV was as busy as possible.

$$X_{ij} = \begin{cases} 1, & \text{Robot } i \text{ assigns order } j, \\ 0, & \text{Others,} \end{cases} \tag{1}$$

where X_{ijk} represents k tasks of outgoing/returning/recycling assigned to the i^{th} AGV under the j^{th} order.

The scheduling algorithm in the unmanned warehouse scenario was designed for the case where potential collisions caused by the AGV handling during task implementation were ignored and aimed to determine the shortest path from the starting point to the target point utilizing the exhaustive method and the Q-learning method [16]. This paper considered two AGV scenarios, i.e., single- and multi-target points.

3.1.1. Single-Target Point Model (Involving Only the Starting and the Target Points). The turning trajectories of AGVs are arcs tangent to the straight-line paths. Consequently, the traveling routes can be viewed as a combination of multiple basic lines and circles.

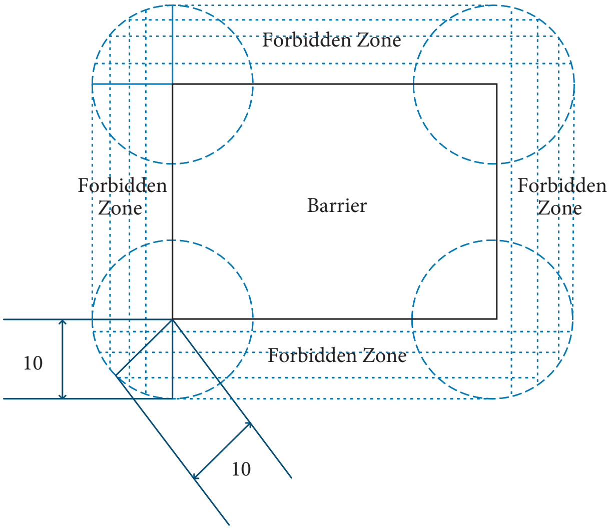

Figure 3: Schematic diagram of prohibited areas.

3.1.2. Division of Prohibited Areas. As shown in Figure 3, given that there is a minimum distance limit between AGVs and obstacles in travel, we first draw the forbidden zone of the enveloping obstacles. The forbidden zone is still a circle for circular obstacles, while the corners of the forbidden zone are circular for obstacles with vertices.

3.1.3. Shortest Path in the Case of AGV Turning at the Vertex of the Prohibited Area. As shown in Figure 4, from point A to point C, an AGV must make a turn. However, making the turn near the edge of the penalty area will shorten the total path.

Without considering the turning radius and other factors, let point D be the top of the penalty area and point B be any point outside the penalty area. From point A to point B, the shortest total path is realized when the AGV makes a turn at point D.

$$AB + BE > AD + DE, \tag{2}$$

$$EC + DE \ > DC. \tag{3}$$

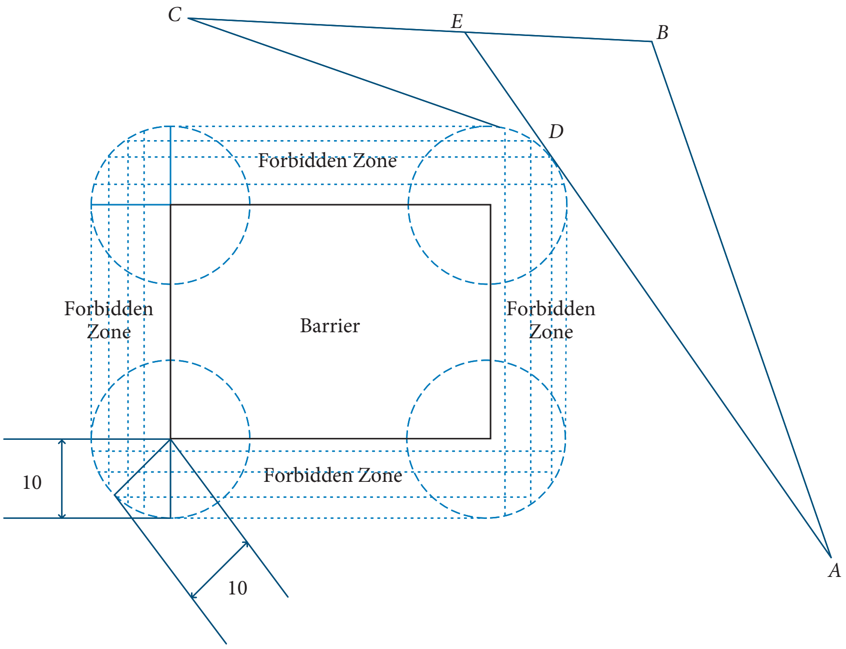

Figure 4: Change of direction at the top of the penalty area.

Adding the two equations yields:

$$AB + BE + EC + DE > AD + DE + DC. \quad (4)$$

Simplifying gives:

$$AB + BC > AD + DC. \quad (5)$$

In the process of A > B, the AGV chooses the shortest path when turning at D.

3.1.4. Shortest Path in the Case of the Minimum Turning Radius. For an AGV to move from point A to point B, it should bypass the prohibited area and turn near its vertex. The smaller the turn radius is, the shorter the path is. As depicted in Figure 5, the path from point A to point B is regarded as a stretchable rope and is assumed to naturally stretch (line segment AB) when the two points connect. The rope is stretched as the AGV needs to steer clear of the prohibited area. Additionally, the minimum turning radius of the AGV and the diameter of the prohibited area are 10. Thus, the rope can pass directly around the edge of the prohibited area.

$$E_P = \frac{1}{2}k\Delta L^2. \quad (6)$$

According to the principle of minimum potential energy, the systems' potential energy reaches its minimum value when the elastic body is in equilibrium. Here, the circle was considered elastic in the initial state illustrated in the above figure. Under the forces illustrated in the figure, the system gradually reaches equilibrium as the circle tends to shrink, ultimately obtaining the minimum potential energy of the elastic rope, which decreases as the circle radius reduces, i.e., the shortest path goes down. This finding proves that the path is the shortest in the case of the minimum turning radius.

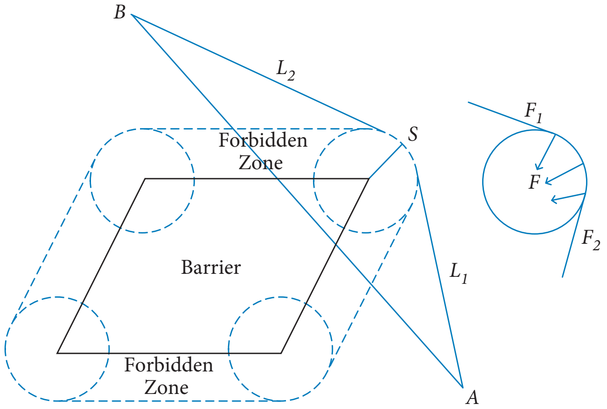

Figure 5: AGV obstacle turning radius.

3.1.5. Shortest Path in the Case of the Center of the Turning Arc Located at the Vertex. When the AGV turns near the vertex of the prohibited area, the path is the shortest if the center of the turning arc is at the vertex. As illustrated in Figure 6, both circles O and O' with a radius of R and R', respectively, bypass the farthest point D. Notably, the center of circle O falls on the vertical line. L_1, S, L_2, and L_1', S', L_2' are the lengths of the tangent segments from points A and B to the circles O and O' and the arcs contained, respectively.

$$L_1 + S + L_2 < L_1' + S' + L_2'. \quad (7)$$

Point D and circle center O fall on the vertical line (passing D) of the line segment AB, while the circle center O' is located beyond this vertical line. Therefore, the two circles

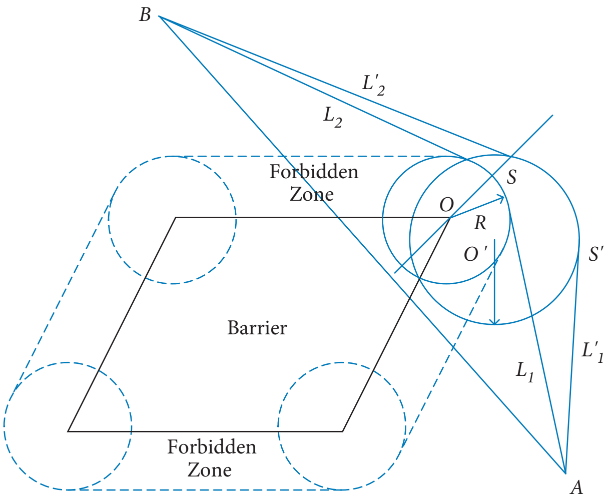

Figure 6: Obstacle avoidance turning circle center position.

are only intersected rather than tangent. To pass point D, the maximum distance from the point on circle O' to the straight-line AB must be greater than that from the point on circle O to the straight-line AB. Based on the conclusions proved above, the result below was generated:

$$L_1 + S + L_2 < L_1' + S' + L_2'. \tag{8}$$

When the AGV bypasses an obstacle with a vertex, the shortest path is the arc turning with the vertex as the center and the minimum turning radius so that the minimum turning radius is the same as the minimum safe distance between the AGV and the obstacle. In other words, the path is the shortest when the AGV makes a turn along the edge of the prohibited area.

3.1.6. Modeling. Based on the above conclusions, the shortest path is always constituted by several tangents and arcs, regardless of the number of obstacles between the starting and target points. As proved previously, the path is the shortest when the AGV passes through all obstacles by turning along the edge of the prohibited area, where the radius of the turning arc is that of the hazardous area. Therefore, in the model below, when passing through obstacles, the AGV turns at the obstacle's vertex with the minimum turning radius r placed at the obstacle's center.

Given that the AGV aims to move from starting point A (x_1, y_1) to target point B (x_2, y_2), while turning on the arc centered at vertex D (x_3, y_3) with radius r, and C and E are the tangency points, the point C and E coordinates and the length of $A\overset{\frown}{CE}B$ should be calculated:

$$AB = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \tag{9}$$

$$AD = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2}, \tag{10}$$

$$BD = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}, \tag{11}$$

$$DE + \mathrm{BEDC} + \mathrm{AC} \tag{12}$$

$$\begin{aligned} BE &= \sqrt{BD^2 - r} \\ &= \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 - r}, \end{aligned} \tag{13}$$

$$\begin{aligned} AC &= \sqrt{AD^2 - r} \\ &= \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2 - r}. \end{aligned} \tag{14}$$

Setting the coordinates of points C and E as (x_i, y_i) and (x_j, y_j), respectively, provides the following formulae:

$$\begin{cases} BE = \sqrt{\left(x_2 - x_j\right)^2 + \left(y_2 - y_j\right)^2}, \\ DE = \sqrt{\left({x_3}^2 - x_j\right)^2 + \left(y_3 - y_j\right)^2}, \end{cases} \tag{15}$$

$$\begin{cases} AC = \sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2}, \\ DC = \sqrt{\left(x_3^2 - x_i\right)^2 + (y_3 - y_i)^2}. \end{cases} \tag{16}$$

Moreover, the coordinates of point E are expressed as:

$$\begin{cases} \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 - r} = \sqrt{\left(x_2 - x_j\right)^2 + \left(y_2 - y_j\right)^2}, \\ \sqrt{\left(x_3^2 - x_j\right)^2 + \left(y_3 - y_j\right)^2} = r. \end{cases} \tag{17}$$

and the coordinates of point C:

$$\begin{cases} \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2 - r} = \sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2}, \\ \sqrt{\left(x_3^2 - x_i\right)^2 + (y_3 - y_i)^2} = r. \end{cases} \tag{18}$$

(17) and (18) can be connected to the coordinates of C and E.

$$\angle ADC = \mathrm{arc\,COS}\frac{AD^2 + BD^2 - AB^2}{2AD \times BD}, \tag{19}$$

$$\angle ADC = \mathrm{arc\,COS}\frac{r}{AD}, \tag{20}$$

$$\angle BDE = \mathrm{arc\,COS}\frac{r}{BD}, \tag{21}$$

$$\angle CDE = 2\pi - \angle ADC - \angle BDE - \angle ADB\ . \tag{22}$$

The length of arc is

$$\overset{\frown}{C}\overset{\frown}{E} = r \times \angle CDE\ . \tag{23}$$

The path length for the AGV turning once on the way was calculated as:

$$S = AC + BE + \overset{\frown}{C}\overset{\frown}{E}. \tag{24}$$

Furthermore, the traveling time was:

$$T = \frac{AC}{v_0} + \frac{BE}{v_0} + \frac{\widehat{CE}}{v_\rho}. \tag{25}$$

3.1.7. Model Calculation. It was assumed that on the path from the starting point to the target point, there were m straight lines with a length of d_m and n arcs with a length of u_n. Thus, the total distance for the AGV moving from the starting point to the target point was expressed as:

$$s = \sum_{m=1}^{m} d_m + \sum_{n=1}^{n} u_n. \tag{26}$$

Moreover, the traveling time was:

$$T = \frac{\sum_{m=1}^{m} d_m}{v_0} + \frac{\sum_{n=1}^{n} u_n}{v_\rho}, \tag{27}$$

where $v_\rho = (v_0/1 + e^{10-0.1\rho^2})$ and ρ is the turning radius.

Figure 7 illustrates the possible optimal paths, as proved in the above sections.

The two paths having a basic line-circle structure could be directly solved by model 1, obtaining 471.032 and 505.9835 units, respectively, through MATLAB. Therefore, the optimal total distance of the AGV passing from the upper left of obstacle 5 was $S = 471.0372$ units, the total traveling time was $T = 96.0178$ s, containing two straight-line segments and one arc line segment (at center (80, 210) and radius $r = 10$). The specific conditions of the path are listed in Table 3:

Through the calculations performed by MATLAB, the optimal total distance of the AGV was $S = 3{,}812.52$ units. The traveling time was $T = 585.6712$ s, involving 16 straight-line segments and 15 arcs. Further details are reported in Tables 4 and 5.

3.2. Task Equalization. This section presents the hardware conditions of the experiment in order to validate the performance of the proposed methodology. The algorithms was coded in MATLAB 2021 software using a computer with the following specifications: Intel (R) Core (TM) i9-10885H CPU @ 2.40 GHz.

To better balance the sorting stations' load and prevent local AGV handling congestion, we proposed a SVM-based AGV scheduling strategy. According to the experimental results of dividing the equilibrium task area, the proposed SVM method outperformed the single-attribute AGV scheduling rules. In the suggested scheme, first, we set a sorting station for each pallet, thus realizing the pairing optimization. On this basis, the model of problem 1 was rebuilt to:

$$X_{ij} = \begin{cases} 1, & \text{Pallet } i \text{ assigns picking station } j, \\ 0, & \text{Others,} \end{cases} \tag{28}$$

where X_{ij} represents the j^{th} picking station specified for the i^{th} pallet.

Generating the training sample involves the following steps. As a supervised learning technique, SVM can output a maximum-boundary hyperplane to perfectly separate two types of training samples [17]. The training samples in the supervised learning comprise features and labels, which refer to the current system state and optimal scheduling rules in the case of AGV scheduling, respectively. This rule can be dynamically chosen based on the current system state.

The training sample set is not linearly separable, i.e., it cannot be well separated by a linear hyperplane. In this case, the kernel function (Φ: $f \longrightarrow H$) will map the feature vector (f) into a higher-dimension Hilbert space (H). Here, the Hilbert space was reproduced by a Gaussian kernel (or radial basis function, RBF), where the RBF parameters were defined by candidate scheduling rules selected *via* an SVM scheduler [18]. Thus, the scheduling rules were taken as the SVM labels. The AGV scheduling rules considering the order due date were also the candidate scheduling rules since the average work delay was taken as the performance indicator.

Partitioning rules for the candidate scheduling area:

(1) Shortest travel time (STT): select the task closest to the AGV.

(2) First come, first served (FCFS): select more urgent tasks.

(3) Minimum remaining output queue space (MROQS): select tasks in the output queue with the remaining space.

(4) Earliest due date (EDD): select the task with the closest due date.

(5) Critical ratio (CR): select tasks with the lowest critical ratio [CR = (due date-this date)/(remaining operation time)].

(6) Dynamic slack (DS): select tasks with the shortest slack time (remaining slack time = remaining operation time).

(7) Nearest vehicle (NV): select the AGV closest to the task location.

(8) Longest idle AGV (LIV): select the AGV with the longest idle time.

(9) Lowest utilization AGV (LU): select the AGV with the lowest utilization ratio.

The division distribution of the ten regions was determined by the division rules as shown in Figure 8.

In machine learning, every system attribute potentially influencing the system's performance should be considered an environmental state. Given the environment of this research, we chose the system attributes relative to the AGV, which empirically are nine, utilized as the training samples. Moreover, the average delay rate was recorded at the end of scheduling and was employed as a performance indicator. After performing N simulations with a fixed random seed, the performance of N scheduling rules was recorded, and the scheduling rule with the highest performance was used as the label of the training sample.

The error diagrams of the SVM scheduling after training in each division are provided in Table 6:

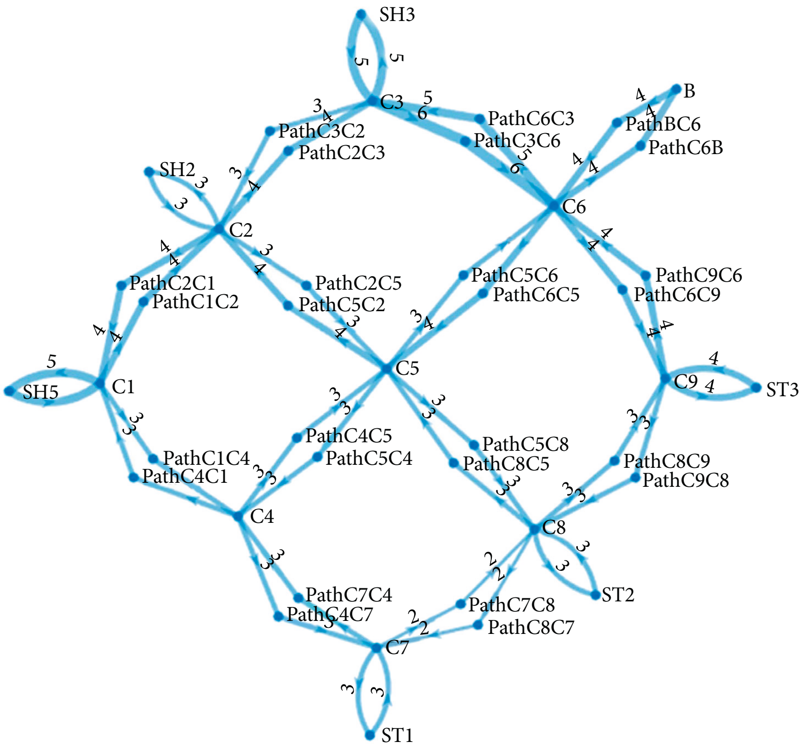

Figure 7: Diagram of possible optimal paths.

Table 3: Shortest path situation.

	Start point coordinates	Endpoint coordinates	Turning arc center	Distance	Time
Straight 1	(0, 0)	(70.50595, 213.1405)	—	224.4994	44.9
Arc 1	(70.50595, 213.1405)	(76.60645, 219.4066)	(80,210)	9.051	3.7402
Straight 2	(76.60645, 219.4066)	(300,300)	—	237.4868	47.4974
Sum	—	—	—	471.0372	96.1376

Table 4: Partial shortest paths for multiple target points.

	Start point coordinates	Endpoint coordinates	Turning arc center	Distance	Time
Straight 1	(0,0)	(70.5059, 213.1405)	—	224.4994	44.8999
Arc 1	(70.5059, 213.1405)	(76.6064, 219.4066)	(80,210)	9.25	3.7
Straight 2	(76.6064, 219.4066)	(294.1547, 294.6636)	—	229.98	45.996
Arc 2	(294.1547, 294.6636)	(281.3443, 301.7553)	(290.8855 304.1141)	15.3589	6.1436
. . .	. . .	. . .	. . .	. . .	. . .
Straight14	(727.9377, 513.9178)	(492.0623, 206.0822)	-	387.81	77.562
Arc length14	(492.0623, 206.0822)	(491.6552, 205.5103)	(500,200)	0.6981	0.2792
Straight15	(491.6552, 205.5103)	(412.1387, 90.2314)	-	133.04	26.608
Arc15	(412.1387, 90.2314)	(418.3348, 94.4085)	(410,100)	7.6794	3.0718
Straight16	(418.3348, 94.4085)	(0,0)	—	421.84	84.368
Sum				2812.52	585.6712

The error of the trained SVM scheduling division results is shown in Figure 9. The single-attribute scheduling rule shows a slow convergence and weak stability. Obviously, the SVM scheduler significantly outperformed most of the single-attribute scheduling rules.

Different combinations of single-attribute scheduling rules and SVM scheduling methods were tested and simulated. For each scheduling method, 30 tests were conducted using a typical random number (CRN) technique. Specifically, the comparison value was expressed as the average delay value of a single scheduling rule divided by that of the SVM scheduler, thereby comparing the performance of the single scheduling rule and the SVM scheduler. A statistical analysis of the results was carried out to determine whether the SVM scheduling rule was superior to the single-attribute scheduling rule [19]. Obviously, the SVM scheduler significantly outperformed most of the single-attribute scheduling rules.

Table 5: Shorstet path through mulitiple target points on-the-way to the specific situation.

	Start point coordinates	Endpoint coordinates	Turning arc center	Distance	Time
Straight 1	(0, 0)	(70.5059, 213.1405)	—	224.4994	44.8999
Arc 1	(70.5059, 213.1405)	(76.6064, 219.4066)	(80, 210)	9.25	3.7
Straight 2	(76.6064, 219.4066)	(294.1547, 294.6636)	—	229.98	45.996
Arc 2	(294.1547, 294.6636)	(281.3443, 301.7553)	(290.8855, 304.1141)	15.3589	6.1436
Straight 3	(281.3443, 301.7553)	(229.8206, 531.8855)	—	236.83	47.366
Arc length 3	(229.8206, 531.8855)	(225.4967, 537.6459)	(220, 530)	6.8068	2.7228
Straight 4	(225.4967, 537.6459)	(144.5033, 591.6462)	—	96.95	19.39
Arc length 4	(144.5033, 591.6462)	(140.8565, 595.9507)	(150, 600)	5.7596	2.3038
Straight 5	(140.8565, 595.9507)	(99.08612, 690.2698)	—	103.16	20.632
Arc 5	(99.08612, 690.2698)	(109.113, 704.2802)	(108.296, 694.3191)	20.7694	8.3078
Straight 6	(109.113 704.2802)	(270.8817, 689.9611)	—	162.4	32.48
Arc length 6	(270.8817, 689.9611)	(272, 689.7980)	(270, 680)	1.0472	0.4188
Straight 7	(272,689.7980)	(368, 670.282)	—	97.98	19.596
Arc 7	(368, 670.282)	(370, 670)	(370, 680)	2.0944	0.8378
Straight 8	(370, 670)	(430, 670)	—	60	12
Arc length 8	(430, 670)	(435.5878, 671.7068)	(420, 680)	5.9341	2.3736
Straight 9	(435.5878, 671.7068)	(534.4115 738.2932)	—	119.16	23.832
Arc 9	(534.4115, 738.2932)	(540, 740)	(540, 730)	5.9341	2.3736
Straight 10	(540, 740)	(670, 740)	—	130	26
Arc length 10	(670, 740)	(679.9126, 731.3196)	(670, 730)	14.3846	5.7538
Straight 11	(679.9126, 731.3196)	(690.9183, 648.6458)	—	83.403	16.6806
Arc 11	(690.9183, 648.6458)	(693.5095, 643.1538)	(709.7933, 642.0227)	6.17	2.468
Straight 12	(693.5095, 643.1538)	(727.3214, 606.8116)	—	129.6305	25.9261
Arc length 12	(727.3214, 606.8116)	(730, 600)	(720, 600)	7.4928	2.997
Straight 13	(730, 600)	(730, 520)	—	80	16
Arc 13	(730, 520)	(727.9377, 513.9178)	(720, 520)	6.4577	2.583
Straight 14	(727.9377, 513.9178)	(492.0623, 206.0822)	—	387.81	77.562
Arc length 14	(492.0623, 206.0822)	(491.6552, 205.5103)	(500, 200)	0.6981	0.2792
Straight 15	(491.6552, 205.5103)	(412.1387, 90.2314)	—	133.04	26.608
Arc 15	(412.1387, 90.2314)	(418.3348, 94.4085)	(410, 100)	7.6794	3.0718
Straight 16	(418.3348, 94.4085)	(0, 0)	—	421.84	84.368
Sum		—		2812.52	585.6712

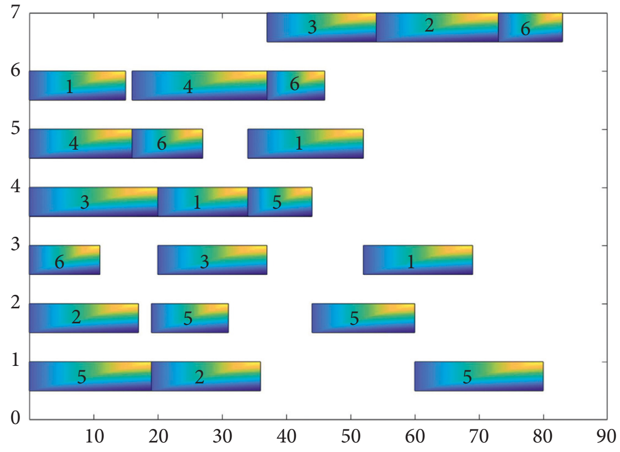

Figure 8: The division of the distribution map.

4. Results and Evaluation

4.1. Simulation and Validation. According to the relationship between the turning radius and speed, path optimization was realized based on the model for calculating the shortest path. Then, a nonlinear programming model for the shortest time was constructed. Utilizing NSGA-II provided the shortest time of 94.22825 s, the turning radius of 12.9886 units, and the circle center coordinates of the turning were 82.1414 and 207.1387 [20]. The obstacles were classified into two categories: having vertices and having no vertices. Next, the path proved to be optimal when the AGV made a turn with the minimum radius and the circle center was located at the obstacle's

Table 6: Scheduling rule performance.

Scheduling area division mode	NV	LIV	LU	SVM
STT	275.02	278.05	279.27	—
MROQS	655.87	727.90	657.08	—
FCFS	1237.31	1235.06	1243.26	—
EDD	746.85	734.53	735.35	—
DS	671.71	671.90	674.12	—
CR	600.33	602.18	602.29	—
SVM	—	—	—	253.99

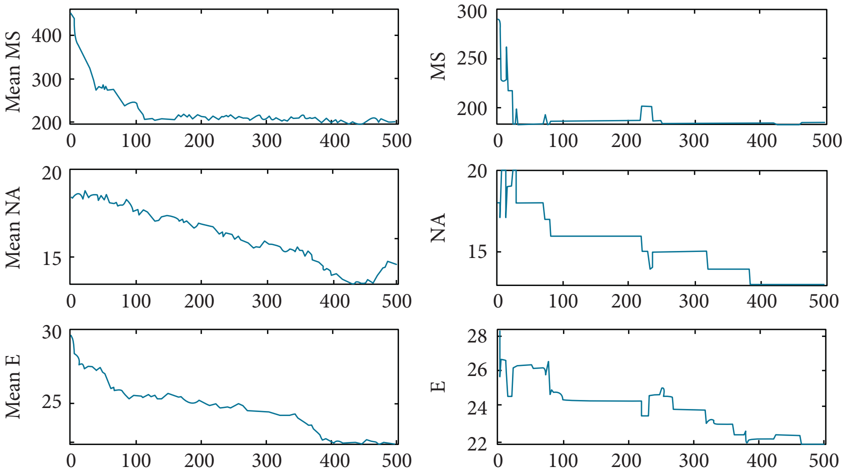

Figure 9: Error diagram.

vertex (the circular obstacle was located at the circle center), thus reasonably solving the problems of obstacle congestion and deadlock.

The problem here was the shortest time spent by the AGV traveling from point O to point A and bypassing obstacle 5. According to the shortest path calculated, the time was also the shortest when the AGV passed from the upper left of obstacle 5. The total time spent by the AGV comprised the time on the straight line and arc line segments. The paths with the shortest time and the shortest distance were different, so the path with the shortest time should be calculated first. Since the closest distance between the AGV and the obstacle must not be less than 10 units, the traveling range of the AGV was determined by the obstacles to be avoided on its path and the range influencing its action.

The maximum turning speed of the AGV was expressed as:

$$\begin{aligned} v &= v(\rho) \\ &= \frac{v_0}{1 + e^{10-0.1\rho^2}}. \end{aligned} \tag{29}$$

The time spent by the AGV in passing the arc was:

$$t = \frac{\alpha\rho}{v_\rho}. \tag{30}$$

Based on the above formulae, decreasing the turning radius slows the AGV, and the arc it followed became shorter. When the turning radius of the AGV increased, both the AGV's turning speed and the arc's length increased. Thus, we did not identify any direct linear relationship between the turning time t and the turning radius ρ. Hence, the turning radius ρ change was limited. In practice, an excessively large ρ may cause collisions, while the AGV may roll over for a very small ρ (less than 10).

Then, a nonlinear programming model for the minimum time was built based on the variation range of ρ, thereby obtaining ρ at the minimum t. Here, $O(x_1, y_1)$ was set as the starting point, $A(x_2, y_2)$ was the target point, and $P(x_3, y_3)$ was the upper left vertex of obstacle 5. It was assumed that the AGV made a turn at point $C(x_c, y_c)$, with $N(x, y)$ as the center and r as the radius, passed arc $\widehat{BC}$, and then turned at point $B(x_b, y_b)$, to ultimately obtain the path with the shortest time spent. Points B and C were connected to generate line segment BC, perpendicular to segment $N\ D$. The length of ON, AN, tangent OC, and tangent AB is denoted as a, b, s_1, and s_2, respectively.

Let $BD = d$. Since both point B and point C are tangent points, the following results were obtained:

$$\angle DNB = \frac{1}{2}\theta, \tag{31}$$

$$BD = \frac{1}{2}BC. \tag{32}$$

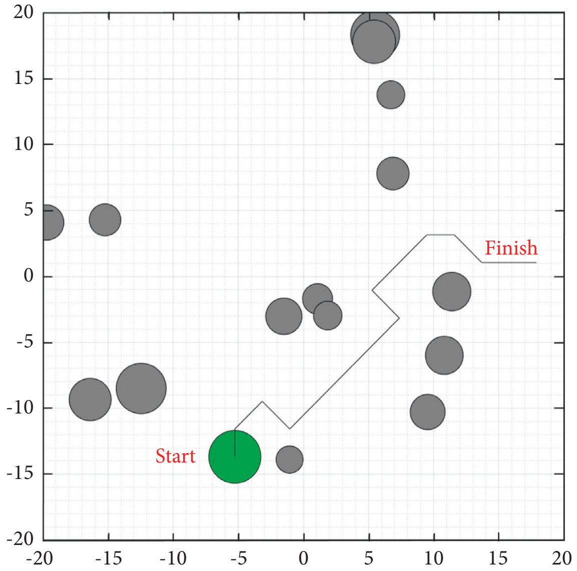

Figure 10: Diagram of AGV obstacle avoidance simulation (a).

The radius of the circle in which B and C are located is

$$a = \sqrt{(x - x_1)^2 + (y - y_1)^2}, \quad (33)$$

$$b = \sqrt{(x - x_2)^2 + (y - y_2)^2}, \quad (34)$$

$$\sqrt{(x_c - x)^2 + (y_c - y)^2} = r, \quad (35)$$

$$\sqrt{(x_b - x)^2 + (y_b - y)^2} = r, \quad (36)$$

$$s_1 = \sqrt{a^2 - r^2}, \quad (37)$$

$$s_2 = \sqrt{b^2 - r^2}, \quad (38)$$

$$BC = \sqrt{(x_b - x_c)^2 + (y_b + y_c)^2}, \quad (39)$$

$$\sin\left(\frac{\theta}{2}\right) = \frac{\sqrt{(x_b - x_c)^2 + (y_b + y_c)^2}}{2 \times r}, \quad (40)$$

$$l = 2r \times \arcsin\left(\frac{\sqrt{(x_b - x_c)^2 + (y_b - y_c)^2}}{2r}\right). \quad (41)$$

The objective of the shortest time was formulated based on the straight-line distance, speed of AGV, the arc length, and the AGV's speed while passing the arc:

$$\text{Min} = \frac{s_1 + s_2}{v_o} + \frac{l}{v_p}. \quad (42)$$

Given that the distance between the AGV and the obstacle must be more than 10 units, the radius r was also constrained, i.e., the arc-obstacle distance must be over 10 units:

$$r - \sqrt{(x - x_3)^2 + (y - y_3)^2} \geq 10. \quad (43)$$

The range of the two tangent points is

$$\begin{aligned} x_c &< 80, \\ y_b &> 210. \end{aligned} \quad (44)$$

The constraints that the circle center had the shortest distance from the obstacle were expressed as follows:

$$80 \leq x \leq 230, \quad (45)$$

$$0 < y \leq 210. \quad (46)$$

The AGV obstacle avoidance simulated through MATLAB programming is illustrated in Figures 10 and 11:

The turning path of an AGV must include an arc tangent to the straight-line path. Points C and B are on a circle with a radius r.

$$\sqrt{(x_c - x)^2 + (y_c - y)^2} = r, \quad (47)$$

$$\sqrt{(x_b - x)^2 + (y_b - y)^2} = r, \quad (48)$$

$$s_1 = \sqrt{a^2 - r^2}, \quad (49)$$

$$s_2 = \sqrt{b^2 - r^2}. \quad (50)$$

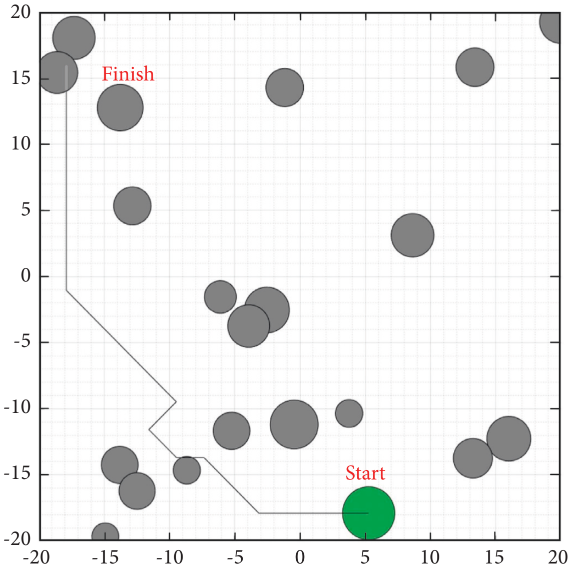

Figure 11: Diagram of AGV obstacle avoidance simulation (b).

Table 7: Shortest path.

	Start point coordinates	Endpoint coordinates	Turning arc center	Distance	Time
Straight 1	(0,0)	(69.8045, 211.9779)	—	223.1755	44.6351
Arc 1	(69.8045, 211.9779)	(77.74918, 220.1387)	(82.1414, 207.9153)	11.7899	2.360433
Straight 2	(77.74918, 220.1387)	(0,0)	—	236.1636	47.23272
Sum	—	471.129	94.22825		

Based on the conditions above, the optimization model for the shortest time was built:

$$\text{Min} = \frac{s_1 + s_2}{v_o} + \frac{l}{v_p}, \tag{51}$$

$$\begin{matrix} s \\ t \end{matrix} \begin{cases} r - \sqrt{(x - x_3)^2 + (y - y_3)^2} \geq 10, \\ r = \sqrt{(x_b - x)^2 + (y_b - y)^2}, \\ r = \sqrt{(x_c - x)^2 + (y_c - y)^2}, \\ s_2 = \sqrt{(x - x_2)^2 + (y - y_2)^2 - r^2}, \\ s_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2 - r^2}, \\ 80 < x < 230, \\ \qquad y_b > \end{cases} 210, x_c < 80, 0 < y < 210. \tag{52}$$

Utilizing Lingo and NSGA-II software, the shortest time of 94.22825 s and the turning radius of $r = 12.9886$ were obtained. The results are reported in Table 7 and the resulting diagram Figure 12:

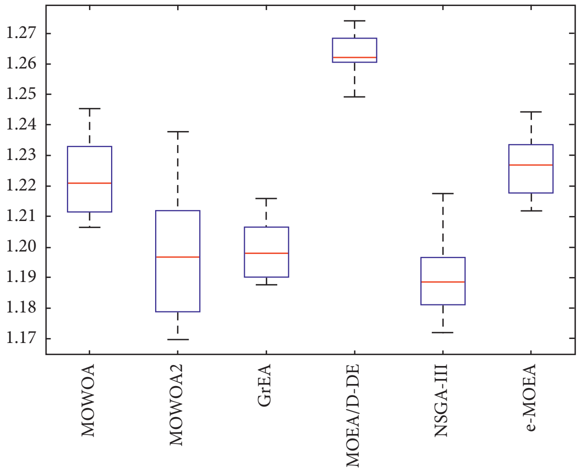

Figure 12: Comparison diagram of AGV obstacle avoidance results obtained by NSGA-II.

4.2. Model Promotion. In this paper, the problem of deadlock was simplified. Nevertheless, further model optimization is required when the actual deadlock situation is more complicated. This work assumed that the AGV moved from the starting point R to the target point M_0, where the path comprised linear segments and arcs, set as m and n, respectively. Thus, the objective function was expressed as:

$$\text{Min} = \sum_{i=1}^{m} d_i + \sum_{j=1}^{n} l_j, \qquad (53)$$
$$\begin{cases} r \geq 1, \\ k \geq 1. \end{cases}$$

Computer software such as MATLAB or Lingo can be used to solve the optimal path between the start point and the target point.

5. Conclusion

According to the features of existing AGV cluster systems, an SVM-based AGV scheduling strategy was developed by determining the shortest possible path between the starting point and the target point. Then, the optimal path was determined by the exhaustive method and the Q-learning method and was optimized by several schemes, through which the optimal path in the relative optimization was obtained. A nonlinear programming model for the shortest time was built based on the relationship between the turning radius and speed based on the shortest path. The calculation with NSGA-II proved that the AGV path was optimal when the AGV turned with the minimum radius, and the circle center was located at the obstacle's vertex (the circular obstacle was located at the circle center). Several simulation tests and calculation results validated the proposed method, which rationally solved the problem of obstacle congestion and deadlock. After optimization, the constructed model was solved with high accuracy via analytical geometry. However, the processing burden increased, resulting in low utilization efficiency. Hence, the proposed method is inefficient for complicated deadlock situations.

Future research will consider the minimum spanning tree, and the shortest path of the AGV combined the dynamic monitoring and real-time data of AGV cluster systems.

Data Availability

The data used to support the findings of the study can be obtained from the author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The author would like to express his gratitude to EditSprings (https://www.editsprings.cn) for the expert linguistic services provided and Zhi Dong Li for experimental assistance. This research was supported by the 2021 Innovation Training Program for University Students (nos. 202112757001, 202112757005, 202112757006, and 202112757016) and the Chongqing Police College Studies Research Project Program.

References

[1] S. Vinoth, H. L. Vemula, B. Haralayya, P. Mamgain, M. F. Hasan, and M. Naved, "Application of cloud computing in banking and e-commerce and related security threats," *Materials Today Proceedings*, vol. 51, pp. 2172–2175, 2022.

[2] P. Giridharan, R. G. Chittawadigi, and G. Udupa, "Intuitive manipulation of delta robot using leap motion," in *Mechanism and Robotics: In Machines*Springer, Singapore, 2022.

[3] J. Chen, X. Zhang, X. Peng, D. Xu, and J. Peng, "Efficient routing for multi-AGV based on optimized Ant-agent," *Computers & Industrial Engineering*, vol. 167, Article ID 108042, 2022.

[4] Y. Zhou and N. Huang, "Airport AGV path optimization model based on ant colony algorithm to optimize Dijkstra algorithm in urban systems," *Sustainable Computing: Informatics and Systems*, vol. 35, Article ID 100716, 2022.

[5] A. Meysami, J. C. Cuillière, V. François, and S. Kelouwani, "Investigating the impact of triangle and quadrangle mesh representations on AGV path planning for various indoor environments: with or without inflation," *Robotics*, vol. 11, no. 2, 2022.

[6] C. Liang, X. Hu, L. Shi, H. Fu, and D. Xu, "Joint Dispatch of Shipment Equipment Considering Underground Container Logistics," *Computers & Industrial Engineering*, vol. 165, Article ID 107874, 2022.

[7] A. Azadeh, M. Ravanbakhsh, M. Rezaei-Malek, and A. Taheri-Moghaddam, "Unique NSGA-II and MOPSO algorithms for improved dynamic cellular manufacturing systems considering human factors," *Applied Mathematical Modelling*, vol. 48, pp. 655–672, 2017.

[8] X. Chu, D. Gao, S. Cheng, L. Wu, J. Chen, and Q. Qin, "Worker assignment with learning-forgetting effect in cellular manufacturing system using adaptive memetic differential search algorithm," *Computers & Industrial Engineering*, vol. 136, pp. 381–396, 2019.

[9] R. Zaghdoud, K. Mesghouni, S. C. Dutilleul, and K. Ghedira, "A hybrid method for assigning containers to AGVs in the dynamic environment of container terminals," *Studies in Informatics and Control*, vol. 24, no. 1, pp. 43–50, 2015.

[10] W. Malopolski, "A sustainable and conflict-free operation of AGVs in a square topology," *Computers & Industrial Engineering*, vol. 126, pp. 472–481, 2018.

[11] S. Kim, H. Jin, M. Seo, and D. Har, "Optimal path planning of automated guided vehicle using dijkstra algorithm under dynamic conditions," in *Proceedings of the 2019 7th International Conference on Robot Intelligence Technology and Applications*, pp. 231–236, Daejeon, Korea, November 2019.

[12] C. Wang, L. Wang, J. Qin et al., "Path planning of automated guided vehicles based on improved A-Star algorithm," in *Proceedings of the 2015 IEEE International Conference on Information and Automation*, pp. 2071–2076, Lijiang, China, August 2015.

[13] R. Tai, J. Wang, and W. Chen, "A prioritized planning algorithm of trajectory coordination based on time windows for

multiple AGVs with delay disturbance," *Assembly Automation*, vol. 39, no. 5, pp. 753–768, 2019.

[14] N. Singh, Q. V. Dang, A. Akcay, I. Adan, and T. Martagan, "A matheuristic for AGV scheduling with battery constraints," *European Journal of Operational Research*, vol. 298, no. 3, pp. 855–873, 2022.

[15] Y. Li, W. Gu, M. Yuan, and Y. Tang, "Real-time data-driven dynamic scheduling for flexible job shop with insufficient transportation resources using hybrid deep Q network," *Robotics and Computer-Integrated Manufacturing*, vol. 74, Article ID 102283, 2022.

[16] Q. Zhu, S. Su, T. Tang, W. Liu, Z. Zhang, and Q. Tian, "An eco-driving algorithm for trains through distributing energy: a Q-Learning approach," *ISA Transactions*, vol. 122, pp. 24–37, 2022.

[17] J. S. Nisha, V. P. Gopi, and P. Palanisamy, "Classification of informative frames in colonoscopy video based on image enhancement and phog feature extraction," *Biomedical Engineering: Applications, Basis and Communications*, vol. 34, no. 2, Article ID 2250015, 2022.

[18] R. Hu, J. Gan, X. Zhu, T. Liu, and X. Shi, "Multi-task multi-modality SVM for early COVID-19 Diagnosis using chest CT data," *Information Processing & Management*, vol. 59, no. 1, Article ID 102782, 2022.

[19] M. A. Djeziri, O. Djedidi, N. Morati, J. L. Seguin, M. Bendahan, and T. Contaret, "A temporal-based SVM approach for the detection and identification of pollutant gases in a gas mixture," *Applied Intelligence*, vol. 52, no. 6, pp. 6065–6078, 2022.

[20] G. K. Suman, J. M. Guerrero, and O. P. Roy, "Stability of microgrid cluster with Diverse Energy Sources: a multi-objective solution using NSGA-II based controller," *Sustainable Energy Technologies and Assessments*, vol. 50, Article ID 101834, 2022.