

Received October 10, 2020, accepted November 15, 2020, date of publication November 20, 2020, date of current version December 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3039515

A Collision and Deadlock Prevention Method With Traffic Sequence Optimization Strategy for UGN-Based AGVS

HAINING XIAO¹, XING WU², (Member, IEEE), DEJIN QIN¹, AND JINGJING ZHAI²

¹College of Mechanical Engineering, Yancheng Institute of Technology, Yancheng 224051, China

²College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Corresponding author: Xing Wu (wustar5353@nuaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 52005427, Grant 61973154, and Grant 61503319; in part by the National Defense Basic Scientific Research Program of China under Grant JCKY2018605C004; in part by the Natural Science Research Project of Jiangsu Higher Education Institutions under Grant 19KJB510013 and Grant 18KJA460009; and in part by the Foundation of Graduate Innovation Center in NUAU under Grant KFJJ20190516.

ABSTRACT Due to the advantages of high efficiency and flexibility, the Automatic Guided Vehicle System (AGVS) has been more and more widely used in many industries. However, one of the main challenges in AGVS control is how to prevent collision and deadlock between vehicles. Although many collision and deadlock prevention algorithms have been proposed, they are inefficient for the AGVS based on a unidirectional guide-path network (UGN). In order to solve this problem, this paper proposes a collision and deadlock prevention method with a traffic sequence optimization strategy for the UGN-based AGVS. First, a vehicle coordination method based on the semaphore theory and Internet of Things (IoT) positioning technology is proposed to prevent collisions and intersection congestion deadlocks. Then, to avoid cycle deadlocks, a cycle deadlock search and avoid algorithm based on the digraph theory is developed. After that a bidding mechanism-based strategy is developed to optimize the vehicle traffic sequence in each path intersection. Finally, extensive simulation tests are conducted to verify the performance of the proposed methods and strategy. Simulation results show that, compared to the zone controlled (ZC) methods, the average travel time of the proposed methods is reduced by 6.2%–29.6%, and the average throughput is increased by 3.5%–27.4%. Also, the bidding mechanism-based traffic sequence optimization strategy can not only increase the average throughput of the processing subsystem but also reduce the congestion and deadlock risk of the logistics transportation subsystem. The proposed method is suitable for an UGN-based AGVS in the manufacturing application environment.

INDEX TERMS Automatic guided vehicle system, collision and deadlock prevention, traffic sequence optimization, unidirectional guide-path network.

I. INTRODUCTION

As a flexible material handling system, the Automated Guided Vehicle System (AGVS) has been widely used in many fields, such as storage and distribution centers, manufacturing workshops, and port transportation [1]. In the control process of the AGVS, many NP-hard problems need to be solved, including task scheduling and vehicle dispatching [2], path planning [3]–[5], and collision and deadlock prevention [6], [7]. Among them, the collision prevention is one of

the main challenges in the AGVS field. In order to reduce the number of conflicts between vehicles, a special guide path should be used [8]. For instance, the single-loop guide path [9] can be adopted to reduce the conflicts between multiple vehicles. Namely, in a single-loop guide path, all vehicles travel in a unidirectional loop, which connects all pick-up and drop points together [10]. Since there is no path intersection in a single-loop guide path, interferences between vehicles at intersection points are completely eliminated. However, since multiple vehicles are running in the same loop, one of them can interfere with the moving of the other vehicles when it is loading or unloading. The tandem configuration guide

The associate editor coordinating the review of this manuscript and approving it for publication was Tiago Cruz¹.

path [11] can be used to eliminate the interferences between vehicles completely. The tandem configuration guide paths consist of several non-overlapping single loops with load transfer-stations to connect with each other [12], where each loop includes only one vehicle. However, more than one vehicle can be required to transport a load from its origin to the destination, causing additional transfer time between vehicles, and thus increasing the load handling time [13]. The special guide paths have the shortcomings of low system efficiency and poor vehicle routing flexibility because each vehicle can operate only in one loop without the possibility of using alternative routes. Therefore, the special guide paths cannot meet the requirements of increasingly efficient and intelligent material transportation development.

The network-based guide paths can be divided into two categories: bidirectional guide path networks (BGNs) [14], and unidirectional guide path networks (UGNs) [15]. In a BGN, vehicles can travel in any direction on each path segment, which makes the BGN-based AGVS has high routing flexibility [16]. However, in the BGNs, vehicles are not allowed to travel in both directions at the same time, and the allowed traveling direction of each path segment has to be adjusted dynamically [17]. The strong connectivity of a BGN and the logistics path between stations cannot be guaranteed all the time [18]. Thus, when assigning a task to a vehicle, it is necessary to determine whether there is an accessible route between the vehicle and the pick-up/drop point of the task [19], which means that the task scheduling and vehicle dispatching of the BGN-based AGVS are highly related to the vehicle routing. Hence, it is sensible to integrate task scheduling, vehicle dispatching, and vehicle routing as an entire solution of the AGVS control [20]–[23]. However, due to such a complex control process, the application of the BGN-based AGVS is limited.

It is noteworthy that strong connectivity of a UGN and logistics path between stations should be ensured in the design phase, and the allowed traveling direction of a vehicle on each path segment should be fixed [24]. As a result, the complexity of the path planning process is reduced. However, as all vehicles share the same UGN, path conflicts between vehicles are inevitable. At present, there are many vehicle collision prevention methods, which can be divided into two categories. The first category of methods relies on centralized collision-avoidance path planning. For instance, the time-window based technology can be applied to coordinate trajectories of the vehicles to avoid vehicle conflicts [25]. However, due to the dynamics and uncertainty of the AGVS, vehicle delays and failures are likely to happen [26]. In that case, a vehicle cannot follow the planned trajectory [27]. This can further affect the effectiveness of vehicles sharing the overlapping paths at the same time slot, and even require the path re-planning [28]. Therefore, centralized path planning methods are applicable only to the small-scale or low-density AGVS [29]. As for the UGN, which represents a static map, vehicles can directly use the classic Dijkstra algorithm [30] or A-star algorithm [31] to realize the autonomous path planning

function. However, vehicles often conflict with each other when they are moving, so it is necessary to use an effective centralized traffic management approach to prevent collisions between vehicles. Therefore, the UGN-based AGVS can use the second-category methods to avoid collisions by combining the distributed autonomous path planning and centralized traffic management.

Currently, the most popular and widely used centralized traffic management is the zone controlled (ZC) method [32]–[34], which divides a guide-path network into several non-overlapping zones and allows at most one vehicle per zone a time. Thus, if a vehicle needs to enter another zone, it has to obtain the corresponding permission from the central controller [35]. Therefore, zone design is a crucial factor affecting the ZC method performance. Namely, using too large zones not only limits the AGVS scale but also leads to a long waiting time when changing the zone [36]. Therefore, using smaller zones can improve system liveness and efficiency [37], [38]. Malopolski *et al.* [39] proposed a square topology-based collision prevention method, which reduces the zones to small squares. However, since vehicles need to pause and wait for permission from the central controller before entering another zone, using too small zones will increase the number of vehicles pauses, thus prolonging the traveling time.

In addition, regardless of adopted collision avoidance strategy, a vehicle can be blocked by another vehicle and unable to move temporarily, and when several vehicles form a cyclic block, a deadlock can occur. Therefore, it is necessary to develop an efficient deadlock detection and prevention algorithm. Shao *et al.* [40] studied the collision and deadlock prevention problem of the BGN-based AGVS. Moorthy *et al.* [41] proposed a cyclic deadlock prediction and avoidance strategy for the AGVS to avoid the collisions between vehicles by using the ZC method. Wu *et al.* [42] addressed the deadlock problems of AGVS using the ZC method and proposed a resource-oriented Petri net-based deadlock prevention method, but this method can cause state explosions and is difficult to be applied to a large AGVS. Zhao *et al.* [6] studied the prediction and avoidance of deadlocks for a square-topology AGVS in a BGN and proposed a dynamic resource reservation-based method. However, the mentioned collision and deadlock prevention methods are inefficient in the UGN-based AGVS. To the best of the authors' knowledge, there have been fewer studies on the collision and deadlock avoidance problems in the UGN-based AGVS.

The main motivation of this article is to develop a collision and deadlock prevention method for the UGN-based AGVS. The main contributions of this work are twofold. On the one hand, according to the characteristics of the UGN-based AGVS, common types of collisions and deadlocks are introduced, and the corresponding collisions and deadlocks prevention methods are proposed. On the other hand, the shortcoming of the existing traffic management methods that do not optimize the AGV traffic sequence is

solved by developing a bidding mechanism-based strategy, which optimizes the vehicle traffic sequence of each intersection according to the dynamic state of the guide-path network and the urgency of handing tasks.

The rest of the paper is organized as follows. A typical application environment of the UGN-based AGVS is described in Section II. In Section III, a collision and deadlock avoidance method for the UGN-based AGVS is introduced. A bidding mechanism-based strategy for optimizing the traffic sequence at each intersection is proposed in Section IV. In Section V, extensive simulation experiments are presented to evaluate the performance of the proposed methods and strategies. Finally, the conclusions are summarized in Section VI.

II. ENVIRONMENT AND PROBLEM DESCRIPTION

A. ENVIRONMENT DESCRIPTION

The manufacturing system that uses the UGN-based AGVS as a material handling system is generally composed of a logistics transportation subsystem and a processing subsystem. An example of a manufacturing system is shown in Fig. 1, where the processing subsystem consists of 15 workstations, an input workstation, an output workstation, and 13 processing workstations. Each processing workstation includes a processing machine, an input buffer, and an output buffer with a finite capacity. The input workstation has only an output buffer, and the output workstation has only an input buffer. The logistics transportation subsystem includes a UGN and several single-load AGVs.

All workpieces enter the system via the input workstation. When a workpiece is processed in the manufacturing system, it is first delivered by the AGVs and then processed by machines. The processing workstation and the processing time of each task are known in advance. After a sequence of processing and handling tasks, the workpiece leaves the manufacturing system via the output workstation.

Important assumptions are given as follows:

- (1) All vehicles are single-load AGVs with the same running speed, and only one workpiece can be carried by a vehicle at a time.
- (2) After receiving a handling task from a central controller, a vehicle immediately plans the shortest route from its current position to the pick-up or drop point of the task using the Dijkstra algorithm. Once it starts to run, it does not change the route by itself.
- (3) If there is no unloading space in the input buffer when a vehicle reaches the drop point of the task, in order to avoid affecting the moving of other vehicles, the vehicle continues to move forward and then returns to the drop point along the shortest cycle route.
- (4) When a vehicle completes the assigned task, if there is no task to perform, it will move to the nearest parking path.
- (5) A processing machine processes only one workpiece at a time, and the processing machine of each workstation

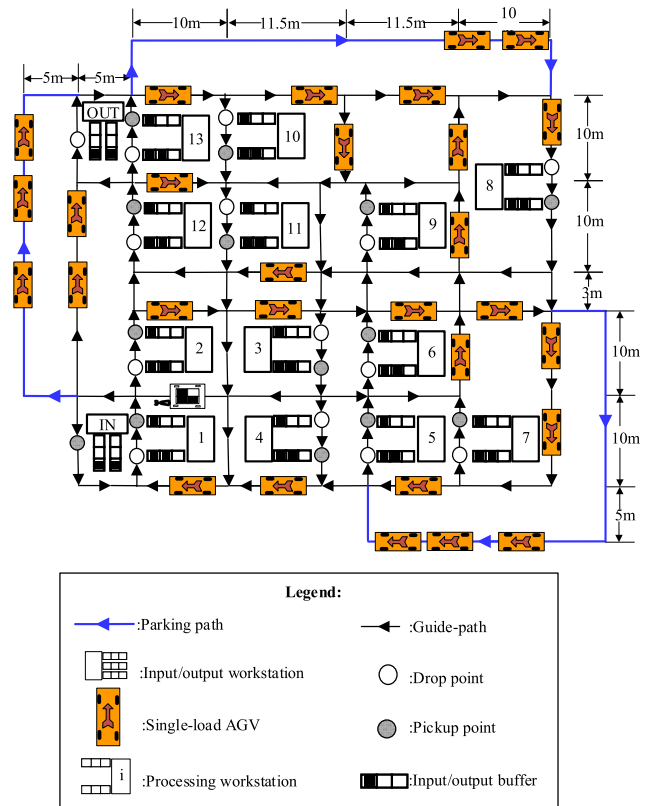


FIGURE 1. An example of a manufacturing system.

processes the workpieces in the input buffer based on the first-come-first-served principle.

- (6) The loading and unloading times of different types of workpieces are the same. When a vehicle is loading or unloading at a pick-up or drop point, other vehicles with similar assigned operations at this point have to wait until the vehicle finishes the loading or unloading process.

B. PROBLEM DESCRIPTION

Avoidance of collisions between vehicles is the first problem to be solved in AGVS traffic management. As shown in Fig. 2, three types of collisions are likely to happen in the UGN-based AGVS: catch-up collision, intersection collision, and special collision. The catching up collision, which is presented in Fig. 2(a), occurs when a vehicle is moving, but the path ahead is occupied by another stationary vehicle. The intersection collision, which is presented in Fig. 2(b), occurs when multiple vehicles enter the intersection simultaneously. However, in some special environments, other types of collisions can also happen. For instance, as shown in Fig. 2(c), due to the limitation on the workshop space, two guide paths are relatively close. Thus, when a vehicle makes a turn in one of the paths, if there is a vehicle in another path, the special collision can occur. In actual AGVS, every vehicle is generally equipped with necessary obstacle detection sensors that can detect obstacles in front of the vehicle. When a vehicle detects an obstacle ahead, it will automatically stop so that to avoid

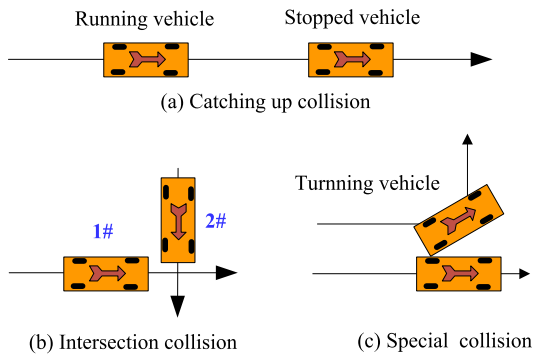


FIGURE 2. Common types of collisions in the UGN-based AGVS.

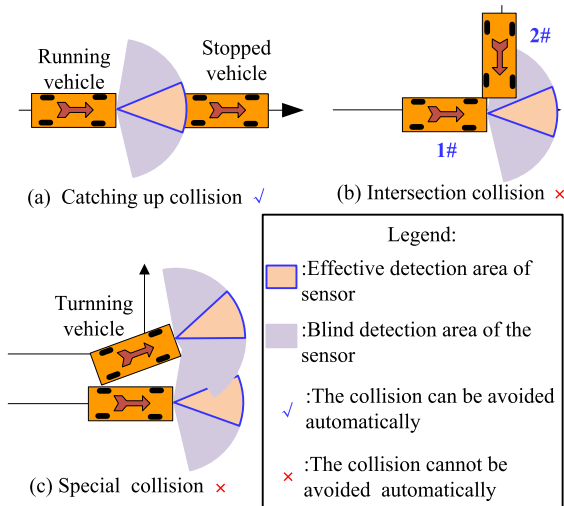


FIGURE 3. An illustration of the autonomous obstacle detection ability of a vehicle.

the catching up collisions, as shown in Fig. 3(a). However, due to the blind area of a sensor, obstacles on the side of a vehicle cannot be detected. Accordingly, a vehicle can avoid neither intersection collision nor special collision automatically, as presented in Figs. 3(b) and 3(c), respectively. Hence, how to avoid the intersection and special collisions is the primary problem to be solved.

Further, regardless of the adopted collision avoidance method, a vehicle can be blocked by other vehicles, which can temporarily disable its further moving. As mentioned previously, if several vehicles form a cycle block, a deadlock will occur. Fig. 4 shows two types of deadlocks in the UGN-based AGVS: intersection congestion deadlock and cycle deadlock. Fig. 4(a) shows a scenario where vehicle 2 cannot move because vehicle 1 is blocking the intersection. At the same time, vehicle 2 blocks the follow-up vehicles in turn, resulting in vehicle 1 occupying the intersection all the time, and finally, forming the intersection congestion deadlock. As shown in Fig. 4(b), when there is no space for any path segment in the closed-loop path, all vehicles will be blocked, and a cycle deadlock will eventually occur. Thus, the two mentioned types of deadlocks can paralyze the AGVS. Hence,

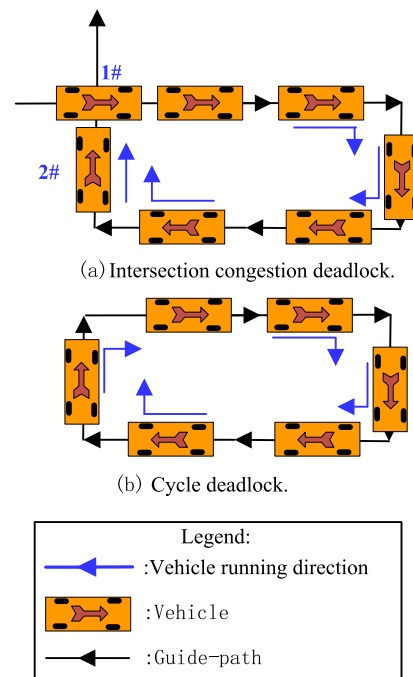


FIGURE 4. Two types of deadlocks in the UGN-based AGVS.

how to avoid deadlocks is another problem to be solved by the AGVS centralized traffic management controller.

In addition, the existing traffic management methods do not optimize traffic sequence at each path intersection, which not only can increase the probability of traffic congestion and deadlock but also can cause a delay in task distribution, thus affecting system performance. Consequently, the third problem to be solved is to develop a vehicle traffic sequence optimization strategy according to the AGV characteristics in the manufacturing environment.

III. COLLISION AND DEADLOCK AVOIDANCE METHODS

Three common conflicts that can cause collision between vehicles in the UGN-based AGVS are shown in Fig. 2 and explained in the previous section. Since each vehicle has a certain autonomous obstacle detection ability, the catching up collision, which is shown in Fig. 3(a), can be avoided by a vehicle autonomous control system. As explained in Section II, due to the blind area of a sensor, obstacles on the side of a vehicle cannot be detected, so the vehicle cannot avoid the intersection and special collisions automatically. In order to solve this problem, a vehicle coordination method based on the semaphore theory and IoT positioning technology, which can avoid collision between vehicles and eliminate the intersection congestion deadlocks, is proposed. Moreover, in order to avoid the cycle deadlocks, a search and avoidance algorithm based on the digraph theory is developed. Before introducing the proposed methods, the notations used in this paper are described in Table 1.

While passing through an intersection, vehicles occupy the intersection space. When more than one vehicle passes through the intersection simultaneously, an intersection

TABLE 1. The notations used in this paper.

Symbol	Meaning of the symbol
N_v	the number of path intersections
$G = (V, E)$	denotes a UGN
$V = \{v_0, v_1 \dots v_i \dots v_{N_v}\}$	denotes a set of intersections
$E = \{e_{i,j} v_i, v_j \in V, v_i \neq v_j\}$	denotes a set of path segments between intersections
$e_{i,j}$	represents a path segment starting from an intersection v_i and ending at an intersection v_j
N_G	the number of vehicles in the AGVS
$\Omega_G = \{AGV_k k \in N \text{ and } 1 \leq k \leq N_G\}$	a set of vehicles in the AGVS
$X_{i,k}$	the assignment relationship between a path intersection v_i and a vehicle AGV_k . The value of one means that vehicle AGV_k is authorized to enter intersection v_i ; the value of zero means that vehicle AGV_k is not authorized to enter intersection v_i
$d_{i,j}$	the length of path segment $e_{i,j}$
L_G	the vehicle length
F_G	the minimum safe distance between vehicles to be kept when the vehicles are moving
$C(e_{i,j})$	the maximum capacity of path segment $e_{i,j}$, which indicates the maximum number of vehicles that can be accommodated in path segment $e_{i,j}$, which can be expressed as: $C(e_{i,j}) = \text{Roundup}(\frac{d_{i,j}}{L_G + F_G})$, where $\text{Roundup}()$ denotes the round up operation function
$I(e_{i,j})$	the number of vehicles currently on path segment $e_{i,j}$
$R(e_{i,j}) = C(e_{i,j}) - I(e_{i,j})$	the remaining capacity of path segment $e_{i,j}$
$\Gamma(AGV_k) = \{e_{1,n}, e_{n,m} \dots e_{j,k}\}$	the route of a vehicle AGV_k from the current location to the target point of the current handing task; thus, the route can be computed by the Dijkstra algorithm once the task is assigned to a vehicle AGV_k
$P(AGV_k)$	the path segment where vehicle AGV_k is currently located at
$N(AGV_k)$	the path segment vehicle AGV_k enters next
$G(e_{i,j})$	the vehicle at the beginning of path segment $e_{i,j}$; it can also be expressed as a vehicle that is the closest to the exit of path segment $e_{i,j}$
$\Omega(e_{i,j})$	the set of path segments starting from intersection v_j , which is the ending intersection of path segment $e_{i,j}$
$N^\Omega(e_{i,j})$	the number of elements in set $\Omega(e_{i,j})$
$\partial_n(e_{i,j})$	the n -th element in set $\Omega(e_{i,j})$

collision can occur. Therefore, in order to avoid the intersection collision, each intersection should be occupied by only one vehicle at a time. Thus, in the AGVS, the following constraint has to be satisfied:

$$\forall v_j \in V \quad \text{has} \quad \sum_{k=1}^{N_G} X_{i,k} \leq 1. \quad (1)$$

The intersection congestion deadlock, which is presented in Fig. 4(a), is caused when vehicle 1 enters the intersection, but there is not enough space in the path section to be entered, resulting in the permanent occupation of the intersection by vehicle 1. Thus, in order to avoid intersection blocking, it is necessary to check whether there is enough space in the next path segment before a vehicle enters it. Accordingly, when a vehicle AGV_k enters section $e_{i,j}$, the following constraint has to be satisfied:

$$R(e_{i,j}) = (C(e_{i,j}) - I(e_{i,j})) \geq 1. \quad (2)$$

A. INTERSECTION COLLISION AND CONGESTION DEADLOCK AVOIDANCE METHOD

In order to satisfy the above constraints, a vehicle coordination method based on the semaphore theory and IoT positioning technology is proposed. First, a mutually exclusive semaphore for each intersection v_i and an integer semaphore

for each path segment $e_{i,j}$ are defined. The mutually exclusive semaphore consists of a counter $BS(v_i)$, a semaphore owner $\Omega_{BO}(v_i)$, and a list of semaphore waiters $\Omega_{BW}(v_i)$. The state of $BS(v_i)$ can be idle $BS(v_i) = 1$, or occupied $BS(v_i) = 0$. The semaphore owner $\Omega_{BO}(v_i)$ stores the ID of a vehicle that occupies intersection v_i . The list of semaphore waiters $\Omega_{BW}(v_i)$ stores the IDs of all vehicles that are waiting to enter intersection v_i . The integer semaphore includes a counter $IS(e_{i,j})$, a semaphore owner queue $\Omega_{IO}(e_{i,j})$, and a list of semaphore waiters $\Omega_{IW}(e_{i,j})$. The counter $IS(e_{i,j})$ indicates the remaining capacity of a path segment $e_{i,j}$. The semaphore owner queue $\Omega_{IO}(e_{i,j})$ stores the IDs of all vehicles that are currently in a path segment $e_{i,j}$ following the first-in-first-out (FIFO) principle. The list of semaphore waiters $\Omega_{IW}(e_{i,j})$ stores the IDs of all vehicles that are waiting for entering a path segment $e_{i,j}$.

Then, the QR code landmarks are set at all entrances and exits of intersections to identify semaphores, as shown in Fig. 5. The QR code landmark contains its location and the corresponding semaphore identification information. Each vehicle is equipped with a QR code signpost recognition device, which can collect the location and semaphore identification information stored in the QR code landmark. When a vehicle detects the QR code landmark at the entrance of an intersection, it immediately stops and accesses to the

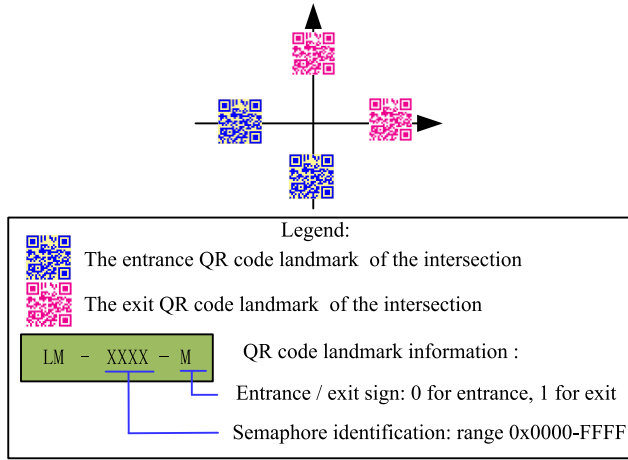


FIGURE 5. The QR code landmark of an intersection.

central traffic management controller for the semaphores of the intersection and path segment to enter next. The vehicle waits until it is authorized to enter the next path segment through the intersection.

For instance, suppose a vehicle AGV_k stops at the beginning of a path segment $e_{m,i}$ and waits to enter a path segment $e_{i,j}$ through intersection v_i . A vehicle coordination method that is shown in Fig. 6 is used to coordinate the running of vehicles at intersections. This method not only ensures that the intersection is occupied by at most one vehicle, but also ensures that the intersection is not blocked. Therefore, both the intersection collision and the intersection congestion deadlock are avoided. The steps of the vehicle coordination method are as follows.

Step 1: Define a mutually exclusive semaphore for each intersection v_i in the intersection set V , and let $BS(v_i) = 1$, $\Omega_{BO}(v_i) = \emptyset$, and $\Omega_{BW}(v_i) = \emptyset$. Define an integer semaphore for each path segment $e_{i,j}$ in the path segment set E , and let $IS(e_{i,j}) = C(e_{i,j})$, $\Omega_{IO}(e_{i,j}) = \emptyset$, and $\Omega_{IW}(e_{i,j}) = \emptyset$; then, go to Step 2.

Step 2: Wait until a vehicle AGV_k arrives at the beginning of a path segment having intersection v_j as its ending intersection. Then, add vehicle AGV_k to the list of semaphore waiter sets $\Omega_{BW}(v_i)$ and $\Omega_{IW}(e_{i,j})$; then, go to Step 3.

Step 3: If $\Omega_{BW}(v_i) \neq \emptyset$, go to Step 4; else, go to Step 2.

Step 4: Use a traffic sequence optimization strategy to determine a vehicle with the first crossing right from $\Omega_{BW}(v_i)$. Assume vehicle AGV_k is waiting to enter the path segment $e_{i,j}$; then, go to Step 5.

Step 5: If $BS(v_i) = 1$ and $IS(e_{i,j}) > 1$, then intersection v_i is not occupied by another vehicle and the path segment has enough space for vehicle AGV_k . Next, authorize vehicle AGV_k to enter the path segment $e_{i,j}$ through intersection v_i , and update the status of related semaphores, including those of the path intersection v_i , the current path segment $P(AGV_k)$, and the next path segment $e_{i,j}$. Let $\Omega_{BW}(v_i) = \Omega_{BW}(v_i) / \{AGV_k\}$, $\Omega_{BO}(v_i) = \{AGV_k\}$, $BS(v_i) = 0$; $\Omega_{IO}(P(AGV_k)) = \Omega_{IO}(P(AGV_k)) / \{AGV_k\}$,

$IS(P(AGV_k)) = IS(P(AGV_k)) + 1$; $\Omega_{IW}(e_{i,j}) = \Omega_{IW}(e_{i,j}) / \{AGV_k\}$, $\Omega_{IO}(e_{i,j}) = \Omega_{IO}(e_{i,j}) \cup \{AGV_k\}$, and $IS(e_{i,j}) = IS(e_{i,j}) - 1$; then, go to Step 6; otherwise, go to Step 3.

Step 6: Vehicle AGV_k occupies the mutually exclusive semaphore of intersection v_i until the exit QR code landmark is acquired. Then, the vehicle requests the central traffic management controller to release the semaphore. Next, let $\Omega_{BO} = \emptyset$ and $BS(v_i) = 1$, and go to Step 7.

Step 7: Vehicle AGV_k occupies the integer semaphore of a path segment $e_{i,j}$ until it is authorized to exit the path segment $e_{i,j}$. Then, the vehicle requests the central traffic management controller to release the semaphore. Next, let $IS(e_{i,j}) = IS(e_{i,j}) + 1$ and $\Omega_{IO}(e_{i,j}) = \Omega_{IO}(e_{i,j}) / \{AGV_k\}$, and go to Step 3.

B. SPECIAL COLLISION AVOIDANCE METHOD

As shown in Fig. 2(c), due to the limitation on the workshop space, the two paths are relatively close. In order to avoid the occurrence of a special collision, the partially overlapping area is regarded as a virtual intersection. The QR code landmarks are set at all entrances and exits of the virtual intersection, as shown in Fig. 7. Then, an exclusive semaphore is defined for the virtual intersection, and the vehicle coordination flow similar to that in Fig. 6 is applied to ensure that the overlapping area can be occupied by one vehicle at most a time. Although there may be other types of conflicts between vehicles under special conditions, they can also be solved by adding virtual intersections.

C. CYCLE DEADLOCK AVOIDANCE METHOD

In order to analyze the conditions that lead to deadlocks, first, the cause of vehicle block is analyzed using the previously-presented vehicle coordination method based on the semaphore theory and IoT positioning technology. If a vehicle at the beginning of a path segment is not blocked, the vehicles behind that vehicle will not be blocked. According to the proposed vehicle coordination method, there are two reasons for a vehicle to be blocked at the beginning of a path segment. The first is that the intersection that the vehicle waits to enter is temporarily occupied by another vehicle. The proposed vehicle control method allows a vehicle to enter the intersection only when the next path segment has the free capacity, so the intersection will not be occupied by a vehicle for a long time, and it is impossible to form a cycle deadlock. The second reason for a vehicle be blocked is that the requested path segment has no free capacity, which is the main cause of a cycle deadlock. With the aim to solve the problem of cycle deadlock caused by the insufficient capacity of a path segment, an AGVS running state model based on the directed graph theory is developed, and based on this model, a cycle deadlock search algorithm is proposed. Finally, two critical states of a deadlock are defined, and the corresponding deadlock avoidance strategies are presented.

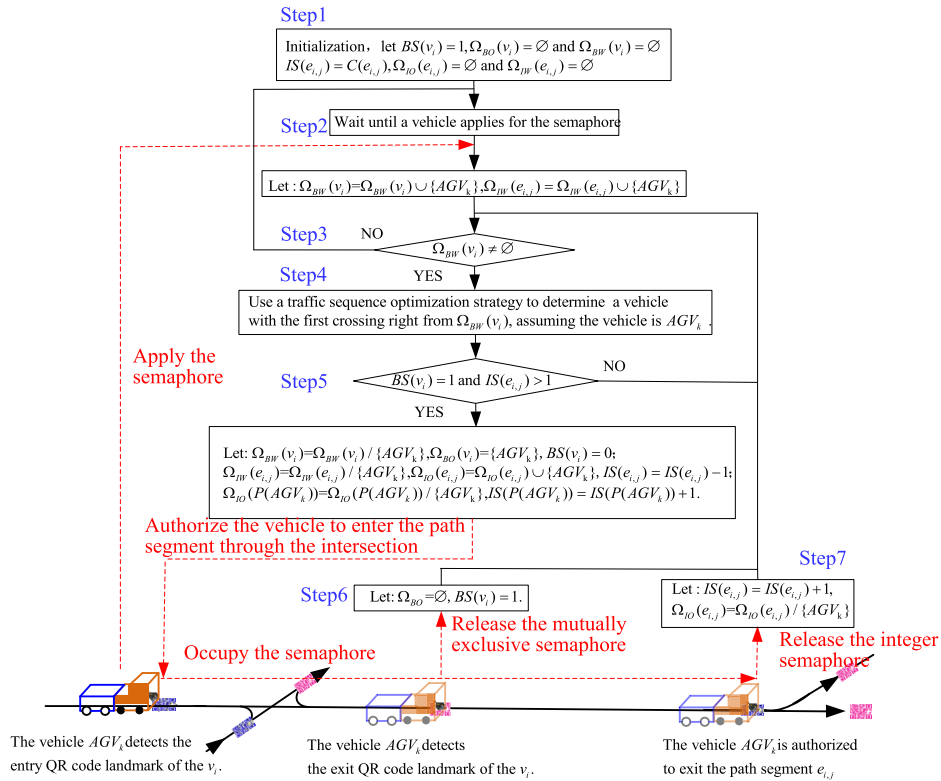


FIGURE 6. Flowchart of the vehicle coordination method.

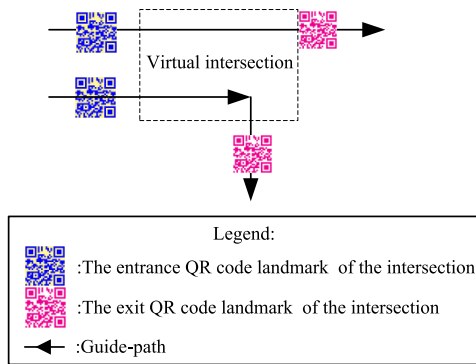


FIGURE 7. The virtual intersection setting.

1) DIRECTED GRAPH-BASED CYCLE DEADLOCK SEARCH ALGORITHM

The AGVS running status model based on the directed graph theory is established to detect a cycle deadlock. In order to reflect the blocking relationship between vehicles and the vehicle blocking loops in the AGVS intuitively, the definition of a vehicle blocking digraph is given.

Definition 1 (Vehicle Blocking Digraph $G_Q(V_Q, E_Q, D_Q)$): The vehicle blocking digraph is a directed graph with attribute values. Its vertex set V_Q , edge set E_Q , and attribute set D_Q are respectively defined as:

$$V_Q = \{e_{i,j} | e_{i,j} \in E \text{ and } I(e_{i,j}) > 0\}. \quad (3)$$

$$E_Q = \left\{ (e_{i,j}, e_{i,k}) \mid \left(G(e_{i,j}) = AGV_k \text{ then } N(AGV_k) = e_{j,k} \right) \right\}. \quad (4)$$

$$D_Q = \{R(e_{i,j}) \mid e_{i,j} \in V_Q\}. \quad (5)$$

According to (3), the vertex in the vehicle blocking digraph is a path segment in the UGN, and only the path segment with at least one vehicle in it can be a vertex. Further, according to (4), the adjacency relationship between vertices in the vehicle blocking digraph depends on the routing of a vehicle at the beginning of a path segment. Namely, edge $(e_{i,j}, e_{i,k})$ will appear in the vehicle blocking digraph only when a vehicle at the beginning of path segment $e_{i,j}$ waits to enter the path segment $e_{i,k}$. The attribute value $R(e_{i,j})$ of each vertex in the attribute set D_Q denotes the remaining capacity of the corresponding path segment.

Example 1: The AGVS status at time t and the corresponding vehicle blocking digraph model.

The AGVS status at time t is displayed in Fig. 8(a), which shows the path segment $(e_{5,2})$ to be entered by two vehicles (AGV#1 and AGV#2) at the beginning of two path segments $(e_{4,5}$ and $e_{8,5})$. The corresponding vehicle blocking digraph model is shown in Fig. 8(b). Since there are no vehicles on path segments $e_{5,6}$, $e_{7,8}$, and $e_{4,7}$, these segments do not appear in the vehicle blocking digraph model. In addition,

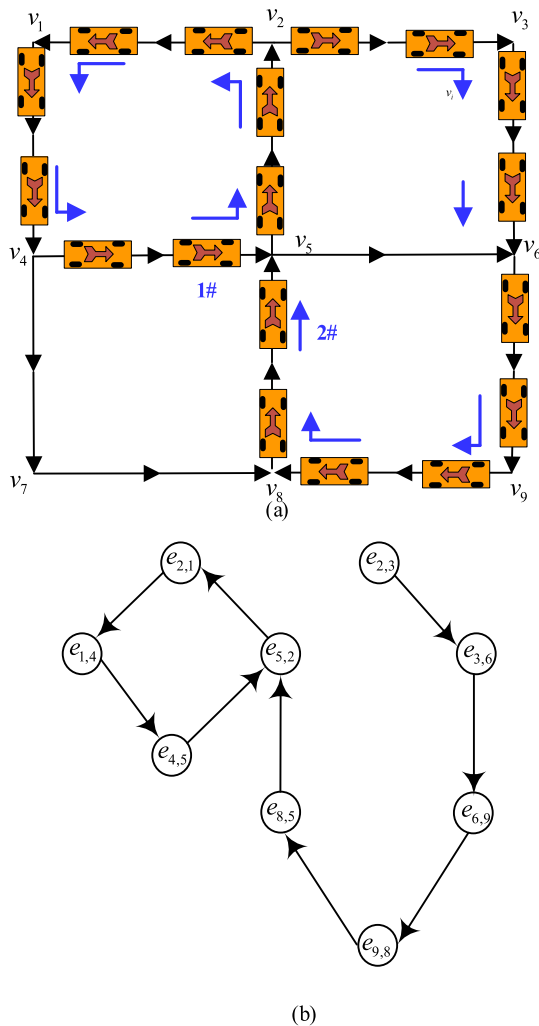


FIGURE 8. The AGVS status at time t and the corresponding vehicle blocking digraph model. (a) the AGVS status at time t ; (b) the corresponding vehicle blocking digraph model.

since no vehicle is waiting to enter path segment $e_{2,3}$, vertex $e_{2,3}$ in the blocking graph has no input edge.

Definition 2: For an edge $(e_{i,j}, e_{j,k})$ in the vehicle blocking digraph $G_Q(V_Q, E_Q)$, vertex $e_{i,j}$ represents the starting point of edge $(e_{i,j}, e_{j,k})$, and vertex $e_{j,k}$ represents the end point of edge $(e_{i,j}, e_{j,k})$. Vertex $e_{j,k}$ is called the adjacent forward node of vertex $e_{i,j}$ and is denoted as $Adj^+(e_{i,j}) = e_{j,k}$. Vertex $e_{i,j}$ is called the backward adjacent node of vertex $e_{j,k}$ and is denoted as $Adj^-(e_{j,k}) = e_{i,j}$.

Definition 3 (Vehicle Blocking Loop): A vehicle blocking loop LP denotes a directed closed loop in the vehicle blocking digraph, so it is represented by an ordered adjacent vertex set of the vehicle blocking digraph and denoted as $LP = \{e_{i,j}, e_{j,k}, e_{k,m} \dots e_{n,i}\}$. For any two adjacent vertices $e_{j,k}$ and $e_{k,m}$ in the vehicle blocking loop LP , it holds that $(e_{i,j}, e_{j,k}) \in E_Q$.

Theorem 1: If a vehicle does not change its own path during the moving process, any vertex in the vehicle blocking digraph belongs to at most one vehicle blocking loop.

Proof: If a vehicle does not change its own path during the moving process, the next path segment of the vehicle at the beginning of each path segment is unique. Therefore, for any vertex in the vehicle blocking digraph, there is at most one edge starting from this vertex. Obviously, any vertex belongs to only one vehicle blocking loop.

Definition 4 (Remaining Capacity of Vehicle Blocking Loop R_{LP}): The remaining capacity of a vehicle blocking loop represents the sum of the remaining capacities of all the path segments of the blocking loop, which is expressed as: $R_{LP} = \sum R(e_{i,j}), \forall e_{i,j} \in LP$.

Theorem 2: If the remaining capacity of a vehicle blocking loop R_{LP} is zero, the vehicle blocking loop $LP = \{e_{i,j}, e_{j,k}, e_{k,m} \dots e_{n,i}\}$ is a cycle deadlock.

Proof: If the remaining capacity of a vehicle blocking loop R_{LP} is zero, then the remaining capacity of a path segment applied to enter by a vehicle at the beginning of each path segment in the vehicle blocking loop is zero. According to the proposed vehicle coordination method, the vehicle at the beginning of each path segment in the vehicle blocking loop is prohibited from moving. Therefore, the vehicle blocking loop $LP = \{e_{i,j}, e_{j,k}, e_{k,m} \dots e_{n,i}\}$ is a cycle deadlock.

When a vehicle AGV_k applies to enter a path segment $e_{i,j}$, the central traffic management controller has to determine whether a cycle deadlock will be caused if AGV_k is allowed to enter the applied path segment $e_{i,j}$. The cycle deadlock search procedure is presented in Algorithm 1.

Algorithm 1 The Cycle Deadlock Searching Procedure

- 01: **Input:** A vehicle currently requesting the semaphores, AGV_k ; the applied path segment, $e_{i,j}$.
- 02: **Output:** Is there a cycle deadlock? And the cycle deadlock found, LP
- 03: **Begin**
- 04: Assume that the vehicle has entered the path segment, and build the vehicle blocking digraph of the AGS; $e_{cu} = e_{i,j}; LP = \{e_{i,j}\}$.
- 05: **If** $R(e_{cu}) > 0$ **then**
- 06: go to **line 19**
- 07: **Else**
- 08: **If** $\exists Adj^+(e_{cu})$ **then**
- 09: **If** $Adj^+(e_{cu}) \neq e_{i,j}$ **then**
- 10: $e_{cu} = Adj^+(e_{cu});$
 $LP = LP \cup \{Adj^+(e_{cu})\}$.
- 11: go to **line 5**
- 12: **Else**
- 13: go to **line 20**
- 14: **End if**
- 15: **else**
- 16: go to **line 19**
- 17: **End if**
- 18: **End if**
- 19: Return (There is no cycle deadlock)
- 20: Return (Cycle deadlock exists)
- 21: **End**

In Algorithm 1, e_{cu} denotes the path segment currently being searched in the vehicle blocking digraph, and $Adj^+(e_{cu})$ denotes the adjacent forward vertex of a vertex e_{cu} . If $R(e_{cu})$ is greater than zero, or e_{cu} has no $Adj^+(e_{cu})$, there is no cycle deadlock in the vehicle blocking digraph. According to Theorem 1, any vertex in a vehicle blocking digraph belongs to at most one loop. Therefore, the algorithm needs to traverse at most all the nodes in the vehicle blocking digraph. Thus, the algorithm complexity is $O(N)$, where N denotes the number of vertexes in the vehicle blocking digraph. According to the definition of the vehicle blocking digraph (Definition 1), N is smaller than the number of path segments or the number of vehicles N_G .

2) TWO TYPES OF CYCLE DEADLOCK CRITICAL STATES AND THE CORRESPONDING DEADLOCK AVOIDANCE STRATEGIES

When a vehicle applies to enter a path segment, the central traffic management controller evaluates whether that will cause a cycle deadlock using the procedure presented in Algorithm 1. If there is a cycle deadlock, the AGVS is in a cycle deadlock critical state, and the corresponding deadlock avoidance strategy must be adopted. According to the AGVS state characteristics, there are two types of deadlock critical states: single-cycle deadlock critical state and multi-cycle deadlock critical state.

Definition 5 (Single- or Multi-Cycle Deadlock CRITICAL STATE): Assume that allowing vehicle 1 to enter a path segment $e_{i,j}$ will cause a cycle deadlock $P = \{e_{i,j}, e_{j,k}, e_{k,m} \dots e_{n,i}\}$. According to the definition of the vehicle blocking digraph (Definition 1), there must be another vehicle, e.g., vehicle 2, at the beginning of path segment $e_{n,i}$ also applying to enter path segment $e_{i,j}$. If allowing vehicle 2 to enter path segment $e_{i,j}$ causes another cycle deadlock, the AGVS is in a multi-cycle critical deadlock state; otherwise, it is in a single-cycle deadlock critical state.

Example 2: Single-cycle deadlock critical state.

An example of a single-cycle deadlock critical state is shown in Fig. 9(a). In Fig. 9, each path segment has a capacity of two, and all vehicles in the same path segment have the same next path segment. At this time, vehicle 1 applies to enter path segment $e_{5,2}$. The AGVS status after vehicle 1 enters path segment $e_{5,2}$ is shown in Fig. 9(b), and the corresponding vehicle blocking digraph is shown in Fig. 9(d), where exists a vehicle blocking loop $LP = \{e_{5,2}, e_{2,1}, e_{1,4}, e_{4,5}\}$ with $R_{LP} = 0$. This means that $LP = \{e_{5,2}, e_{2,1}, e_{1,4}, e_{4,5}\}$ is a cycle deadlock. Obviously, vehicle 2 at the beginning of path segment $e_{4,5}$ also wants to enter $e_{5,2}$. If the vehicle entering path segment $e_{5,2}$ is vehicle 2 instead of vehicle 1, as shown in Fig. 9(c), the vehicle blocking digraph will be as shown in Fig. 9(d). Although the vehicle blocking loop $LP = \{e_{5,2}, e_{2,1}, e_{1,4}, e_{4,5}\}$ still exists in the vehicle blocking digraph, the remaining capacity of $LP = \{e_{5,2}, e_{2,1}, e_{1,4}, e_{4,5}\}$ is one. Therefore, $LP = \{e_{5,2}, e_{2,1}, e_{1,4}, e_{4,5}\}$ is not a cycle deadlock. Since a cycle deadlock occurs only when vehicle 1 is allowed to enter the

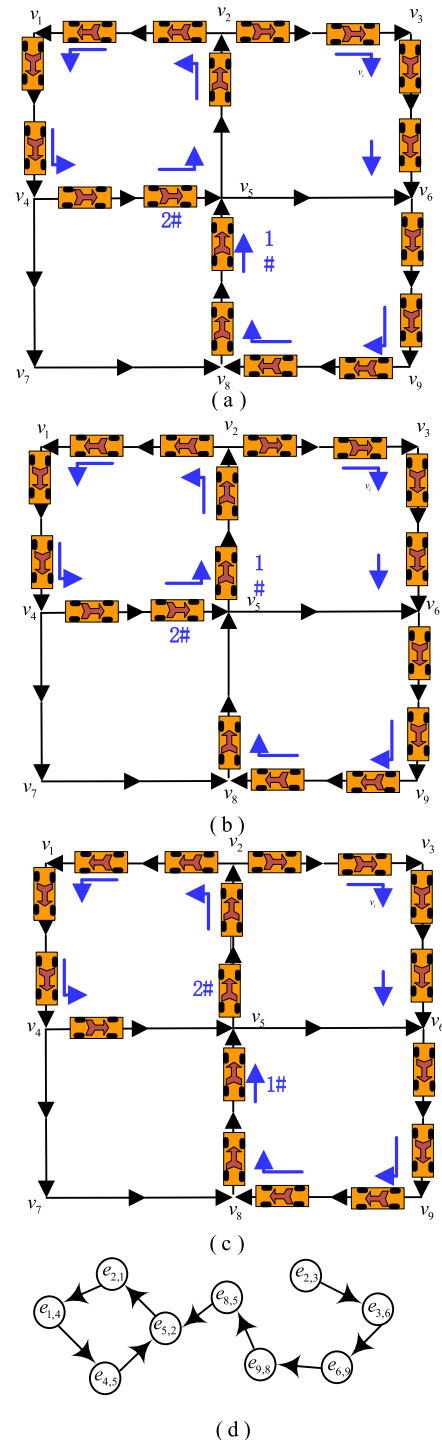


FIGURE 9. Example of a single-cycle deadlock critical state. (a) A single-cycle deadlock critical state; (b) the AGVS status after vehicle 1 enters path segment $e_{5,2}$; (c) the AGVS status after vehicle 2 enters path segment $e_{5,2}$; (d) the corresponding vehicle blocking digraph model.

next path segment, the AGVS is in a single-cycle deadlock critical state.

Example 3: Multi-cycle deadlock critical state.

An example of a multi-cycle deadlock critical state is shown in Fig. 10(a). In Fig. 10, the capacity of each path segment is one, and the path segment to be entered by the

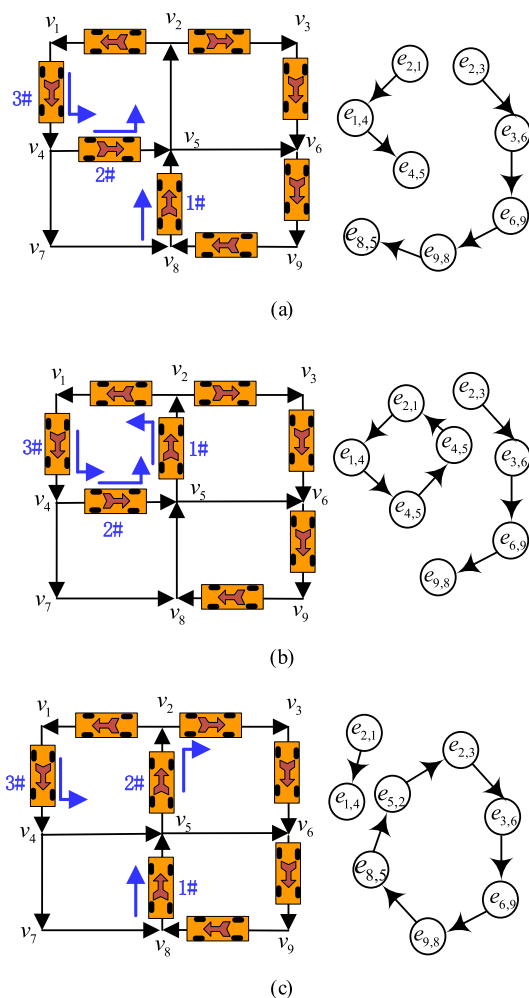


FIGURE 10. Example of a multi-cycle deadlock critical state. (a) A multiple cycle deadlock critical state and its corresponding vehicle blocking digraph; (b) the AGVS status after vehicle 1 enters $e_{5,2}$ and its corresponding vehicle blocking digraph; (c) the AGVS status after vehicle 2 enters $e_{5,2}$ and its corresponding vehicle blocking digraph.

vehicle at the beginning of each path segment is shown in Fig. 10(a). The routes of vehicles 1 and 2 are represented as $\Gamma(AGV_1) = \{e_{8,5}, e_{5,2}, e_{2,1}\}$ and $\Gamma(AGV_2) = \{e_{4,5}, e_{5,2}, e_{2,3}\}$, respectively. At this time, vehicle 1 applies to enter path segment $e_{5,2}$. The AGVS status after vehicle 1 enters path segment $e_{5,2}$ and the corresponding vehicle blocking digraph are shown in Fig. 10(b), where it can be seen that there is a cycle deadlock $LP = \{e_{5,2}, e_{2,1}, e_{1,4}, e_{4,5}\}$. Obviously, vehicle 2 at the beginning of path segment $e_{4,5}$ also wants to enter path segment $e_{5,2}$. If the vehicle entering path segment $e_{5,2}$ is vehicle 2 instead of vehicle 1, the AGVS status and the corresponding vehicle blocking digraph are as shown in Fig. 10(c), where it can be seen that there is a cycle deadlock $LP = \{e_{5,2}, e_{2,3}, e_{3,6}, e_{6,9}, e_{9,8}, e_{8,5}\}$. Regardless of which vehicle is allowed to enter the next path segment, a cycle deadlock will occur. Therefore, the AGVS is in a multi-cycle deadlock critical state.

a) Deadlock avoidance strategy for AGVS in a single-cycle deadlock critical state

If vehicle 1 is allowed to enter path segment $e_{i,j}$, a cycle deadlock $LP = \{e_{i,j}, e_{j,k}, e_{k,m} \dots e_{n,i}\}$ will be caused. According to the definition of the vehicle blocking digraph (Definition 1), there must be another vehicle, e.g., vehicle 2, at the beginning of path segment $e_{n,i}$, which is also applying to enter the same path segment $e_{i,j}$. Then, cycle deadlock $LP = \{e_{i,j}, e_{j,k}, e_{k,m} \dots e_{n,i}\}$ can be avoided by allowing vehicle 2 to enter the path section instead of vehicle 1. According to the definition of a single-cycle deadlock critical state, allowing vehicle 2 to enter path segment $e_{i,j}$ will not cause additional cycle deadlock.

b) Deadlock avoidance strategy for AGVS in a multi-cycle deadlock critical state

If vehicle 1 is allowed to enter a path segment $e_{i,j}$, a cycle deadlock $LP = \{e_{i,j}, e_{j,k}, e_{k,m} \dots e_{n,i}\}$ will be caused, and the AGVS will be in a multi-cycle deadlock critical state. In this case, if the deadlock avoidance strategy for AGVS in a single-cycle deadlock critical state is adopted, additional cycle deadlock will occur. In order to avoid creating an additional cycle deadlock, the procedure shown in Algorithm 2 is used to re-plan the vehicle route in the loop. This strategy is not a universal solution $LP = \{e_{i,j}, e_{j,k}, e_{k,m} \dots e_{n,i}\}$ for the AGVS in a multi-cycle deadlock critical state because it fails when all the path segments connected to the loop belong to the other deadlock loops. However, it should be noted that the density of AGVs in a general manufacturing environment is moderate, and the probability of all path segments belonging to deadlock loops at the same time is very low, which makes the procedure presented in Algorithm 2 applicable to most of the cases.

Algorithm 2 The Route re-Planning Procedure

- 01: **Input:** Vehicle blocking loop, $LP = \{e_{i,j}, e_{j,k}, e_{k,m} \dots e_{n,i}\}$.
- 02: **Output:** A re-routed vehicle in the loop.
- 03: **Begin**
- 04: $e_{c,u} = e_{i,j}$;
- 05: **If** $G(e_{i,j}) \neq \emptyset$ **then**
- 06: **For** $n = 1$ to $N^\Omega(e_{i,j})$ **loop**
- 07: **If** $\partial_n(e_{i,j}) \notin LP$ and $R(\partial_n(e_{i,j})) > 0$ **then**
- 08: Take a path segment $\partial_n(e_{i,j})$ as the next passing path of a vehicle $G(e_{i,j})$, and use the Dijkstra algorithm to re-plan the route for vehicle $G(e_{i,j})$.
Return (0).
- 09: **End if**
- 10: **Next** n
- 11: **End if**
- 12: $e_{c,u} = Adj^+(e_{c,u})$ go to **line 5**
- 13: **End**

Example 4: Deadlock avoidance strategy for AGVS in a multi-cycle deadlock critical state.

Take AGVS in a multi-cycle deadlock critical state shown in Fig. 10(a) as an example. The residual routes of vehicles 1, 2, and 3 can be represented as

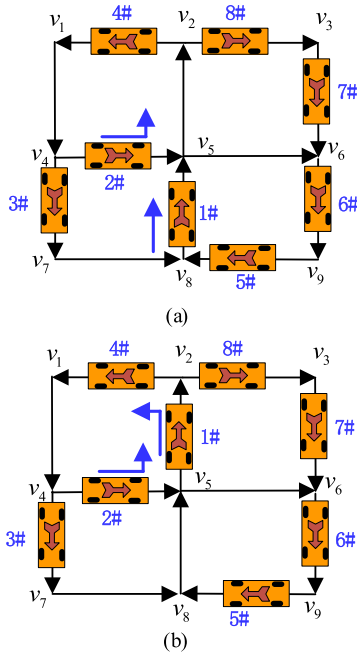


FIGURE 11. Example of the deadlock avoidance strategy for AGVS in a multi-cycle deadlock critical state. (a) The AGVS status after vehicle 3 enters $e_{4,7}$; (b) the AGVS status after vehicle 1 enters $e_{5,2}$.

$\Gamma(AGV_1) = \{e_{8,5}, e_{5,2}, e_{2,1}\}$, $\Gamma(AGV_2) = \{e_{4,5}, e_{5,2}, e_{2,3}\}$, and $\Gamma(AGV_3) = \{e_{1,4}, e_{4,5}, e_{5,6}\}$, respectively. In this case, allowing vehicle 1 to enter a path segment $e_{i,j}$ will cause a cycle deadlock $LP = \{e_{5,2}, e_{2,1}, e_{1,4}, e_{4,5}\}$. Therefore, Algorithm 2 is used to re-plan the vehicle route in the loop $LP = \{e_{5,2}, e_{2,1}, e_{1,4}, e_{4,5}\}$. After the re-planning procedure is completed, the residual route of vehicle 3 is updated, and it can be represented as $\Gamma(AGV_3) = \{e_{1,4}, e_{4,7}, e_{7,8}, e_{8,5}, e_{5,6}\}$. Then, vehicle 3 is authorized to enter path segment $e_{4,7}$. The AGVS status after vehicle 3 enters path segment $e_{4,7}$ is shown in Fig. 11(a), where it can be seen that a cycle deadlock $LP = \{e_{5,2}, e_{2,1}, e_{1,4}, e_{4,5}\}$ is avoided. Also, a cycle deadlock $LP = \{e_{5,2}, e_{2,3}, e_{3,6}, e_{6,9}, e_{9,8}, e_{8,5}\}$ can be avoided after vehicle 1 enters path segment $e_{5,2}$, as shown in Fig. 11(b).

IV. TRAFFIC SEQUENCE OPTIMIZATION STRATEGY

As mentioned previously, the existing traffic management methods do not optimize the traffic sequence of all path intersections, which affects and mostly reduces system performance. In order to solve this problem, a bidding mechanism-based strategy is proposed to optimize the vehicle traffic sequence in all path intersections. First, a delivery task urgency evaluation model of a vehicle is constructed to reflect the impact of vehicle tasks on the processing subsystem performance. Also, a traffic urgency evaluation model of a vehicle at the beginning of a path segment is proposed to reflect the influence of vehicle movement on the logistics transportation subsystem performance. Finally, based on the evaluation models of task urgency and traffic urgency, a comprehensive bidding model of vehicle is constructed and

TABLE 2. The notations used in this paper.

Symbol	Meaning of the symbol
$W(m)$	a processing workstation m
$C_i(m)$	the input buffer capacity of $W(m)$
$c_i(m)$	the current number of workpieces in the input buffer of $W(m)$
$C_o(m)$	the output buffer capacity of $W(m)$
$c_o(m)$	the current number of workpieces in the output buffer of $W(m)$
T_{ij}^n	the workpiece handling task with $W(j)$ as a destination workstation, and $W(i)$ as a source workstation. This task is the n th handling task once the system starts operating.
$N^{\Omega_{Bw}}(v_i)$	the number of vehicles in the semaphore waiting list $\Omega_{Bw}(v_i)$.
$AGV_k^{\Omega_{Bw}}(v_i)$	the k th vehicle in the semaphore waiting list $\Omega_{Bw}(v_i)$.

used to optimize the vehicle traffic order at each intersection. Before introducing the proposed traffic sequence optimization strategy, the notations are presented in Table 2.

A. TASK URGENCY EVALUATION MODEL OF VEHICLE

The task urgency of a vehicle is an metric that reflects the impact of vehicle tasks on the processing subsystem. The processing subsystem studied in this work consists of a series of processing workstations, where each processing workstation consists of a processing machine, an input buffer, and an output buffer with a limited capacity. Namely, in a practical scenario, the limited buffer capacity of each workstation cannot be ignored. On the one hand, if the output buffer of a processing workstation is full because the workpieces stored in it are not moved out in time, the processing machine will be blocked due to a lack of storage space. On the other hand, if the input buffer of a processing workstation has no workpiece waiting to be processed, the processing machine will be starved. Both the blocking and the starvation of a processing machine reduce the efficiency of the processing subsystem.

In order to improve the efficiency of the processing subsystem, it is necessary to optimize the traffic sequence according to the states of input and output buffers for a task carried by a vehicle. If there is a little capacity left in the output buffer for a task undertaken by a vehicle, or there are few workpieces in the input buffer for the task, the vehicle should pass the intersection. Therefore, the task urgency of a vehicle can be evaluated according to the state of the corresponding input or output buffer for the task that is currently carried by the vehicle. Depending on whether the vehicle is performing a workpiece handling task, and whether there is a workpiece on the vehicle, the vehicle state can be idle, busy but empty, or busy and loaded. The task urgency evaluation functions for the three vehicle states are as follows.

1) TASK URGENCY EVALUATION FUNCTION OF AN IDLE VEHICLE

The idle vehicle refers to a vehicle that has not been assigned a workpiece handling task. According to Assumption 4, if there

is no workpiece to carry, the vehicle will move to the nearest parking path. So, for the AGVS type studied in this paper, all vehicles with the parking path destination represent idle vehicles. Also, since no workpiece handling task is assigned to the vehicle, the vehicle has no effect on the processing subsystem. Therefore, the task urgency of an idle vehicle can be defined as:

$$U_T (AGV_k) = 0. \quad (6)$$

2) TASK URGENCY EVALUATION FUNCTION OF A BUSY BUT EMPTY VEHICLE

A busy but empty vehicle refers to a vehicle that has been assigned a workpiece handling task T_{ij}^n , but has not arrived at the pick-up point of the source workstation yet. Because the vehicle operation affects not only the workpiece loading time from $W(i)$, but also the time when the workpiece reaches $W(j)$, the output buffer state of $W(i)$ and the input buffer state of $W(j)$ can be used to calculate the task urgency of a busy but empty vehicle, and it is calculated by:

$$U_T (AGV_k) = \frac{\left(1 - \frac{c_I(j)}{C_I(j)}\right) + \frac{c_O(i)}{C_O(i)}}{2}. \quad (7)$$

3) TASK URGENCY EVALUATION FUNCTION OF A BUSY AND LOADED VEHICLE

A busy and loaded vehicle is a vehicle that has taken the assigned workpiece. Once a vehicle AGV_k has left the source workstation, the operation of vehicle AGV_k will affect only the status update time of the input buffer of the destination workstation $W(j)$. Therefore, the task urgency of the busy and loaded vehicle is expressed as:

$$U_T (AGV_k) = 1 - \frac{c_I(j)}{C_I(j)}. \quad (8)$$

B. TRAFFIC URGENCY EVALUATION MODEL OF VEHICLE

The traffic urgency of a vehicle represents an index that reflects the influence of vehicle movement on the logistics transportation subsystem. In a UGN-based AGVS, in order to reduce the risk of deadlock and vehicle congestion, it is necessary to improve the balance of vehicle distribution in the UGN. Accordingly, if the remaining capacity of a path segment where a vehicle is located is small and the remaining capacity of the next path segment is large, the vehicle should pass the intersection, and the traffic urgency evaluation function of a vehicle is given by:

$$U_S (AGV_k) = \frac{\left(1 - \frac{I(N(AGV_k))}{C(N(AGV_k))}\right) + \frac{I(P(AGV_k))}{C(P(AGV_k))}}{2}. \quad (9)$$

C. BIDDING MECHANISM-BASED TRAFFIC SEQUENCE OPTIMIZATION STRATEGY

If a vehicle at the beginning of a path segment needs to enter an intersection v_i with multiple input path segments, as shown in Fig. 12, the vehicle at the beginning of each path segment will be in the semaphore waiting list $\Omega_{B_w}(v_i)$. Then,

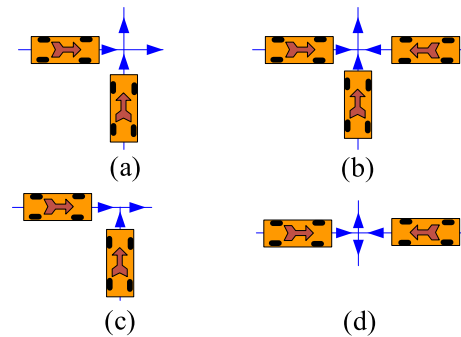


FIGURE 12. Typical intersection. (a) Double-in and double-out intersection; (b) three-in and single-out intersection; (c) double-in and single-out intersection; (d) double-in and double-out intersection.

a comprehensive bidding model is used to calculate the comprehensive bidding price of each vehicle in the semaphore waiting list $\Omega_{B_w}(v_i)$. The procedure shown in Algorithm 3 is used to determine a vehicle with the highest priority right of entering the intersection. First, the vehicle with the highest comprehensive bidding price is given the highest priority right of entering the intersection, and then, the presented collision and deadlock avoidance strategies are used to coordinate the passing order of vehicles at the intersection. The comprehensive bidding model is defined as follows:

$$U_C (AGV_k) = \omega_T U_T (AGV_k) + (1 - \omega_T) U_S (AGV_k), \quad (10)$$

where ω_T represents the weight of $U_T (AGV_k)$.

V. SIMULATION RESULTS ANALYSIS

Extensive simulation tests were conducted to verify the performance of the proposed methods and strategies. In this section, first, the simulation settings and performance indexes of both the processing subsystem and the logistics transportation subsystem are given. Then, several simulations are presented to demonstrate the effectiveness of the proposed collision and deadlock avoidance methods, and the performance of the proposed methods is compared with that of the ZC method. Finally, the performance of the proposed traffic sequence optimization strategy is analyzed at different numbers of vehicles.

A. SIMULATION SETTINGS

The manufacturing system with a UGN-based AGVS, which is shown in Fig. 1, was used in the experiments. The capacities of the input and output buffers of each processing workstation were ten and five, respectively. In order to avoid the dispatching deadlock [43] caused by insufficient buffer capacity of the processing subsystem, the manufacturing system was modeled as a closed system [44], and the number of workpieces in the manufacturing system was fixed and set to 60. The preliminary experiments showed that this number was large enough to keep most of the processing machines busy. The operation sequence of the workpiece set is given in Table 3.

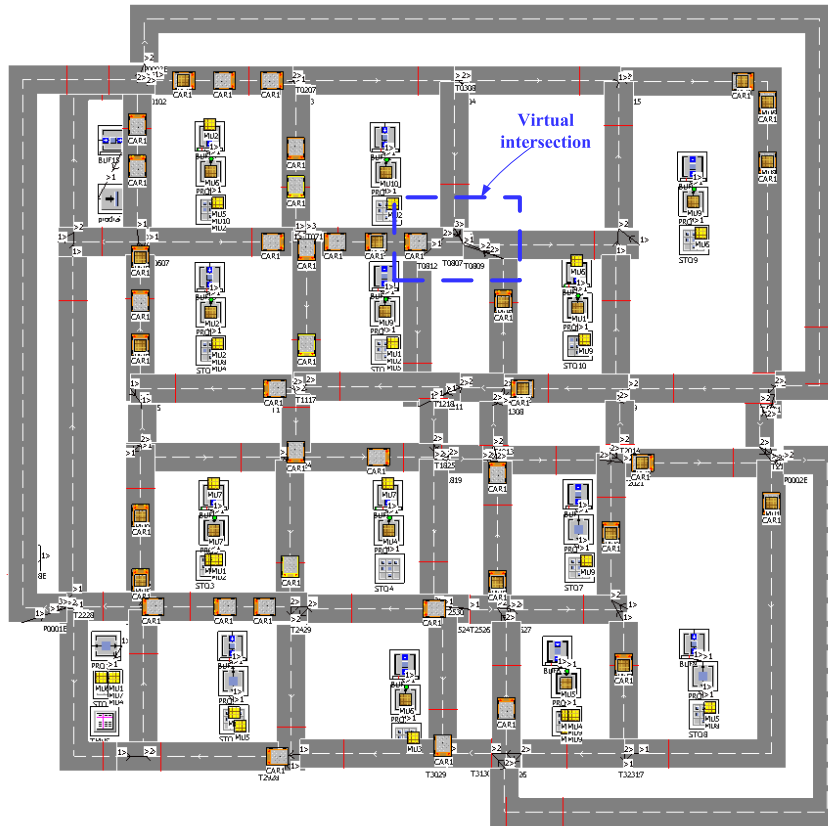


FIGURE 13. The simulation interface.

Algorithm 3 The Bidding Mechanism-Based Traffic Sequence Optimization Procedure

- 01: **Input:** Semaphore waiting list $\Omega_{Bw}(v_i)$.
- 02: **Output:** A vehicle with the highest priority right of entering the intersection.
- 03: **Begin**
- 04: **If** $\Omega_{Bw}(v_i) \neq \emptyset$ **then**
- 05: **For** $k = 1$ to $N^{\Omega_{Bw}}(v_i)$ **loop**
- 06: Calculate the comprehensive bidding price of vehicle $AGV_k^{\Omega_{Bw}}(v_i)$
- 07: Update the vehicle with the highest comprehensive bidding price.
- 08: **Next** k
- 09: **End if**
- 10: Determine the vehicle with the highest comprehensive bidding price, assuming the vehicle is AGV_k , and the applied path segment by AGV_k is $e_{i,j}$.
- 11: Use Algorithm 1 to determine whether a cycle deadlock will be caused if AGV_k is allowed to enter $e_{i,j}$.
- 12: **If** a cycle deadlock is found by Algorithm 1 **then**
- 13: Use the proposed deadlock avoidance strategies to avoid the cycle deadlock.
- 14: **End if**
- 15: Return (AGV_k).
- 16: **End**

TABLE 3. The operation sequence of the workpiece set.

Workpiece type	Operation sequence											Mix ratio (%)				
A	1	2	12	8	4	14	13	10	7	6	3	5	9	11	15	10
B	1	3	13	9	5	4	14	11	8	7	2	6	10	12	15	10
C	1	4	14	10	6	5	2	12	9	8	3	7	11	13	15	10
D	1	5	2	11	7	6	3	13	10	9	4	8	12	14	15	10
E	1	6	2	13	8	7	4	14	11	10	3	5	9	12	15	10
F	1	7	3	13	9	8	5	2	12	11	4	6	10	14	15	10
G	1	8	4	14	10	9	6	3	13	12	2	5	11	7	15	10
H	1	9	5	3	11	10	7	4	14	13	2	6	8	12	15	10
I	1	10	6	2	12	11	8	5	4	14	3	7	9	13	15	10
J	1	11	7	3	13	12	9	6	5	2	4	8	10	14	15	10

The processing time of each workpiece in each workstation was 30 s. The vehicle length was 2 m, and the minimum safe distance between vehicles was 1 m. The vehicle acceleration was set to 0.4 m/s². During the simulation, when a vehicle became idle, the central controller assigned a handing task with the longest waiting time to it. The software Tecnomatix Plant Simulation 15.0 [45] was utilized in the simulation. The simulation interface is shown in Fig. 13. The simulation warm-up time was 48 h, and the simulation time was 120 h.

B. PERFORMANCE METRICS

In the simulations, five performance metrics were used to evaluate the effectiveness of the proposed collision and deadlock prevention methods:

- Average throughput: the average number of workpieces completed by the manufacturing system per hour.
- Average traveling time per handling task \bar{t} : it was calculated by: $\bar{t} = \frac{1}{N_T} \sum_{n=1}^{N_T} (t_E^n - t_S^n)$, where t_E^n denoted the ending time of a workpiece handling task T_{ij}^n , and t_S^n denoted the starting time of a workpiece handling task T_{ij}^n ; N_T represented the number of workpiece handling tasks completed by the AGVS.
- Average traveling distance per handling task $\bar{\Psi}$: it was calculated by: $\bar{\Psi} = \frac{1}{N_T} \sum_{n=1}^{N_T} \Psi_n$, where Ψ_n denoted the distance traveled by a vehicle during the carrying of a workpiece handling task T_{ij}^n , and N_T denoted the number of tasks completed by the AGVS.
- Frequency of deadlock critical state: the average number of deadlock critical states per hour.
- Frequency of multi-cycle deadlock critical state: the average number of multi-cycle deadlock critical states per hour.

The first metric reflected the efficiency of the processing subsystem, and the other four metrics reflected the efficiency of the logistics transportation subsystem.

C. SIMULATION OF PROPOSED COLLISION AND DEADLOCK AVOIDANCE METHOD

The effectiveness of the proposed collision and deadlock avoidance methods was verified by simulations, and the performance of the proposed methods was compared with the performance of the ZC method [32]. During the simulation, the strategy proposed by Moorthy *et al.* [41] was adopted to avoid the cycle deadlocks of the ZC AGVS. Since the zone size has a great influence on the ZC method performance, the performance of the ZC method was analyzed using two zone sizes: large zone size and small zone size. The parameters of the methods were set as follows.

- The proposed collision and deadlock avoidance method (CDAM): As shown in Fig. 13, the UGN had 60 path segments and 35 intersections, including one virtual intersection. Thus, a total of 60 integer semaphores and 35 mutually exclusive semaphores were defined. According to the vehicle length and safety distance, all path segments had a total capacity of 224, of which three parking path segments have a total capacity of 58.
- The ZC method under a large zone size (ZCML): Each path segment or intersection was defined as a zone, i.e., each path segment or intersection could be occupied by only one vehicle; so, 95 zones were defined.
- The ZC method under a small zone size (ZCMS): Each intersection was defined as a zone, and each path segment was defined as a multi-zone segment according to its capacity; so, 259 zones were defined. It should be noted that the ZC method under a small zone size was similar to the collision prevention method proposed by Zhao *et al.* [6], who reduced the zones to small squares and regarded the squares as resource points.

In the simulation, all five performance metrics were used to evaluate and compare the performances of the methods. The

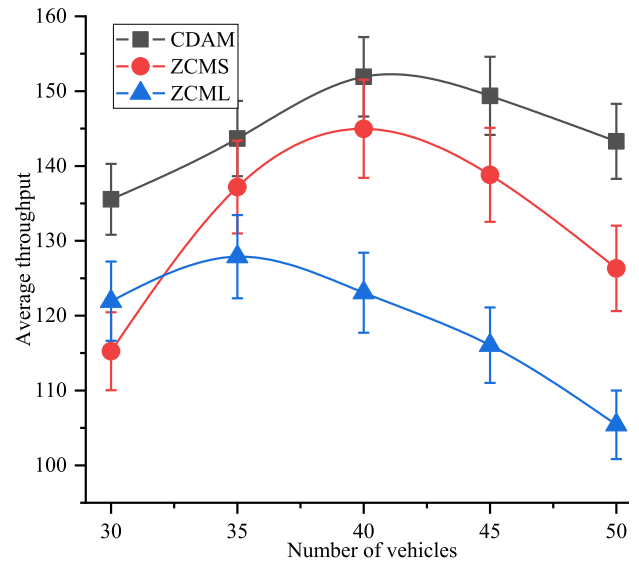


FIGURE 14. Average throughput per hour of different methods at different numbers of vehicles.

vehicle speed was 1 m/s, and the rest of the parameters were the same as described in Section V.A. The number of vehicles varied from 30 to 50 with a step of 5. The results of the five metrics are shown in Figs. 14–18.

As shown in Fig. 14, when the number of vehicles was smaller than 40, the average throughput of the CDAM increased with the vehicle number. The best average throughput of the CDAM was achieved for 40 vehicles. However, a further increase in the number of vehicles reduced method performance. Particularly, when the number of vehicles was larger than 45, the performance of the CDAM declined sharply. The changing trends of the ZCMS and ZCML methods were similar to that of the CDAM. Obviously, the CDAM achieved the best average throughput at different numbers of vehicles; the average throughput of the CDAM was 3.5%–27.4% higher than those of the ZCMS and ZCML methods. Consequently, the CDAM is more efficient than the ZC methods for the processing subsystem.

As displayed in Fig. 15, the average traveling time of all the methods increased with the number of vehicles. Particularly, when the number of vehicles was larger than 45, the average traveling time of all the methods increased significantly. The reason for this phenomenon was that with the increase in the number of vehicles, the AGVS became more congested, and the frequency of deadlock critical state increased significantly, as shown in Figs. 16 and 17, thus prolonging the task traveling time. According to the results in Fig. 15, the average traveling time of the CDAM was 6.2%–29.6% shorter than those of the ZCMS and ZCML methods at different numbers of vehicles. As shown in Fig. 16, the deadlock critical state frequency of the CDAM was 15.1%–39.8% smaller than those of the ZCMS and ZCML methods when the number of vehicles was larger than 40. Consequently, the CDAM can reduce the risks of congestion and deadlock, while improving the traffic status of the logistics transportation subsystem.

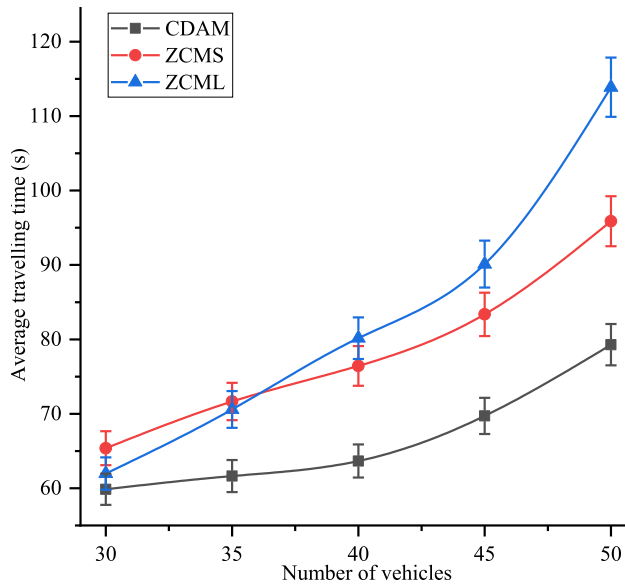


FIGURE 15. Average traveling time per task of different methods at different numbers of vehicles.

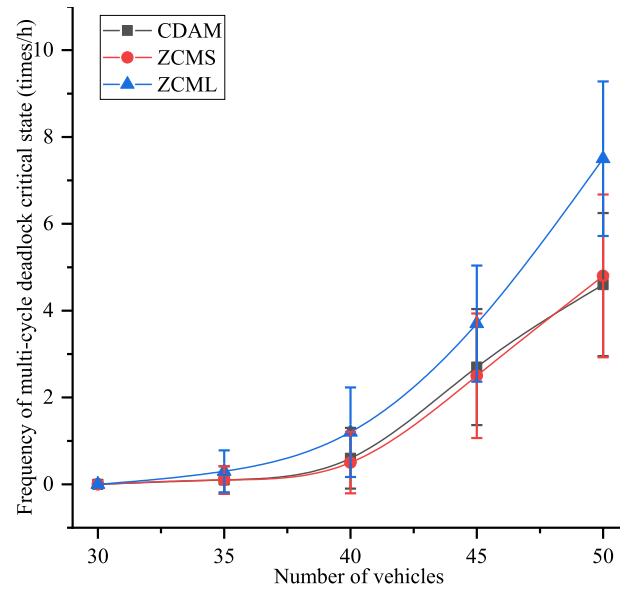


FIGURE 17. Frequency of multi-cycle deadlock critical state of different methods at different numbers of vehicles.

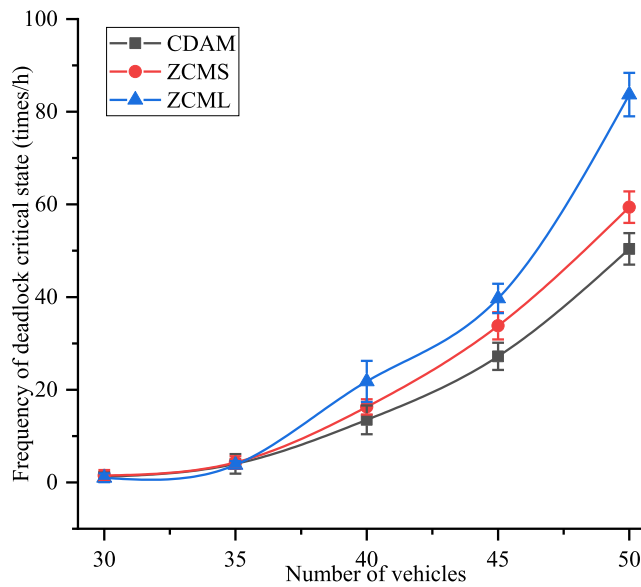


FIGURE 16. Frequency of deadlock critical state of different methods at different numbers of vehicles.

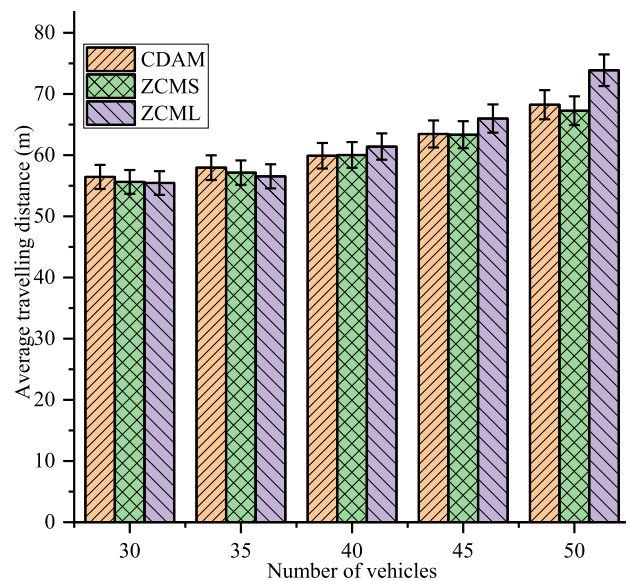


FIGURE 18. Average traveling distance per task of different methods at different numbers of vehicles.

As shown in Fig. 16, when the number of vehicles was 35, the cycle deadlock critical state was detected. With the further increase in the number of vehicles, the frequency of the deadlock critical state increased significantly. By using the avoidance strategy proposed in this paper, the cycle deadlock was avoided, which proved the effectiveness of the deadlock avoidance strategy.

In Fig. 17, it can be seen that the frequency of the multi-cycle deadlock critical state was relatively low. Even when the number of vehicles was 50, the frequency of the multi-cycle deadlock critical state was still less than 8 times/h, which indicated that the probability of re-planning the vehicle path was very low. Therefore, as shown in Fig. 18,

as the number of vehicles increased, the average traveling distance per task slightly increased.

Figs. 14 and 15 show that the average traveling time of the ZCML method was shorter than that of the ZCMS method when the number of vehicles was smaller than 35. However, when the number of vehicles was larger than 35; the situation was the opposite. Therefore, in order to improve the efficiency of the ZC method, the zone size needs to be designed reasonably according to the number of vehicles in the system. Since the ZCMS defined each intersection as a zone and used smaller and more zones, each path loop of ZCMS could accommodate more vehicles than the CDAM and the ZCML. As shown in Figs. 17 and 18, the ZCMS could achieved

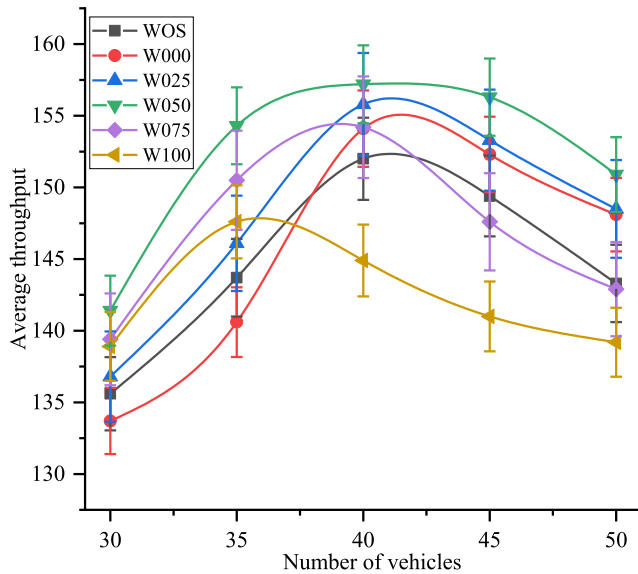


FIGURE 19. Average throughput per hour at different numbers of vehicles.

comparable performance to the CDAM or even outperform it in terms of frequency of multi-cycle deadlock critical states and average traveling distance per task. However, the shortest distance did not guarantee the shortest time. The waiting time was also an important part of the traveling task time. Therefore, the average traveling time per handling task was a more important metric, which directly reflected the performance of the handling system. Since vehicles needed to pause and wait for permission from the central controller before entering another zone, using smaller and more zones increased the number of vehicles pauses, thus prolonging the traveling times of vehicles. As shown in Fig. 15, the average traveling time of the ZCMS was longer than that of the CDAM. In summary, as shown in Figs. 14 and 15, the CDAM achieved the best results among the ZC methods at all numbers of vehicles, so it can be concluded that the CDAM is more suitable for the UGN-based AGVS than the ZC methods.

D. SIMULATION OF PROPOSED TRAFFIC SEQUENCE OPTIMIZATION STRATEGY

In order to evaluate the proposed traffic sequence optimization strategy further, the performance of the optimization strategy was analyzed for different weights, and the results were compared to those obtained without using any optimization strategy. The obtained performances are shown in Figs. 19–23, where the WOS represents the CDAM without any traffic optimization strategy, and the W000 represents the CDAM with the proposed optimization strategy when $\omega_T = 000/100 = 0$; the W025 represents the CDAM with the optimization strategy when $\omega_T = 0.25$; the W050 represents the CDAM with the optimization strategy when $\omega_T = 0.50$; the W075 represents the CDAM with the optimization strategy when $\omega_T = 0.75$; lastly, the W100 represents the CDAM with the optimization strategy when $\omega_T = 1.00$.

As shown in Fig. 19, each optimization strategy had a different optimal number of vehicles. For instance, the average

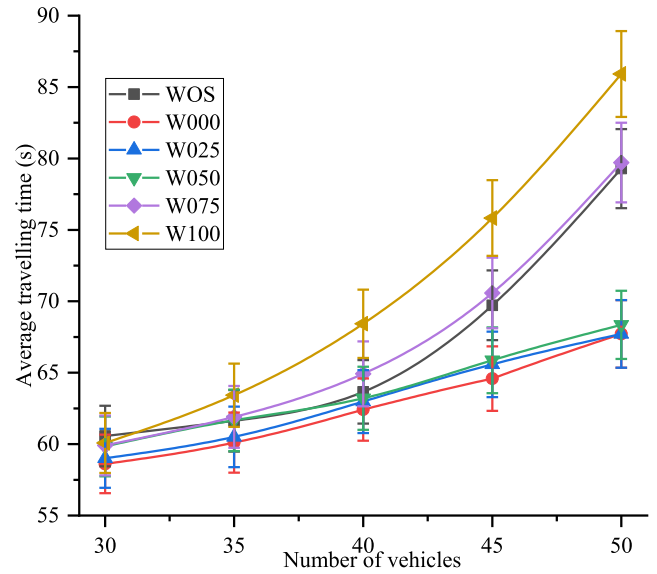


FIGURE 20. Average traveling time per task at different numbers of vehicles.

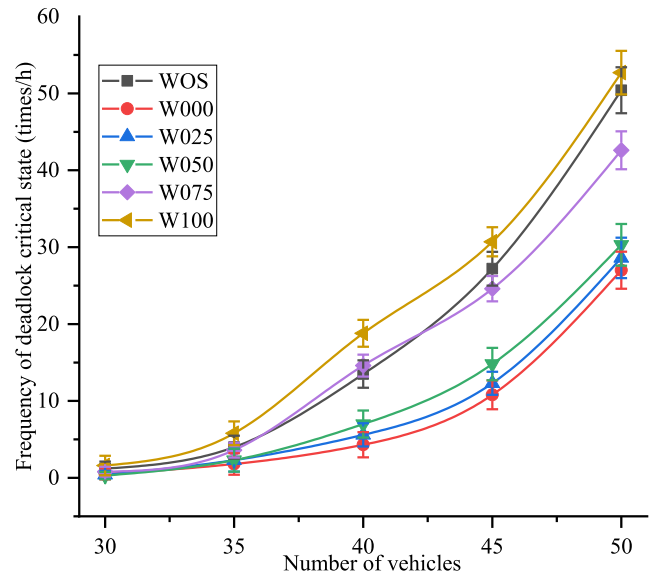


FIGURE 21. Frequency of the deadlock critical state at different numbers of vehicles.

throughput of the W050 was the largest when the number of vehicles was 40, and further increasing or decreasing the number of vehicles reduced the average throughput of the system. This can be explained as follows. On the one hand, when the number of vehicles was small, the processing machine of each processing workstation could easily block or starve due to insufficient transportation capacity of the handling subsystem. On the other hand, when there were too many vehicles in the system, the probability of system traffic congestion and deadlock increased significantly, as shown in Figs. 21 and 22. Thus, the average traveling time per task was prolonged, and the system throughput was reduced.

The simulation results showed that the weight had a great impact on the performance of the optimization strategy. Since

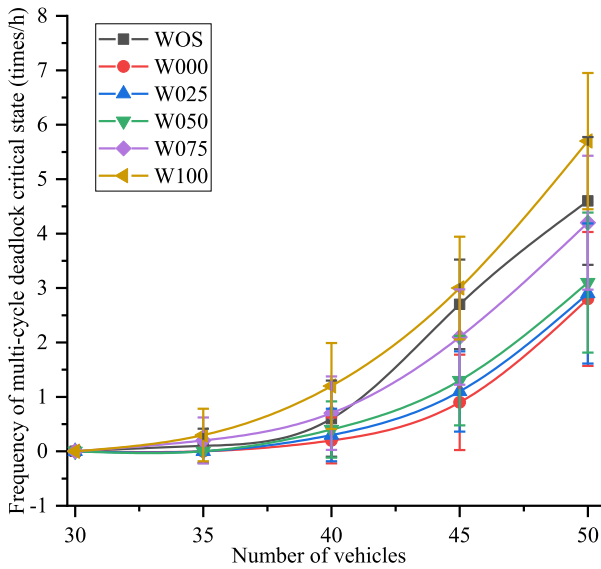


FIGURE 22. Frequency of the multi-cycle deadlock critical state at different numbers of vehicles.

the W100 optimized the traffic sequence only based on the task urgency of vehicles, the vehicle carrying out urgent tasks had the priority over the intersection, so the risk of machine blocking or idling at each workstation was reduced. As shown in Fig. 19, when the number of vehicles was smaller than 35, the average throughput of the W100 was better than those of W000, W025, and WOS. However, with the increase in the number of vehicles, the probability of system traffic congestion and deadlock increased, as shown in Figs. 21 and 22. Since the traffic urgency of the vehicle was not considered, the performances of the W100 decreased, as shown in Figs. 19-23.

Namely, the W000 optimizes the traffic sequence only based on the traffic urgency of vehicles. Accordingly, if the remaining capacity of a path segment where a vehicle is located is small, and the remaining capacity of the next path segment is large, the vehicle will pass the intersection, thus reducing the risk of congestion and deadlock. As shown in Figs. 20–23, the W000 achieved the best performance among all the methods in terms of the performance metrics reflecting the logistics transportation subsystem. As displayed in Fig. 19, when the number of vehicles was larger than 40, the average throughput of the W000 was better than those of W075, W100, and WOS. However, when the number of vehicles was less than 35, the average throughput of the W000 was the worst among all the methods because the transport capacity of the logistics transportation subsystem was insufficient, and W000 did not consider the task urgency of vehicles. Therefore, in order to ensure the efficiency of both the processing subsystem and the logistics transportation subsystem at the same time, both the task urgency and the traffic urgency of vehicles must be considered when optimizing the traffic sequence.

Since both the vehicle’s task urgency and the traffic urgency were considered, the average throughput of the

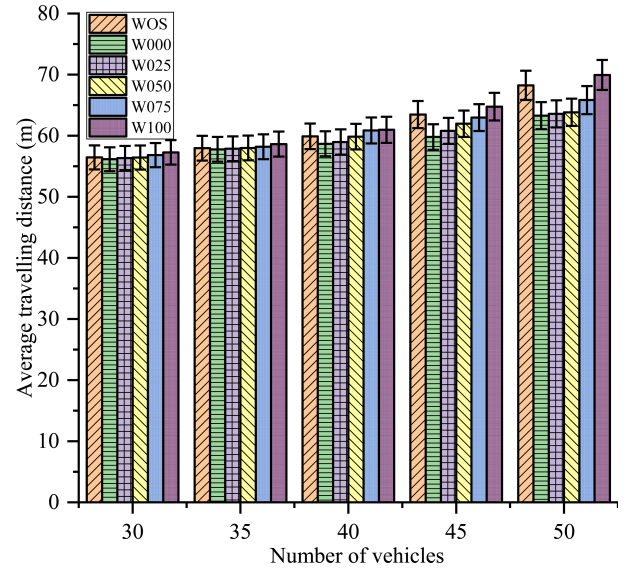


FIGURE 23. Average traveling distance per task at different numbers of vehicles.

W025 was better than that of the W000, and the average throughput of the W075 was better than that of the W100, as shown in Fig. 19. However, the performances of the W025 and W075 were not stable due to the unbalanced weight distribution. As shown in Fig. 19, the average throughput of the W075 was better than that of the W025, when the number of vehicles was up to 35. On the contrary, when the number of vehicles was larger than 35, the average throughput of the W025 was better than that of the W075. Since the W050 had equal weights on task urgency and traffic urgency, it could achieve the best average throughput among all the methods at different numbers of vehicles. In addition, the average throughput of the W050 was 3.5%–5.3% better than that of the WOS. Based on the obtained results, the W050 can improve the efficiency of the processing subsystem.

The results of the transportation subsystem efficiency metrics of the optimization strategies are shown in Figs. 20–23. The weight of the vehicle’s task urgency was denoted as ω_T , and the smaller ω_T was, the larger the weight of the vehicle’s traffic urgency was, and the better the performance of metrics related to the logistics transportation subsystem was. As displayed in Figs. 20–23, the average travel time and distance of the W000 were the smallest (refer to Figs. 20 and 23), and the frequency of deadlock critical state was the lowest (refer to Figs. 21 and 22) among all the methods; in contrast, the average time and distance, and the frequency of the deadlock critical state of the W100 were the largest. Although the performances of the W025, W050, and W075 were not optimal, they were very close to that of the W000. As shown in Fig. 21, the deadlock critical state frequency of the W050 was 39.8%–47.9% smaller than that of the WOS. By comparing Fig. 22 with Fig. 17, it could be seen that the frequency of multi-cycle deadlock critical states of the W050 was lower than that of the ZCMS at all numbers of vehicles. And the average traveling distance per

task of the W050 was shorter than that of the ZCMS (refer to Figs. 23 and 18). The results show that it is necessary to optimize the intersection traffic sequence, and the W050 can reduce the congestion and deadlock risk of the AGVS, thus improving the traffic status of the logistics transportation subsystem.

As displayed in Figs. 19 and 20, the W050 had the highest average throughput but not the shortest traveling time per handling task among all the methods. This can be explained as follows. The average throughput of the processing subsystem mainly depended on the utilization rate of machine tools in each workstation. The idle or blocked state of each machine tool would reduce its utilization rate. In order to reduce the risk of idle or blocked machine tools in each workstation, it was necessary to deliver each workpiece to the corresponding input buffer of machine tools at an appropriate time, rather than only simply reducing the average traveling time of all workpieces. The W000 optimized the traffic sequence only based on the traffic urgency of vehicles. It achieved the shortest traveling time per handling task. Since the task urgency of vehicles was not considered, the risk of machine blocking or idling in each workstation was increased, and the average throughput of the W000 was reduced. Unlike the W000, the W050 considered both the vehicle's task urgency and the traffic urgency. Compared to the W000, the W050 increased the risk of traffic congestion and deadlock, thus prolonging the task execution time, but it could reduce the risk of idle or blocked machine tools in each workstation, thus improving the throughput of the system. As shown in Fig. 20, although the average traveling time per task obtained by the W050 was longer than that of the W000, it was very close to it. The simulation results show that the W050 not only can improve the average throughput of the processing subsystem but also ensure the efficiency of the logistics and transportation subsystem.

VI. CONCLUSION

In this paper, a collision and deadlock prevention method with a traffic sequence optimization strategy for the UGN-based AGVS is proposed. First, a vehicle coordination mechanism based on the semaphore theory and IoT positioning technology, which can avoid the collisions between vehicles and eliminate the intersection congestion deadlocks, is proposed. Next, in order to avoid the cycle deadlocks, a search and avoidance algorithm based on the digraph theory is developed. Then, a bidding mechanism-based strategy is proposed to optimize the vehicle traffic sequence in each path intersection. Finally, extensive simulation tests are performed to verify the performance of the proposed methods and strategy. Simulation results show that, compared with the ZC methods, the average travel time of the proposed method is reduced by 6.2%–29.6%, and the average throughput is increased by 3.5%–27.4%. Thus, the proposed collision and deadlock prevention method is more suitable for the UGN-based AGVS than the ZC methods. Moreover, the proposed traffic sequence optimization strategy increases the average

throughput of the processing subsystem by 3.5%–5.3% and reduces the deadlock critical state frequency of the transportation subsystem by 39.8%–47.9%, which proves the effectiveness of the proposed traffic sequence optimization strategy.

Besides, the simulation results show that the number of vehicles in a system has a great impact on system performance. Therefore, how to determine an optimal number of vehicles in a system will be studied in our future work. In addition, the traffic sequence optimization strategy proposed in this paper is applicable only to a single-load AGVS. However, in the logistics transportation field, the multi-load AGVS has become popular recently, due to the advantages of small fleet size and high system throughput. Therefore, our future work will also include the development of a traffic sequence optimization strategy for the multi-load AGVS.

REFERENCES

- [1] W. Hellmann, D. Marino, M. Megahed, M. Suggs, J. Borowski, and A. Negahban, "Human, AGV or AIV? An integrated framework for material handling system selection with real-world application in an injection molding facility," *Int. J. Adv. Manuf. Technol.*, vol. 101, nos. 1–4, pp. 815–824, Mar. 2019.
- [2] V. K. Chawla, A. K. Chanda, S. Angra, and S. Rani, "Effect of nature-inspired algorithms and hybrid dispatching rules on the performance of automatic guided vehicles in the flexible manufacturing system," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 41, no. 10, p. 391, Oct. 2019.
- [3] M. Zhong, Y. Yang, Y. Dessouky, and O. Postolache, "Multi-AGV scheduling for conflict-free path planning in automated container terminals," *Comput. Ind. Eng.*, vol. 142, Apr. 2020, Art. no. 106371.
- [4] U. Orozco-Rosas, K. Picos, and O. Montiel, "Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots," *IEEE Access*, vol. 7, pp. 156787–156803, 2019.
- [5] B. Yang, W. Li, J. Wang, J. Yang, T. Wang, and X. Liu, "A novel Path planning algorithm for warehouse robots based on a two-dimensional grid model," *IEEE Access*, vol. 8, pp. 80347–80357, 2020.
- [6] Y. Zhao, X. Liu, G. Wang, S. Wu, and S. Han, "Dynamic resource reservation based collision and deadlock prevention for multi-AGVs," *IEEE Access*, vol. 8, pp. 82120–82130, 2020.
- [7] D. Deplano, M. Franceschelli, S. Ware, S. Rong, and A. Giua, "A discrete event formulation for multi-robot collision avoidance on pre-planned trajectories," *IEEE Access*, vol. 8, pp. 92637–92646, 2020.
- [8] P. Reddy Gutta, V. Sai Chinthala, R. Venkatesh Manchoju, V. Charan MVN, and R. Purohit, "A review on facility layout design of an automated guided vehicle in flexible manufacturing system," *Mater. Today, Proc.*, vol. 5, no. 2, pp. 3981–3986, 2018.
- [9] A. Asef-Vaziri and M. Kazemi, "Covering and connectivity constraints in loop-based formulation of material flow network design in facility layout," *Eur. J. Oper. Res.*, vol. 264, no. 3, pp. 1033–1044, Feb. 2018.
- [10] J. A. Ventura, S. Pazhani, and A. Mendoza, "Finding optimal dwell points for automated guided vehicles in general guide-path layouts," *Int. J. Prod. Econ.*, vol. 170, pp. 850–861, Dec. 2015.
- [11] B. Rahimikelarjani, M. Saidimehrabad, and F. J. Barzinpour, "A mathematical model for multiple-load AGVs in tandem layout," *J. Optim. Ind. Eng.*, vol. 13, no. 1, pp. 67–80, 2020.
- [12] A. Ahmadi-Javid and N. Ramshe, "Designing flexible loop-based material handling AGV paths with cell-adjacency priorities: An efficient cutting-plane algorithm," *4OR*, vol. 17, no. 4, pp. 373–400, Dec. 2019.
- [13] I. F. A. Vis, "Survey of research in the design and control of automated guided vehicle systems," *Eur. J. Oper. Res.*, vol. 170, no. 3, pp. 677–709, May 2006.
- [14] W. Xing, L. Peihuang, Y. Jun, Q. Xiaoming, and T. Dunbing, "Intersection recognition and guide-path selection for a vision-based AGV in a bidirectional flow network," *Int. J. Adv. Robot. Syst.*, vol. 11, no. 3, p. 39, Mar. 2014.
- [15] X. Guan, X. Dai, and J. Li, "Revised electromagnetism-like mechanism for flow path design of unidirectional AGV systems," *Int. J. Prod. Res.*, vol. 49, no. 2, pp. 401–429, Jan. 2011.

- [16] K. Murakami, "Time-space network model and MILP formulation of the conflict-free routing problem of a capacitated AGV system," *Comput. Ind. Eng.*, vol. 141, Mar. 2020, Art. no. 106270.
- [17] H. Fazlollahabadi and S. Hassanli, "Hybrid cost and time path planning for multiple autonomous guided vehicles," *Int. J. Speech Technol.*, vol. 48, no. 2, pp. 482–498, Feb. 2018.
- [18] T. Miyamoto and K. Inoue, "Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems," *Comput. Ind. Eng.*, vol. 91, pp. 1–9, Jan. 2016.
- [19] E. Xidias, P. Zacharia, and A. Nearchou, "Path planning and scheduling for a fleet of autonomous vehicles," *Robotica*, vol. 34, no. 10, pp. 2257–2273, Oct. 2016.
- [20] X. Lyu, Y. Song, C. He, Q. Lei, and W. Guo, "Approach to integrated scheduling problems considering optimal number of automated guided vehicles and conflict-free routing in flexible manufacturing systems," *IEEE Access*, vol. 7, pp. 74909–74924, 2019.
- [21] Y. Yang, M. Zhong, Y. Dessouky, and O. Postolache, "An integrated scheduling method for AGV routing in automated container terminals," *Comput. Ind. Eng.*, vol. 126, pp. 482–493, Dec. 2018.
- [22] T. Nishi, Y. Hiranaka, and I. E. Grossmann, "A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles," *Comput. Oper. Res.*, vol. 38, no. 5, pp. 876–888, May 2011.
- [23] T. Nishi and Y. Tanaka, "Petri net decomposition approach for dispatching and conflict-free routing of bidirectional automated guided vehicle systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 5, pp. 1230–1243, Sep. 2012.
- [24] H. Xiao, X. Wu, Y. Zeng, and J. Zhai, "A CEGA-based optimization approach for integrated designing of a unidirectional guide-path network and scheduling of AGVs," *Math. Problems Eng.*, vol. 2020, pp. 1–16, Feb. 2020.
- [25] T. Adamo, T. Bektaş, G. Ghiani, E. Guerriero, and E. Manni, "Path and speed optimization for conflict-free pickup and delivery under time windows," *Transp. Sci.*, vol. 52, no. 4, pp. 739–755, Aug. 2018.
- [26] L. Z. Du, S. F. Ke, Z. Wang, J. Tao, L. Q. Yu, and H. J. Li, "Research on multi-load AGV path planning of weaving workshop based on time priority," *Math. Biosci. Eng.*, vol. 16, no. 4, pp. 2277–2292, 2019.
- [27] J.-J. Li, B.-W. Xu, O. Postolache, Y.-S. Yang, and H.-F. Wu, "Impact analysis of travel time uncertainty on AGV catch-up conflict and the associated dynamic adjustment," *Math. Problems Eng.*, vol. 2018, pp. 1–11, Jan. 2018.
- [28] R. Tai, J. Wang, and W. Chen, "A prioritized planning algorithm of trajectory coordination based on time windows for multiple AGVs with delay disturbance," *Assem. Autom.*, vol. 39, no. 5, pp. 753–768, Nov. 2019.
- [29] S. G. Kumbhar, R. B. Thombare, and A. B. Salunkhe, "Automated guided vehicles for small manufacturing enterprises: A review," *SAE Int. J. Mater. Manuf.*, vol. 11, no. 3, pp. 253–258, Sep. 2018.
- [30] A. Sedeño-noda and M. Colebrook, "A biobjective dijkstra algorithm," *Eur. J. Oper. Res.*, vol. 276, no. 1, pp. 106–118, Jul. 2019.
- [31] Y. Zhang, L. L. Li, H. C. Lin, Z. W. Ma, and J. Zhao, "Development of Path planning approach using improved a-star algorithm in AGV system," *J. Internet Technol.*, vol. 20, no. 3, pp. 915–924, 2019.
- [32] S. Reveliotis, "An MPC scheme for traffic coordination in open and irreversible, zone-controlled, guidepath-based transport systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 3, pp. 1528–1542, Jul. 2020.
- [33] Y.-C. Ho and T.-W. Liao, "Zone design and control for vehicle collision prevention and load balancing in a zone control AGV system," *Comput. Ind. Eng.*, vol. 56, no. 1, pp. 417–432, Feb. 2009.
- [34] J. Zhang, X. Wang, and K. Huang, "On-line scheduling of order picking and delivery with multiple zones and limited vehicle capacity," *Omega*, vol. 79, pp. 104–115, Sep. 2018.
- [35] M. P. Fanti, "Event-based controller to avoid deadlock and collisions in zone-control AGVS," *Int. J. Prod. Res.*, vol. 40, no. 6, pp. 1453–1478, Jan. 2002.
- [36] F. Pan and Q. Sun, "A traffic control strategy of the heavy-duty AGVS in a square topology," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2019, pp. 263–268.
- [37] E. Roszkowska and S. A. Reveliotis, "On the liveness of guidepath-based, zone-controlled dynamically routed, closed traffic systems," *IEEE Trans. Autom. Control*, vol. 53, no. 7, pp. 1689–1695, Aug. 2008.
- [38] S. Reveliotis and T. Masopust, "Efficient liveness assessment for traffic states in open, irreversible, dynamically routed, zone-controlled guidepath-based transport systems," *IEEE Trans. Autom. Control*, vol. 65, no. 7, pp. 2883–2898, Jul. 2020.
- [39] W. Małopolski, "A sustainable and conflict-free operation of AGVs in a square topology," *Comput. Ind. Eng.*, vol. 126, pp. 472–481, Dec. 2018.
- [40] S. Shao, Z. Xia, G. Chen, J. Zhang, Y. Hu, and J. Zhang, "A new scheme of multiple automated guided vehicle system for collision and deadlock free," in *Proc. 4th IEEE Int. Conf. Inf. Sci. Technol.*, Apr. 2014, pp. 606–610.
- [41] R. L. Moorthy, W. Hock-Guan, N. Wing-Cheong, and T. Chung-Piaw, "Cyclic deadlock prediction and avoidance for zone-controlled AGV system," *Int. J. Prod. Econ.*, vol. 83, no. 3, pp. 309–324, Mar. 2003.
- [42] N. Wu, M. Zhou, and G. Hu, "One-step look-ahead maximally permissive deadlock control of AMS by using Petri nets," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 1, pp. 1–23, Jan. 2013.
- [43] X. Guan and X. Dai, "Deadlock-free multi-attribute dispatching method for AGV systems," *Int. J. Adv. Manuf. Technol.*, vol. 45, nos. 5–6, pp. 603–615, Nov. 2009.
- [44] Y.-C. Ho, H.-C. Liu, and Y. Yih, "A multiple-attribute method for concurrently solving the pickup-dispatching problem and the load-selection problem of multiple-load AGVs," *J. Manuf. Syst.*, vol. 31, no. 3, pp. 288–300, Jul. 2012.
- [45] P. Trebuna, M. Pekarcikova, and M. Edl, "Digital value stream mapping using the tecnomatix plant simulation software," *Int. J. Simul. Model.*, vol. 18, no. 1, pp. 19–32, Mar. 2019.



HAINING XIAO received the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014. He is currently a Lecturer with the Yancheng Institute of Technology. His research interests include coordinated control and intelligent scheduling of multi-AGV systems.



XING WU (Member, IEEE) received the B.Eng. and Ph.D. degrees in mechanical engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2004 and 2010, respectively. In 2010, he joined the College of Mechanical and Electrical Engineering, NUAA, where he is currently an Associate Professor. He was a Visiting Professor with the Centre for Intelligent Machines, McGill University, Canada, from November 2016 to November 2017.

His research interests include design, guidance, navigation, and control of mobile robots; and coordinated control and intelligent scheduling of multi-robot systems. He has authored or coauthored about 40 journal articles and conference papers, and has obtained more than 20 issued China patents. He received the second Prize for Progress in Science and Technology, Ministry of Education of China, in 2013 and 2017, respectively.



DEJIN QIN received the B.S. degree in mechanical engineering from Nanhang Jincheng College, Jiangsu, China, in 2016. He is currently pursuing the M.Sc. degree in mechanical engineering with the Yancheng Institute of Technology, China. His research interests include robots path planning and logistics simulation.



JINGJING ZHAI received the B.S. degree in mechanical engineering from the Anhui University of Science and Technology, Anhui, China, in 2018. She is currently pursuing the M.Sc. degree in mechatronic engineering with the Nanjing University of Aeronautics and Astronautics, China. Her current research interests include intelligent dispatching and traffic management of AGVS.

• • •