Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A novel motion coordination method for variable-sized multi-mobile robots*

Zichao XING†, Xingkai WANG, Shuo WANG, Weimin WU†‡, Ruifen HU

State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China

†E-mail: zcxing@zju.edu.cn; wmwu@iipc.zju.edu.cn

Received Apr. 19, 2022; Revision accepted Dec. 31, 2022; Crosschecked Feb. 8, 2023

Abstract: Multi-mobile robot systems (MMRSs) are widely used for transportation in industrial scenes such as manufacturing and warehousing. In an MMRS, motion coordination is important as collisions and deadlocks may lead to losses or system stagnation. However, in some scenarios, robot sizes are different when loaded and unloaded, which means that the robots are variable-sized, making motion coordination more difficult. The methods based on zone control need to first divide the environment into disjoint zones, and then allocate the zones statically or dynamically for motion coordination. The zone-control-based methods are not accurate enough for variable-sized multi-mobile robots and reduce the efficiency of the system. This paper describes a motion coordination method based on glued nodes, which can dynamically avoid collisions and deadlocks according to the roadmap structure and the real-time paths of robots. Dynamic features make this method directly applicable to various scenarios, instead of dividing a roadmap into disjoint zones. The proposed method has been applied to many industrial projects, and this study is based on some manufacturing projects for experiments. Theoretical analysis and experimental results show that the proposed algorithm is effective and efficient.

Key words: Multi-mobile robot system; Collision avoidance; Deadlock avoidance; Glued nodes; Motion coordination

https://doi.org/10.1631/FITEE.2200160 **CLC number:** TP242.6

1 Introduction

Enterprises are paying more and more attention to production efficiency and production safety. As a widely used automated material handling system, the multi-mobile robot system (MMRS) plays an important role in manufacturing, warehousing (Krnjak et al., 2015), container terminal (Zhong et al., 2020), and other scenarios. A stable and efficient MMRS can reduce labor cost and improve production safety. An MMRS involves multiple technologies, such as task allocation, positioning, path planning, and motion coordination (de Ryck et al., 2020). Motion coordination includes collision and deadlock handling, which is an overwhelmingly significant issue in MMRSs.

The roads on which the robots can travel are usually planned in advance in industrial environments. These roads form a roadmap, in which an edge represents a road or part of a road, and a node represents an intersection or a point on a road. Unlike an abstract graph, a roadmap is a map of a real robot environment whose edges can be unidirectional or bidirectional and can be curved or straight. In such a roadmap, the nodes are allocated to robots in a certain way as resources to realize motion coordination among robots. However, the compact layout

‡ Corresponding author

* Project supported by the Key Research and Development Program of Zhejiang Province, China (No. 2023C01174)

ORCID: Zichao XING, https://orcid.org/0009-0009-2061-9699; Weimin WU, https://orcid.org/0000-0002-1958-1920

of some roadmaps results in two nodes not necessarily being collision-free for two robots. In particular, whether two nodes are collision-free for two robots depends on whether the robots are loaded. As shown in Fig. 1a, R_1 and R_2 are not loaded and all nodes are collision-free. However, as can be seen in Fig. 1b, when R_1 and R_2 are loaded, they cannot pass through V_2 and V_5 at the same time, which means that V_2 and V_5 are not collision-free for R_1 and R_2. Compact roadmaps present difficulties for motion coordination of multi-mobile robots.

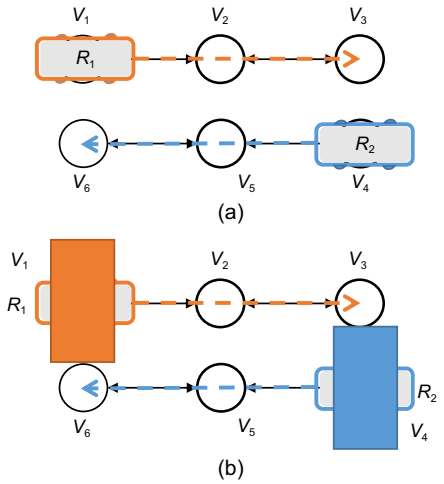

Fig. 1 Examples of the relationship between collision-free nodes and robot loading: (a) all nodes are collision-free for R_1 and R_2; (b) V_2 and V_5 are not collision-free for R_1 and R_2

Motion coordination is focused mainly on collision avoidance and deadlock handling. When it comes to collision avoidance, there are generally two methods: coupling and decoupling methods.

The coupling method statically or dynamically plans collision-free paths or trajectories for robots. There are many methods that have been studied, such as graph theory (Yu and LaValle, 2016), reciprocal collision avoidance (Alonso-Mora et al., 2013), state lattice (Draganjac et al., 2020), and cell decomposition (Zhang et al., 2007).

The other is decoupling that decouples path planning and motion coordination; that is, paths for robots are first planned with a certain goal (such as the shortest distance or shortest task time), and then collisions are avoided based on existing paths. This method is suitable for scenarios where the robot paths are fixed or the roadmaps have been planned, and it is necessary to ensure that only one robot can pass through a physical space at a time. The decoupling method is more suitable for application in industrial scenarios (de Ryck et al., 2020). Wang et al. (2015) set different initial delays for different robots to avoid collisions, but this kind of method is time-sensitive and not robust. Some methods based on zone control (Luo et al., 2020; Reveliotis, 2020; Zhao et al., 2021) divide the environment into disjoint zones, and each zone can accommodate a robot, thus avoiding collisions. Luo et al. (2020) designed an optimal controller to prevent robots from any collision based on Petri nets. They converted the collision-free problem into admissible ones using an algorithm to significantly reduce the computational overhead.

Deadlock handling was originally studied in computer operating systems (Habermann, 1969; Coffman et al., 1971). There are three major approaches for deadlock handling in MMRSs: deadlock detection and resolution, deadlock prevention, and deadlock avoidance (Coffman et al., 1971).

For the method of deadlock detection and resolution, robots are allowed to form deadlocks. Once the system detects a deadlock, the deadlock resolution strategy is applied to unlock the deadlock (Jager and Nebel, 2001; Wu and Zhou, 2007; Alonso-Mora et al., 2018). In Jager and Nebel (2001), the trajectory planner of each involved robot was successively requested to plan an alternative trajectory until the deadlock was resolved. This method is suitable for systems where the number of robots is small and deadlocks are rare.

Deadlock prevention is an offline method that applies some simple rules or designs to the controller before the system release to prevent the occurrence of deadlock (Chen et al., 2016; Xing KY et al., 2018). However, this method suffers from low efficiency and high complexity (Chen et al., 2016).

As an online method, deadlock avoidance has been widely studied. To avoid deadlocks, it is necessary to dynamically determine whether the robot's next move will cause a deadlock and, if so, terminate the action of the robot. Moorthy et al. (2003) developed an efficient deadlock prediction and avoidance algorithm for an automated container port, which has a complex layout and involves close to 80 automated guided vehicles (AGVs). Yoo et al. (2005) presented a deadlock avoidance algorithm that uses the graph-theoretic approach for a robot system in

flexible manufacturing. Fanti et al. (2015) proposed a decentralized coordination protocol based on the zone-control method; deadlocks and collisions can be avoided with a decentralized coordination protocol. Li et al. (2016) presented time-efficient traffic control for MMRSs based on a discrete-event zone-control model. All inter-vehicle collisions and system deadlocks can be avoided. Zhou et al. (2020) modeled robot motion through labeled transition systems and proposed a distributed algorithm to avoid deadlocks in such systems. Each robot has a predetermined and closed path to execute persistent motion. The method is based on fixed path scenarios, where all robots' paths are fixed. Zając and Małopolski (2021) distinguished unidirectional zones and deadlock-risk zones, and introduced a structural online control policy to avoid deadlocks among robots. Małopolski (2018) divided the layout of a transportation system into squares and proposed a method of collision and deadlock avoidance based on chains of reservations.

These collision and deadlock avoidance methods explicitly or implicitly divide the environment into zones, each of which can accommodate only one robot at a specific time. This allows an MMRS to be considered as a discrete event system, and collisions and deadlocks are regarded as resource conflicts. However, the division of zones increases the difficulty of method application, and the static characteristics of zones lead to imprecision and wasted space.

In our previous work (Xing ZC et al., 2022), we proposed the concept of glued nodes, based on which a basic collision and deadlock avoidance algorithm was proposed. Based on the concept of glued nodes, the system can dynamically determine the mutual influence between nodes based on the real-time sizes and paths of the robots. Due to the dynamic features, the glued nodes can be applied to a variety of complex MMRSs. This work improves the definition and properties of glued nodes and provides a more detailed analysis of collision and deadlock avoidance issues. The deadlock is divided into direct deadlock and impending deadlock, and the corresponding collision and deadlock avoidance algorithms are given. In addition, we conduct experiments based on multiple industrial scenarios to verify the performance of the algorithms.

The main contributions of this paper are as follows: (1) A concept of glued nodes is introduced and used to design a collision and deadlock avoidance algorithm; (2) The proposed collision and deadlock avoidance algorithm can be directly applied to an MMRS containing variable-sized robots and a roadmap with various structures; (3) The difference between direct deadlock and impending deadlock is analyzed, and a unified method to avoid them is provided; (4) A hybrid architecture is proposed, where the control center is responsible for allocating nodes, and the robots apply for nodes on demand.

2 System modeling and problem statement

This section discusses the MMRS model, gives definitions and descriptions of the glued nodes, and states the collision and deadlock avoidance problem.

2.1 Multi-mobile robot system

Let $\mathbb{N} = \{1, 2, ..., N\}$ be the indices of robots and N be the number of robots. R_i ($i \in \mathbb{N}$) denotes a robot in the MMRS. The roadmap in the system consists of a set of nodes and a set of edges, denoted as $G = (V, E)$. Let $\mathbb{M} = \{1, 2, ..., M\}$ be the indices of nodes in G. V_i ($i \in \mathbb{M}$) denotes a node in G. The edges in the roadmap are denoted as $E \subseteq (V, V)$, and $E_{m,n} = (V_m, V_n)$ denotes an edge from V_m to V_n. There may be two edges between two nodes, which means $E_{m,n} \neq E_{n,m}$. A path of R_i is denoted as $P_i = \{V_1, E_{1,2}, ..., E_{k-1,k}, V_k\}$, consisting of a series of nodes and edges. As shown in Fig. 2, the path of R_1 is $P_1 = \{V_4, E_{4,3}, V_3, E_{3,7}, V_7\}$.

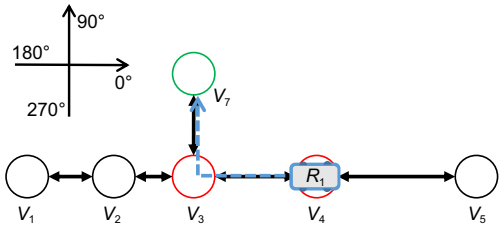

Fig. 2 An example of the path of robot R_1 (V_4 and V_3 are authorized to R_1)

There are many ways to generate paths for robots (Willms and Yang, 2006; Guruji et al., 2016), but this is not the focus of this study. Once the path of a robot is planned, the robot will move along the path to the end node. However, all nodes are managed by the control center, and the robot can move only along the path to the nodes that are authorized

by the control center. Fig. 3 is the architecture of the MMRS. Each robot sends its path and position information to the control center in real time and applies for the nodes ahead at a specific frequency when it needs to move. The control center determines whether the nodes requested by a robot can be authorized according to the collision and deadlock avoidance algorithm. The control center ensures that the nodes authorized to the robot are collision-free and deadlock-free. Then, the robot can move along the path to these authorized nodes.

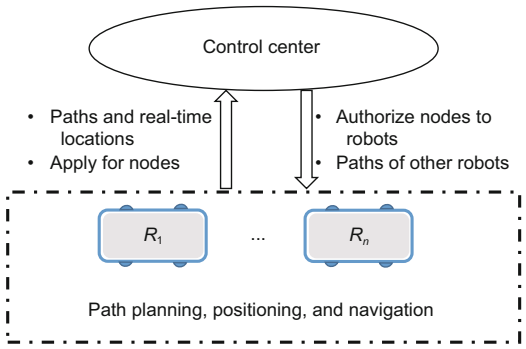

Fig. 3 Architecture of the multi-mobile robot system

To better explain the interaction process between the robot and the control center, we give the definitions of occupied nodes and applying nodes:

Definition 1 (Occupied nodes) The nodes for which a robot R_i has obtained authorization from the control center are called occupied nodes, denoted as OV_i.

Definition 2 (Applying nodes) The nodes for which a robot R_i applies to the control center are called applying nodes, denoted as AV_i.

The manner in which a robot R_i applies for nodes in this study is as follows: before R_i reaches the end node, once the distance from the robot to the last occupied node is less than LA_i, the robot applies for new nodes within the length of LA_i along the path. Those authorized applying nodes will become occupied nodes. The control law of LA_i is shown in Eq. (1), in which v_i is the real-time speed of R_i, a_i is the braking acceleration of R_i, and γ_i is a fixed value. The first term of the right-hand side of Eq. (1) is the braking distance of R_i, and the second term is used to reduce the number of times of robot brakes. Once the robot reaches a new node, the previous node on the path is released, and the path information also changes at this time.

$$\mathrm{LA}_i = \frac{v_i^2}{2a_i} + \gamma_i. \tag{1}$$

An example of the interaction process between the robots and the control center is shown in Fig. 2. Because V_4 and V_3 are already authorized to R_1, i.e., $\mathrm{OV}_i = \{V_4, V_3\}$, R_1 can move the farthest to V_3. However, to prevent R_1 from slowing down, R_1 starts to apply for V_7 before it reaches V_3 according to its own kinematic model, i.e., $\mathrm{AV}_i = \{V_7\}$. If V_7 is not authorized to R_1, then R_1 will stop after reaching V_3 and wait for the new nodes to be authorized. Once V_7 is authorized to R_1, denoted as $\mathrm{OV}_i = \{V_4, V_3, V_7\}$ (or $\mathrm{OV}_i = \{V_3, V_7\}$) and $\mathrm{AV}_i = \varnothing$, R_1 can move along the path to V_7. When R_1 reaches V_3, V_4 will be released and the path of R_1 changes to $P_1 = \{V_3, E_{3,7}, V_7\}$. A node cannot be authorized to other robots before it is released.

2.2 Glued nodes

A roadmap mentioned in this study is similar to a roadmap in the traffic road network. Nodes can be intersections of lanes, task points, or discrete points in the lanes. For example, in an MMRS of quick response (QR) code positioning, a straight road will be paved with multiple QR codes, and each QR code can be regarded as a node. The direction of the edge in the roadmap can be unidirectional or bidirectional, and the type can be a straight edge, a circular arc, a Bézier curve, and so on. Fig. 4 is an example of a roadmap.

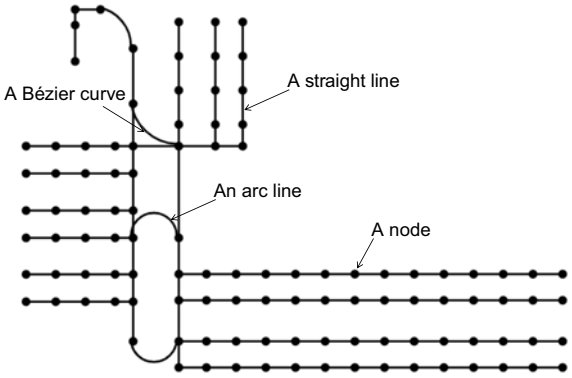

Fig. 4 An example of a roadmap

After a robot obtains the authorized nodes, the process of the robot moving along the path to the authorized nodes is no longer controlled by the control

center. Therefore, it is necessary to ensure that the robot can safely move to these nodes once the control center authorizes the nodes to the robot. However, the location of nodes and the types of edges in the roadmap may be diverse, which may lead to collisions among robots.

When two nodes are far enough apart, they are collision-free for two robots. As shown in Fig. 5a, the path nodes of the two robots are collision-free, but that is not always the case. As shown in Fig. 5b, robots R_1 and R_2 are moving on the way to V_2 and V_4, respectively. The distance between $E_{1,2}$ and $E_{3,4}$ is too small, causing the risk of collision if the robots continue to move. Similarly, in Fig. 5c, there may be a collision when R_1 moves on $E_{1,2}$ and R_2 rotates on V_5. In Fig. 5d, when R_1 and R_2 rotate on V_2 and V_5 simultaneously, they will collide with each other. To solve this kind of problem, we define the concept of glued nodes:

Definition 3 (Node action) A node action of R_i is denoted as a quadruple, $\Gamma_i^m = \langle {}^\bullet E_m, \alpha_m^i, V_m, \beta_m^i \rangle$, where the following concepts are defined:

(1) ${}^\bullet E_m$ is the preceding edge of V_m in P_i;

(2) α_m^i is the angle when the robot reaches V_m along ${}^\bullet E_m$ or the angle of the robot if V_m is the first node of P_i;

(3) V_m is a node in G;

(4) β_m^i is the angle at which the robot leaves V_m. If V_m is the last node of P_i, then $\beta_m^i = \alpha_m^i$.

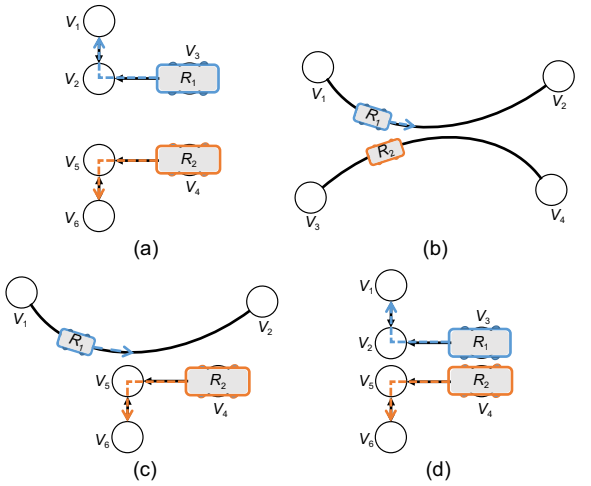

Fig. 5 Examples when two nodes are collision-free or not collision-free: (a) any two nodes are collision-free; (b) there is a risk of collision between two robots in the moving process; (c) there is a risk of collision when a robot rotates and a robot moves; (d) there is a risk of collision between two robots during rotation

Definition 4 (Action path) An action path of R_i is a set of node actions, denoted as $\mathscr{P}_i = \{\Gamma_i^1, \Gamma_i^2, \ldots, \Gamma_i^m\}$, $i \in \mathbb{N}$.

Node action describes the process of the robot moving from one node to another. As shown in Fig. 2, V_4 is the first node of the path and has no preceding edge, such that ${}^\bullet E_4 \in \varnothing$ and $\Gamma_1^4 = \langle {}^\bullet E, 180°, V_4, 180° \rangle$. $E_{4,3}$ is the preceding edge of V_3; after R_1 moves along $E_{4,3}$ to V_3, R_1 will rotate 90° clockwise to continue to V_7. So, $\Gamma_1^3 = \langle E_{4,3}, 180°, V_3, 90° \rangle$. Similarly, $\Gamma_1^7 = \langle E_{3,7}, 90°, V_7, 90° \rangle$. The action path of R_1 is $\mathscr{P}_1 = \{\Gamma_1^4, \Gamma_1^3, \Gamma_1^7\}$. A robot without a path can be considered to have an action path that contains only the node where it is currently located. As an example, when robot R_i has no path and its current node is V_m, it can be considered that $\mathscr{P}_i = \{\Gamma_i^m\}$. In this way, idle and working robots have a unified way of expression.

Definition 5 (Action area) When a robot R_i performs a node action Γ_i^m, the area it sweeps is called the action area, denoted as Θ_i^m.

Definition 6 (Glued nodes) $\forall m, n \in \mathbb{M}, \forall i, j \in \mathbb{N}$, $\Gamma_i^m \in \mathscr{P}_i$, $\Gamma_j^n \in \mathscr{P}_j$, if $\Theta_i^m \cap \Theta_j^n \neq \varnothing$, then V_m and V_n are a pair of glued nodes for R_i and R_j.

If V_m and V_n are a pair of glued nodes for R_i and R_j, we define a flag $\mathrm{GN}_{m,n}^{i,j} = 1$; otherwise, $\mathrm{GN}_{m,n}^{i,j} = 0$. Obviously, when $\mathrm{GN}_{m,n}^{i,j} = 1$, $\mathrm{GN}_{n,m}^{j,i} = 1$.

The concept of glued nodes is dynamic; that is, the glued relationship between two nodes varies with different robot sizes and different paths. At first, the concept of glued nodes is related to the real-time robot paths. As shown in Figs. 6a and 6b, if V_2 is authorized to R_1, because $\Theta_1^2 \cap \Theta_2^5 \neq \varnothing$, $\mathrm{GN}_{2,5}^{1,2} = 1$. However, as shown in Figs. 6c and 6d, although R_1 still passes through node V_2, $\Theta_1^2 \cap \Theta_2^5 = \varnothing$, $\mathrm{GN}_{2,5}^{1,2} = 0$. Furthermore, the concept of glued nodes is related to the robot sizes. As shown in Figs. 6c and 6d, if the two robots are larger, there may be $\mathrm{GN}_{2,5}^{1,2} = 1$. In other words, the concept of glued nodes is related to the real-time sizes and paths of the robots, so it is dynamic.

2.3 Problem statement

When robots move, potential collisions can result in damage to the robots, and deadlocks may lead to system stagnation until the deadlock is resolved. Therefore, a collision and deadlock avoidance algorithm is necessary for MMRSs.

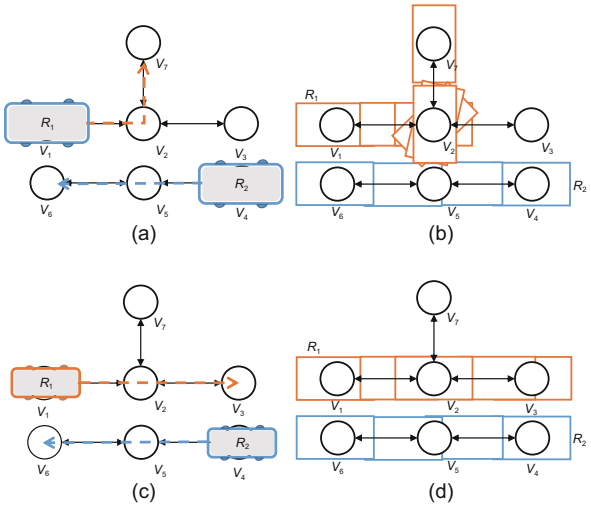

Fig. 6 An example showing the characteristics of the glued nodes: (a) paths of the two robots contain a pair of glued nodes, $\mathrm{GN}_{2,5}^{1,2}=1$; (b) areas swept by the two robots when executing Γ_1^2 and Γ_2^5 corresponding to (a); (c) paths of the two robots do not contain glued nodes; (d) areas swept by the two robots when executing Γ_1^2 and Γ_2^5 corresponding to (c)

The problem statement can be described as follows: Given an MMRS with a roadmap, find an effective controller to avoid collisions and deadlocks during the movement of robots.

Stable operation of MMRSs relies on various technologies, such as a navigation algorithm, communication technology, sensor technology, and motion control. This work focuses on motion coordination in MMRSs. For the convenience of study, we make some additional assumptions, but these assumptions do not detract from the contributions of our work: (1) The communication between robots and the control center is based on a wireless network, and there is no delay, error, or packet loss in the communication; (2) Each robot can always move along the planned path within an accepted derivation; (3) Taking into account the positioning accuracy, robots can stop only at nodes, and the robots are fault-free; (4) When an idle robot blocks a robot with a path, the idle robot will automatically plan a path that does not block the robot with a path.

3 Motion coordination

3.1 Collision avoidance

There is no doubt that whatever method is used to avoid collisions among robots, the essence is that multiple robots cannot appear in the same space simultaneously. In this subsection, we present the collision avoidance algorithm based on glued nodes. The main idea is that the control center decides whether to authorize a node to a robot according to whether the robot will collide with other robots during movement to the node.

Lemma 1 If $\exists V_m \in \mathrm{OV}_i$ and $\exists V_n \in \mathrm{OV}_j$, such that $\mathrm{GN}_{m,n}^{i,j}=1$, then R_i and R_j are at risk of collision during movement.

Proof $\mathrm{GN}_{m,n}^{i,j}=1$ means that when R_i and R_j execute Γ_i^m and Γ_j^n, $\Theta_i^m \cap \Theta_j^n \neq \varnothing$; i.e., the swept areas will overlap. This may cause R_i and R_j to collide with each other.

Lemma 2 If $\exists V_m | V_m \in \mathrm{OV}_i \wedge V_m \in \mathrm{OV}_j$, R_i and R_j are at risk of collision.

Proof Once a node is authorized to two robots, it means that the two robots can move to this node. Obviously, it is possible for the two robots to pass through this node simultaneously; i.e., a collision occurs.

Note that Lemmas 1 and 2 are about possible collisions, not inevitable collisions. For example, even if a node is authorized to two robots at the same time, but if the robots do not pass the node simultaneously, there is no risk of collision between the robots. However, in an actual industrial environment, a robot's movement process may be affected by human operations, faults, and so on, so the system cannot accurately estimate when the robot will arrive and leave a particular position. To ensure the robustness of the system, this study does not consider the time dimension.

Theorem 1 $\forall V_m \in \mathrm{OV}_i$ and $\forall V_n \in \mathrm{OV}_j$, if $\mathrm{OV}_i \cap \mathrm{OV}_j = \varnothing$ and $\mathrm{GN}_{m,n}^{i,j}=0$, then R_i and R_j are collision-free.

Proof If $\mathrm{OV}_i \cap \mathrm{OV}_j = \varnothing$, then a node is not authorized to the two robots at the same time. R_i and R_j will not collide with each other by passing through the same node at the same time. According to Definition 6, $\mathrm{GN}_{m,n}^{i,j}=0$ means that the areas swept by R_i and R_j when performing Γ_i^m and Γ_j^n do not overlap. In a word, regardless of whether OV_i and OV_j contain the same node, the robots will not collide with each other during the movement to the occupied nodes.

According to Theorem 1, the collision avoidance algorithm can be stated as shown in Algorithm 1. Let O_m denote node V_m to which the robot is authorized.

$O_m = R_i$ means $V_m \in \mathrm{OV}_i$, and $O_m \in \varnothing$ means that V_m is not authorized to any robots. As shown in Algorithm 1, along the sequence of nodes in the path, the control center determines whether the applying nodes can be authorized to the robot one by one (line 4). An applying node cannot be authorized to the robot if the node has been authorized to another robot (lines 5–7). In addition, if the applying node and another node are a pair of glued nodes, and another node has been authorized to other robots, the applying node cannot be authorized to the robot (lines 11–13). Once a node cannot be authorized, other applying nodes no longer need to be calculated (lines 6 and 12). Line 19 returns the nodes that can be authorized to R_i. Let N be the number of robots, and H and L denote the maximum numbers of the applying nodes and occupied nodes of the robot, respectively. The complexity of the algorithm is $O(NHL)$.

Algorithm 1 Collision avoidance: CA(R_i, AV_i)

1: **Input:** robot R_i and its applying nodes AV_i
2: **Output:** AN
// nodes without collision
3: Initialization: AN= $\varnothing$
4: **for** V_m in AV_i **do**
5: **if** $O_m \notin \varnothing$ **then**
6: **return** AN
7: **end if**
8: **for** R_j in R **do**
9: **if** $R_j \neq R_i$ **then**
10: **for** V_n in OV_j **do**
11: **if** $\mathrm{GN}^{i,j}_{m,n} = 1$ **then**
12: **return** AN
13: **end if**
14: **end for**
15: **end if**
16: **end for**
17: add V_m to AN
18: **end for**
19: **return** AN

In the example shown in Fig. 7, $\mathscr{P}_1 = \{\Gamma_1^1, \Gamma_1^2, \Gamma_1^3, \Gamma_1^4, \Gamma_1^8\}$, $\mathscr{P}_2 = \{\Gamma_2^5, \Gamma_2^4, \Gamma_2^3, \Gamma_2^7\}$, $\mathrm{OV}_1 = \{V_1, V_2\}$, $\mathrm{OV}_2 = \{V_5\}$, $\mathrm{AV}_1 = \{V_3, V_4, V_8\}$, and $\mathrm{AV}_2 = \{V_4, V_3, V_7\}$. $\mathrm{AV}_1 \cap \mathrm{AV}_2 \neq \varnothing$ means that R_1 and R_2 apply for the same nodes. However, the robot to which the nodes are authorized can be judged by factors such as task priority and road congestion. This study uses a simple strategy which is first-apply first-occupy. Assuming that R_2 applies to the control center for nodes earlier than R_1, according to Algorithm 1, V_4 can be authorized to R_2, but V_3 cannot be authorized for the reason $\mathrm{GN}^{1,2}_{2,3} = 1$. Thus, the result is that only V_4 is authorized to R_2. Note that this result will cause a deadlock between R_1 and R_2. The detection and avoidance of deadlocks will be presented in Section 3.2.

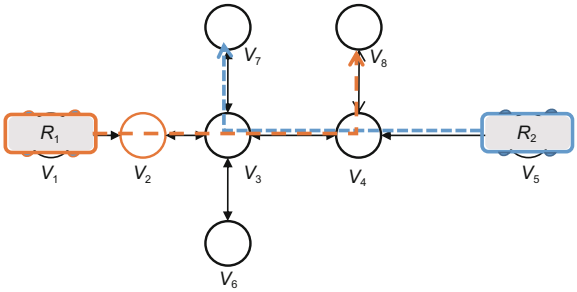

Fig. 7 An example of collision avoidance

3.2 Collision and deadlock avoidance

This subsection presents the analysis of the causes and categories of deadlocks, and designs collision and deadlock avoidance algorithms.

3.2.1 Direct deadlock

In Section 3.1, we propose an algorithm to avoid collisions among multiple robots. Each robot applies for nodes in the forward direction according to its kinematic model. If multiple robots block each other, deadlocks may occur. For example, in Fig. 8a, $\mathrm{GN}^{1,2}_{2,3} = 1$, $\mathrm{GN}^{1,2}_{3,4} = 1$, $\mathrm{AV}_1 = \mathrm{AV}_2 = \{V_3, V_5\}$, $\mathrm{OV}_1 = \{V_2\}$, and $\mathrm{OV}_2 = \{V_4\}$. However, V_3 cannot be authorized to R_1 and R_2 because $\mathrm{GN}^{1,2}_{2,3} = 1$ and $\mathrm{GN}^{1,2}_{3,4} = 1$, meaning that R_1 and R_2 wait for each other (i.e., a deadlock occurs). In Fig. 8b, $\mathrm{AV}_1 = \mathrm{OV}_2 = \{V_2\}$, $\mathrm{AV}_2 = \mathrm{OV}_3 = \{V_4\}$, $\mathrm{AV}_3 = \mathrm{OV}_4 = \{V_3\}$, and $\mathrm{AV}_4 = \mathrm{OV}_1 = \{V_1\}$. The four robots cannot move because the next node in the path of each robot is occupied by another robot, and the circular wait leads to a deadlock.

Definition 7 (Direct deadlock) There is a direct deadlock in an MMRS if multiple robots are in a circular wait.

Definition 8 (Block) R_i and R_j are two robots in an MMRS, and V_e is the last node in OV_i. If V_e is not the last node of P_i and V_m is the next node of V_e in P_i, $\forall V_n \in \mathrm{OV}_j$, if $V_m = V_n$ or $\mathrm{GN}^{i,j}_{m,n} = 1$, then R_i is blocked by R_j, denoted as $(R_i \to R_j)$.

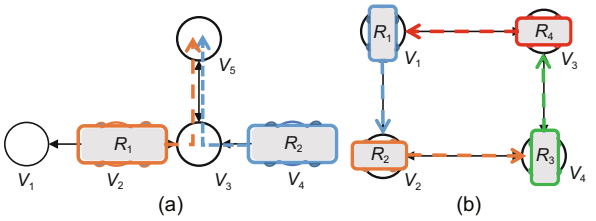

Fig. 8 Examples of deadlocks: (a) a deadlock of two robots; (b) a deadlock of four robots

Definition 9 (Block circle) A block circle is a sequence of blocks, like $\langle(R_1 \to R_2), (R_2 \to R_3), ..., (R_{n-1} \to R_n), (R_n \to R_1)\rangle$.

A block means that new nodes cannot be authorized to a robot because the robot avoids collisions with other robots. For example, as shown in Fig. 8a, V_3 is the next path node of the last node in OV_1 and OV_2. $\mathrm{GN}_{2,3}^{1,2} = 1$ and $\mathrm{GN}_{3,4}^{1,2} = 1$; according to Definition 8, there are two blocks $(R_1 \to R_2)$ and $(R_2 \to R_1)$, and the two blocks generate a block circle $\langle(R_1 \to R_2)(R_2 \to R_1)\rangle$.

Theorem 2 If there are no conflict circles in the system, there will be no direct deadlocks in the system.

Proof According to the definitions of the block and the block circle, if there is no block circle among multiple robots, at least one robot can apply for a new node because of other robots. Therefore, there will be no circular wait among the robots; i.e., there will be no direct deadlock among them. If there are no block circles among any number of robots, there are no direct deadlocks in the system.

Based on Theorem 2, the control center can detect whether the authorization of some nodes to a robot will cause a block circle, and prevent block circles from forming to avoid deadlocks. The combined collision and direct deadlock avoidance algorithm is shown in Algorithm 2. Let C be all blocks in the system, and C will be updated once a robot releases a node. ANs are those collision-free nodes that are returned by Algorithm 1 (line 4). Lines 5–7 try to add the nodes in AN to the occupied nodes of R_i, and generate new blocks. If there are newly generated blocks, then they are added to C (lines 8 and 9). If there are block circles in the system (line 10), it means that authorizing this node may lead to deadlock. The algorithm does not need to continue the calculation. Lines 13 and 16 add nodes that do not cause collisions and deadlocks to DAN to be returned.

Algorithm 2 Collision and direct deadlock avoidance: CDDA(R_i, AV_i)

1: **Input:** robot R_i and its applying nodes AV_i
2: **Output:** DAN
// nodes without collisions or deadlocks
3: Initialization: DAN= $\varnothing$
4: AN=CA(R_i, AV_i)
5: **for** V_m in AN **do**
6: add V_m to OV_i
7: generate new block CN
8: **if** CN$\notin \varnothing$ **then**
9: add CN to C
10: **if** there are block circles in C **then**
11: **break**
12: **else**
13: add V_m to DAN
14: **end if**
15: **else**
16: add V_m to DAN
17: **end if**
18: **end for**
19: **return** DAN

3.2.2 Impending deadlock

Algorithm 2 can avoid only direct deadlocks but cannot avoid impending deadlocks. Let us take Fig. 7 as an example to illustrate the impending deadlock. When R_2 applies node V_4, V_4 will be authorized to R_2 because it will not cause any collisions or direct deadlocks. V_3 cannot be authorized to R_2 because $\mathrm{GN}_{2,3}^{1,2} = 1$; i.e., R_2 is blocked by R_1. However, at this point, R_1 is not blocked by R_2, and if V_3 is authorized to R_1, then R_1 will move to V_3, forming a direct deadlock. Even if V_3 is not authorized to R_1 to avoid the deadlock, the result is that neither R_1 nor R_2 can continue to move. Therefore, a new deadlock occurs. In other words, when V_4 is authorized to R_2, R_1 and R_2 will inevitably form a deadlock.

Definition 10 (Impending deadlock) There is currently no deadlock among multiple robots, but these robots will inevitably form a deadlock after taking a few steps. This phenomenon is called an impending deadlock.

Because there may be impending deadlocks in the system, deadlock avoidance requires not only avoiding direct deadlocks, but also avoiding impending deadlocks.

Definition 11 (Conflict node) $\forall i, j \in \mathbb{N}, \forall m, n \in \mathbb{M}$, V_m is a conflict node for R_i and R_j if it meets one of the following conditions:

(1) $V_m \in P_i$ and $V_m \in P_j$;

(2) $\mathrm{GN}_{m,n}^{i,j} = 1$, $V_m \in P_i$, and $V_n \in P_j$.

Definition 12 (Conflict area) $\forall i, j \in \mathbb{N}$, the conflict area of R_i and R_j is the set of all their conflict nodes, denoted as $\Phi_{i,j}$.

In Fig. 7, V_2, V_3, and V_4 are conflict nodes, and they compose a conflict area $\Phi_{1,2} = \{V_2, V_3, V_4\}$. Let us analyze the impending deadlock according to the definition of the conflict area. Once two robots occupy the same conflict area, a deadlock will inevitably occur because the two robots will stop in this conflict area no matter how they move. There are similar laws for multiple robots, which will be discussed and proved below.

Definition 13 (Conflict occupation) If $\mathrm{OV}_i \cap \Phi_{i,j} \neq \varnothing$, there is a conflict occupation of R_i, denoted as $R_i \to \Phi_{i,j}$.

Definition 14 (Conflict circle) A conflict circle is a sequence of conflict occupation, like $\langle R_1 \to \Phi_{1,2}, R_2 \to \Phi_{2,3}, ..., R_{n-1} \to \Phi_{n-1,n}, R_n \to \Phi_{n,1} \rangle$.

As shown in Fig. 7, because V_2 has been authorized to R_1, there is a conflict occupation $R_1 \to \Phi_{1,2}$. If V_4 is authorized to R_2 at the same time, there will be another conflict occupation $R_2 \to \Phi_{2,1}$, and a conflict circle $\langle R_1 \to \Phi_{1,2}, R_2 \to \Phi_{2,1} \rangle$ occurs.

Lemma 3 A block circle is a type of conflict circle.

Proof Based on the definition of the block, $(R_i \to R_j)$ means $V_m = V_n$ or $\mathrm{GN}_{m,n}^{i,j} = 1$, $V_m \in P_i$ and $V_n \in P_j$. If $V_m = V_n$, then $V_m \in P_i$ and $V_m \in P_j$, which meets the first condition of Definition 11. $\mathrm{GN}_{m,n}^{i,j} = 1$ meets the second conditions of Definition 11. Therefore, V_m is a conflict node, and $V_m \in \Phi_{i,j}$. Then $(R_i \to R_j)$ can be transformed to $R_j \to \Phi_{i,j}$. So, the block circle $\langle (R_1 \to R_2), (R_2 \to R_3), ..., (R_{n-1} \to R_n), (R_n \to R_1) \rangle$ can be transformed to $\langle R_1 \to \Phi_{1,2}, R_2 \to \Phi_{2,3}, ..., R_{n-1} \to \Phi_{n-1,n}, R_n \to \Phi_{n,1} \rangle$.

Note that the conflict circle is generated based on the conflict areas formed by the remaining unfinished paths of robots instead of the complete static paths. The reason is that the nodes that a robot has gone through will be released. It will not cause any collisions or deadlocks, except for those glued nodes. Furthermore, based on the above definitions and theorems, this study describes the relationship between conflict circles and deadlocks, as stated in Theorem 3:

Theorem 3 If there are no conflict circles in the system, there will be no deadlocks in the system.

Proof According to Lemma 3, there is no block circle if there is no conflict circle in the system. Furthermore, based on Theorem 2, there will be no direct deadlock without a block circle. Therefore, we can conclude that there will be no direct deadlock without a conflict circle. In addition, for impending deadlock, according to the definitions of conflict occupation and conflict circle, a conflict area is actually a common space swept by two robots when they move along their paths. Once a robot occupies the conflict area, it may induce another robot to avoid collisions and stop in the future. If multiple robots do not form a conflict circle, then at least one of these robots can move without being blocked by other robots. There will be no direct deadlocks among the robots in the future. Therefore, there are no impending deadlocks among the robots. The absence of a conflict circle means that no deadlock can be formed among any number of robots; that is, the system is free of deadlocks.

Now that the conditions for avoiding deadlock have been proved, we present the collision and deadlock avoidance algorithm as shown in Algorithm 3. Line 4 calls Algorithm 1 to obtain collision-free nodes, and then the algorithm judges whether these nodes are deadlock-free. Lines 5–20 search for the farthest node among these nodes that is not in conflict areas, and the robot can move to this node without deadlock, because if a node is not in a conflict area, a conflict occupation cannot be formed and there will be no conflict circle. Lines 21–30 judge whether the authorized nodes form a conflict circle, and only those nodes that do not form a conflict circle can be occupied by the robot.

In the worst case, all nodes are in conflict areas. Therefore, in lines 8 and 22, the complexity of judging whether a node is in a conflict area is $O(M)$, in which M is the number of nodes in G. The occupation relationship of the conflict areas among robots can form a directed graph, and the complexity of finding the circles in the directed graph is $O(V + E)$ (Tarjan, 1971). V and E are the numbers of nodes and edges in the graph, respectively (line 23). The complexity of the algorithm is $O(HM(V + E))$, where H is the number of nodes in AN.

Fig. 9 shows the collision and deadlock avoidance process of four robots. Initially, as shown

Algorithm 3 Collision and deadlock avoidance: CDA(R_i, AV$_i$)

Input: robot R_i and its applying nodes AV$_i$
Output: DAN
// nodes without collisions or deadlocks
Initialization: DAN= $\varnothing$
AN= CA(R_i, AV$_i$)
$V_s \in \varnothing$
flag=true
for V_m in AN **do**
  if V_m not in any conflict areas **then**
    $V_s = V_m$
    flag=false
  end if
end for
for V_m in AN **do**
  if flag=false **then**
    if $V_m = V_s$ **then**
      flag=true
    end if
    add V_m to DAN
    continue
  end if
  add V_m to OV$_i$
  if $R_i \to \Phi_{i,j}$ **then**
    if a conflict circle containing $R_i \to \Phi_{i,j}$ is generated **then**
      break
    else
      add V_m to OV$_i$
    end if
  else
    add V_m to DAN
    // V_m not in any conflict areas
  end if
end for
return DAN

in Fig. 9a, there are four conflict areas: $\Phi_{1,2} = \{V_4, V_6\}$, $\Phi_{2,3} = \{V_2\}$, $\Phi_{3,4} = \{V_1, V_5, V_9\}$, and $\Phi_{4,1} = \{V_3\}$. Suppose that R_3 first applies for node V_3 and receives authorization. Subsequently, the application of R_2 for node V_6 will not succeed, because once V_6 is authorized to R_2, a conflict circle $\langle R_1 \to \Phi_{1,4}, R_4 \to \Phi_{4,3}, R_3 \to \Phi_{3,2}, R_2 \to \Phi_{2,1} \rangle$ will be generated. Therefore, Algorithm 3 will prevent V_6 from being authorized to R_2, as shown in Fig. 9b. Then, as shown in Figs. 9c and 9d, R_3 continues to move forward, and the four robots will eventually reach their respective destinations.

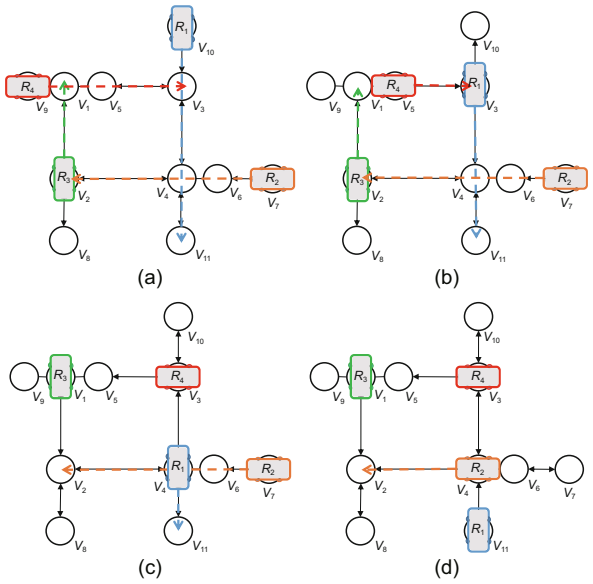

Fig. 9 An example of collision and deadlock avoidance for four robots: (a) conflict occupations ($R_3 \to \Phi_{3,4}$), blocks ($R_3 \to R_4$), glued nodes ($GN^{3,4}_{1,9}$=1, $GN^{3,4}_{1,5}$=1, $GN^{1,2}_{4,6}$=1); (b) conflict occupations ($R_3 \to \Phi_{3,4}$, $R_4 \to \Phi_{4,1}$, $R_2 \to \Phi_{2,3}$), blocks ($R_3 \to R_4$, $R_4 \to R_1$), glued nodes ($GN^{3,4}_{1,5}$=1, $GN^{1,2}_{4,6}$ =1); (c) conflict occupations ($R_1 \to \Phi_{1,2}$), blocks ($R_1 \to R_1$); (d) no conflict occupations, no blocks, and no glued nodes

4 Experiment implementation and results

In this section, we describe multiple experiments to verify the effect of the proposed collision and deadlock avoidance algorithm. All experiments are realized with robot scheduling software and robot simulation software. The robot scheduling software has been applied in many real projects, with functions of task dispatch, path planning, collision and deadlock avoidance, and deadlock unlocking. All simulations run on a desktop running Windows 10, equipped with an Intel® Xeon® Platinum 8280L 2.6 GHz CPU and 64 GB of RAM.

4.1 Comparison of different collision and deadlock avoidance algorithms

To compare the performance of different collision and deadlock avoidance methods, we compare Algorithm 2 (collision and direct deadlock avoidance, CDDA) and Algorithm 3 (collision and deadlock avoidance, CDA) based on a simulation of a real industrial scene.

The experiments are based on an actual air

conditioning production line containing eight mobile robots and 38 workstations, as shown in Fig. 10. This scene contains two production lines, where the edges in the roadmap are all bidirectional, and the robots are constantly carrying out transportation tasks between workstations. Once a task is generated, the idle robot with the closest distance will be assigned to perform the task.

The parameters of robots in the simulation are shown in Table 1, where v denotes the maximum moving speed of robots, δ denotes the acceleration, θ denotes the braking deceleration, el and ew denote the length and width of the robots when they are not loaded respectively, and fl and fw denote the length and width of the robots when they are loaded respectively. γ takes a fixed value of 4 m. Because CDDA cannot avoid the impending deadlock, once a deadlock occurs, the software will call the deadlock unlocking method to resolve the deadlock.

The number of deadlock avoidances, the number of deadlock unlockings, and the average task time of these two algorithms are counted. Let $\mathbb{K} = \{1, 2, ..., K\}$ be the indices of tasks, and K be the number of tasks. TK_i $(i \in \mathbb{K})$ denotes a task in the system. The average task time is calculated as

$$\sum_{i=1}^{K} \frac{T_{\mathrm{e}}^i - T_{\mathrm{s}}^i}{K}, \tag{2}$$

Table 1 Experimental parameters of robots in the scene shown in Fig. 10

Parameter	Value
v	1.5 m/s
θ	0.8 m/s^2
δ	0.7 m/s^2
el×ew	2.0 m×1.4 m
fl×fw	3.2 m×2.0 m
γ	4 m

where T_{s}^i and T_{e}^i denote the start time and end time of TK_i, respectively.

The experiments randomly generate 300 transportation tasks among the workstations, and the experimental results are shown in Table 2. Although the number of deadlock avoidances of CDDA is not much different from that of CDA, 29 deadlocks are not successfully avoided. These deadlocks are caused by the existence of bidirectional roads, which causes the robots' running paths to overlap, resulting in impending deadlocks. In addition, because CDDA cannot avoid impending deadlocks, even if direct deadlocks are avoided, the robots still cannot move forward. The process of unlocking the deadlock increases the time the robot takes to perform the task. It also proves that deadlock avoidance is more efficient than deadlock detection and resolution.

CDDA spends ≥22.43% of the average task time compared to CDA, which is caused by numerous deadlocks being unlocked. Similar results can be drawn from Fig. 11. Regardless of the number of tasks, the total task time of CDDA is much higher than that of CDA. So, CDA has better performance.

To study the relationship between the number of deadlocks and the number of robots, we initially set up 300 tasks and count the number of deadlock avoidances with different numbers of robots. Using CDA, the experimental results are shown in Fig. 12. It can be seen that when the number of robots increases, the number of deadlock avoidances

Table 2 Experimental results of CDDA and CDA

Algorithm	Number of deadlock avoidances	Number of deadlock unlocks	Average task time (s)
CDDA	34	29	43.11
CDA	32	0	35.21

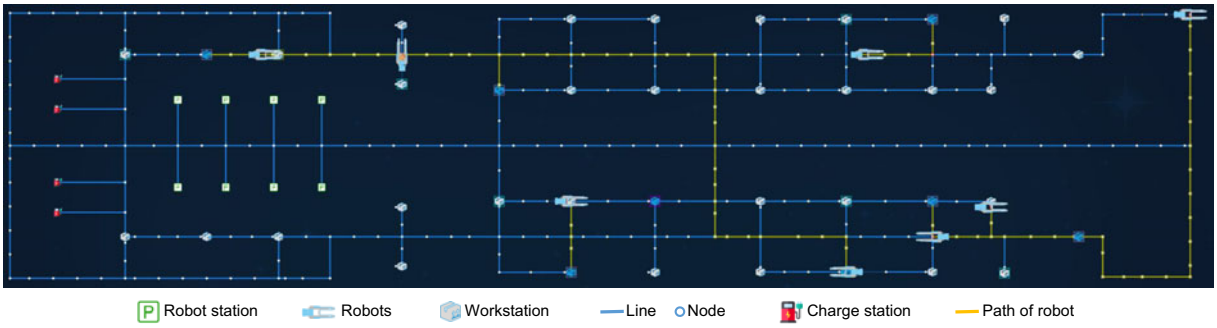

Fig. 10 An experimental scene of an air conditioning production line with eight mobile robots

also gradually increases. The increase in the number of robots leads to a rise in the possibility of overlapping paths, making it easier to trigger deadlock avoidance. In addition, there is no collision in all experiments, which means that the collision avoidance algorithm is effective.

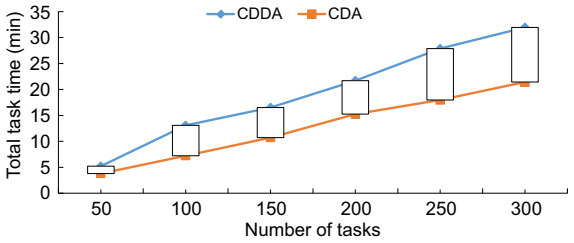

Fig. 11 Relationship between the total task time and the number of tasks with the collision and deadlock avoidance (CDA) algorithm and collision and direct deadlock avoidance (CDDA) algorithm

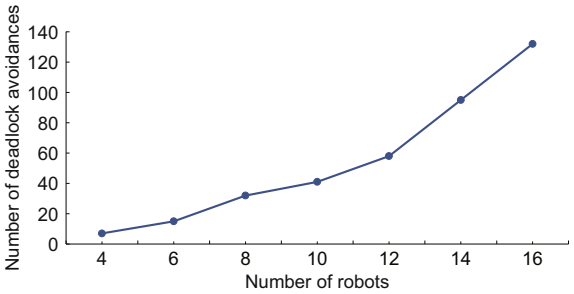

Fig. 12 Simulation results of the number of deadlock avoidances under different numbers of robots with the collision and deadlock avoidance (CDA) algorithm

4.2 Comparison with other methods

Methods based on zone control are classic and effective for motion coordination in MMRSs. The map of robots is composed of disjoint zones. Each robot has its own zone at every moment, so there will be no collisions among the robots. Deadlock avoidance is achieved by analyzing the sequence of zones that the robots' paths contain. However, this method is a bit conservative.

For example, in a compact roadmap shown in Fig. 13a, using the method in this study, when neither robot is loaded, there are no glued nodes, and both robots can pass normally. Only when the two robots are loaded, do the glued nodes make it impossible for the two robots to pass through at the same time (Fig. 13b). However, based on the principle of zone control, there are three zones shown in Fig. 13c. This means that the two robots cannot pass through this channel at the same time under any circumstance.

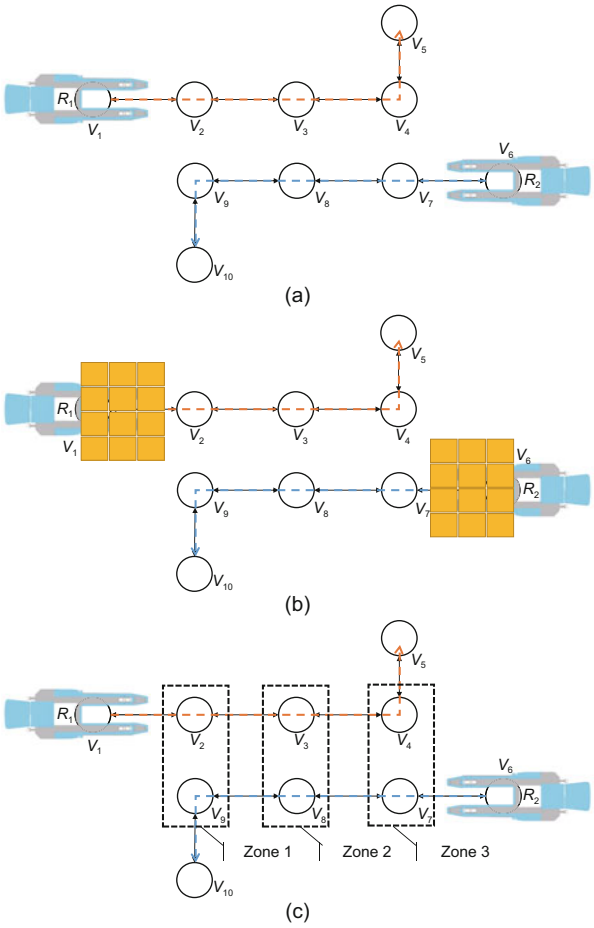

Fig. 13 Experimental scene with a compact roadmap: (a) there are no glued nodes when the two robots are not loaded; (b) when the two robots are loaded, there are three pairs of glued nodes $GN^{1,2}_{2,9}$=1, $GN^{1,2}_{3,8}$ =1, and $GN^{1,2}_{4,7}$=1; (c) based on the principle of zone control, V_2, V_3, V_4, V_7, V_8, and V_9 belong to three zones

This study verifies the efficiency of CDA based on a casting production line scenario, where all robots are based on QR code positioning and navigation. Each QR code can be regarded as a node in the roadmap and all edges in the roadmap are bidirectional. As shown in Fig. 14, there are 22 workstations, among which robots transport materials. The task of the robot is always to transport materials from one workstation to another. Each task consists of four subtasks: (1) move to a workstation, (2) load cargo, (3) move to another workstation with cargo, and (4) unload the cargo. Robots with no tasks will move to the robot stations and wait for new tasks.

Whenever the battery level of a robot drops below a set value (15% in the experiments), the robot will move to a charge station to charge. The parameters of the robots are shown in Table 3.

Each set of experiments is set up with 100 tasks, and each task is always assigned to the closest idle robot. The paths of the robots are generated using the A* algorithm with the shortest distance as the goal. We compare the CDA method with the methods in Małopolski (2018) (chains of reservations, COR) and Zając and Małopolski (2021) (structural on-line control policy, SOCP). We use the average task time, average robot waiting time, and total mileage of robots to evaluate the performance of different methods. The average task time is calculated using expression (2). The average waiting time is calculated as

$$\sum_{i=1}^{K}\sum_{j=1}^{N}\frac{T_{\mathrm{w}}^{i,j}}{K}, \tag{3}$$

where $T_{\mathrm{w}}^{i,j}$ denotes the time at which R_j stops and waits while executing TK_i to avoid collisions and deadlocks. When TK_i is not executed by R_j, $T_{\mathrm{w}}^{i,j}=0$. The total mileage is calculated as

$$\mathrm{TM}=\sum_{i=1}^{K}\sum_{j=1}^{N}M_{i,j}, \tag{4}$$

where $M_{i,j}$ denotes the mileage traveled by R_j when it executes TK_i. When TK_i is not executed by R_j, $M_{i,j}=0$.

Because the number of tasks remains unchanged in each experiment, the average task time reflects the efficiency of different methods. The total mileage represents the energy consumption of the robots. The experimental results with different numbers of robots are shown in Table 4.

The experimental results are similar to those in Zając and Małopolski (2021), and COR and SOCP perform poorly in roadmaps where all edges are bidirectional. CDA is based on the real-time paths of robots to avoid deadlocks and is not affected by bidirectional edges. In terms of the average task time, as the number of robots increases, the advantages of CDA become more and more obvious, as shown in Fig. 15. When the number of robots is 6, CDA consumes 34.09% and 41.01% less time than SOCP and COR, respectively. In terms of the total mileage, when using GN, the robot travels fewer miles (less energy consumption) than other methods. As shown in Figs. 15 and 16, both the average task time and the average waiting time increase with the increase of the number of robots. This is because the more robots there are in the system, the more traffic congestion is created; thus, the robots are more likely to stop and wait to avoid collisions and deadlocks.

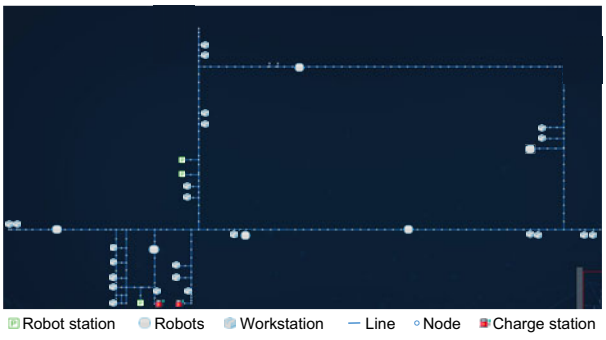

Fig. 14 An experimental scene of a casting production line

Table 3 Experimental parameters of robots in the scene shown in Fig. 14

Parameter	Value
v	1.2 m/s
θ	0.6 m/s^2
δ	0.5 m/s^2
el×ew	1.5 m×1.2 m
fl×fw	1.7 m×1.2 m
γ	4 m

5 Conclusions and future work

This paper begins by describing the concept of glued nodes based on the roadmap. The concept of

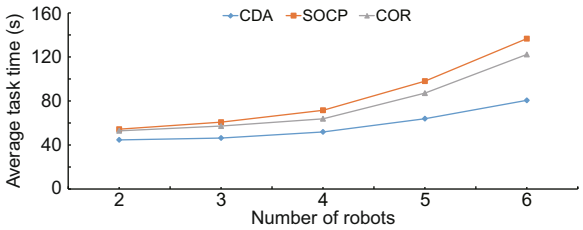

Fig. 15 Experimental results of the average task time

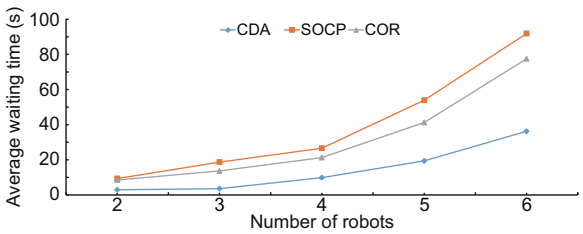

Fig. 16 Experimental results of the average waiting time

Table 4 Experimental results of different methods

Number of robots	Average task time (s)			Average waiting time (s)			Total mileage (m)		
	CDA	COR	SOCP	CDA	COR	SOCP	CDA	COR	SOCP
2	44.64	54.32	52.87	2.92	9.35	8.51	3998.42	4803.14	4822.61
3	46.33	60.71	57.19	3.57	18.67	13.66	3785.81	4523.50	4510.94
4	51.86	71.53	63.77	9.87	26.58	21.26	3624.12	4342.62	4328.61
5	63.87	97.99	87.12	19.39	53.93	41.24	3562.41	4261.56	4248.36
6	80.56	136.58	122.24	36.25	91.89	77.52	3521.86	4207.23	4196.52

glued nodes is related to the real-time paths and sizes of robots, and is dynamic and more precise compared to zone-control method.

Then, we propose a hybrid control architecture, by which the robots and the control center interact using the application and authorization of nodes. Theories and methods of collision and deadlock avoidance are proposed based on the concept of glued nodes. We analyze the difference between direct deadlock and impending deadlock, and present collision and deadlock avoidance algorithms.

Finally, we verify the effectiveness of the proposed algorithms in several experiments based on multiple industrial scenarios. Compared with CDDA, CDA can effectively avoid deadlock and improve the efficiency of the system. Compared with the other zone-control methods, CDA shows superiority in terms of average task time, average waiting time, and total mileage.

The proposed method has some limitations, however. It must be calculated based on a roadmap, and it can be applied only online based on real-time robot data.

In the future, we will further study path planning and deadlock unlocking based on the concept of glued nodes.

Contributors

Zichao XING and Weimin WU designed the research. Zichao XING and Xingkai WANG processed the data. Zichao XING drafted the paper. Zichao XING, Xingkai WANG, and Shuo WANG performed the experiments. Ruifen HU helped organize the paper. Zichao XING and Weimin WU revised and finalized the paper.

Compliance with ethics guidelines

Zichao XING, Xingkai WANG, Shuo WANG, Weimin WU, and Ruifen HU declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Alonso-Mora J, Breitenmoser A, Rufli M, et al., 2013. Optimal reciprocal collision avoidance for multiple non-holonomic robots. 10th Int Symp on Distributed Autonomous Robotic Systems, p.203-216. https://doi.org/10.1007/978-3-642-32723-0_15

Alonso-Mora J, DeCastro JA, Raman V, et al., 2018. Reactive mission and motion planning with deadlock resolution avoiding dynamic obstacles. *Auton Robot*, 42(4):801-824. https://doi.org/10.1007/s10514-017-9665-6

Chen HF, Wu NQ, Zhou MC, 2016. A novel method for deadlock prevention of AMS by using resource-oriented Petri nets. *Inform Sci*, 363:178-189. https://doi.org/10.1016/j.ins.2015.08.016

Coffman EG, Elphick M, Shoshani A, 1971. System deadlocks. *ACM Comput Surv*, 3(2):67-78. https://doi.org/10.1145/356586.356588

de Ryck M, Versteyhe M, Debrouwere F, 2020. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *J Manuf Syst*, 54:152-173. https://doi.org/10.1016/j.jmsy.2019.12.002

Draganjac I, Petrović T, Miklić D, et al., 2020. Highly-scalable traffic management of autonomous industrial transportation systems. *Robot Comput-Integr Manuf*, 63:101915. https://doi.org/10.1016/j.rcim.2019.101915

Fanti MP, Mangini AM, Pedroncelli G, et al., 2015. Decentralized deadlock-free control for AGV systems. American Control Conf, p.2414-2419. https://doi.org/10.1109/ACC.2015.7171094

Guruji AK, Agarwal H, Parsediya DK, 2016. Time-efficient A* algorithm for robot path planning. *Proc Technol*, 23:144-149. https://doi.org/10.1016/j.protcy.2016.03.010

Habermann AN, 1969. Prevention of system deadlocks. *Commun ACM*, 12(7):373-377, 385. https://doi.org/10.1145/363156.363160

Jager M, Nebel B, 2001. Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots. Proc IEEE/RSJ Int Conf on Intelligent Robots and Systems, Expanding the Societal Role of Robotics in the Next Millennium, p.1213-1219. https://doi.org/10.1109/IROS.2001.977148

Krnjak A, Draganjac I, Bogdan S, et al., 2015. Decentralized control of free ranging AGVs in warehouse environments. IEEE Int Conf on Robotics and Automation, p.2034-2041. https://doi.org/10.1109/ICRA.2015.7139465

Li Q, Pogromsky A, Adriaansen T, et al., 2016. A control of collision and deadlock avoidance for automated guided vehicles with a fault-tolerance capability. *Int J Adv Robot Syst*, 13(2):64. https://doi.org/10.5772/62685

Luo JL, Wan YX, Wu WM, et al., 2020. Optimal Petri-net controller for avoiding collisions in a class of automated guided vehicle systems. *IEEE Trans Intell Transp Syst*, 21(11):4526-4537. https://doi.org/10.1109/TITS.2019.2937058

Małopolski W, 2018. A sustainable and conflict-free operation of AGVs in a square topology. *Comput Ind Eng*, 126:472-481. https://doi.org/10.1016/j.cie.2018.10.002

Moorthy RL, Hock-Guan W, Wing-Cheong N, et al., 2003. Cyclic deadlock prediction and avoidance for zone-controlled AGV system. *Int J Prod Econ*, 83(3):309-324. https://doi.org/10.1016/S0925-5273(02)00370-5

Reveliotis S, 2020. An MPC scheme for traffic coordination in open and irreversible, zone-controlled, guidepath-based transport systems. *IEEE Trans Autom Sci Eng*, 17(3):1528-1542. https://doi.org/10.1109/TASE.2019.2963632

Tarjan R, 1971. Depth-first search and linear graph algorithms. 12th Annual Symp on Switching and Automata Theory, p.114-121. https://doi.org/10.1109/SWAT.1971.10

Wang X, Kloetzer M, Mahulea C, et al., 2015. Collision avoidance of mobile robots by using initial time delays. 54th IEEE Conf on Decision and Control, p.324-329. https://doi.org/10.1109/CDC.2015.7402221

Willms AR, Yang SX, 2006. An efficient dynamic system for real-time robot-path planning. *IEEE Trans Syst Man Cybern Part B Cybern*, 36(4):755-766. https://doi.org/10.1109/TSMCB.2005.862724

Wu NQ, Zhou MC, 2007. Deadlock resolution in automated manufacturing systems with robots. *IEEE Trans Autom Sci Eng*, 4(3):474-480. https://doi.org/10.1109/TASE.2006.888049

Xing KY, Wang F, Zhou MC, et al., 2018. Deadlock characterization and control of flexible assembly systems with Petri nets. *Automatica*, 87:358-364. https://doi.org/10.1016/j.automatica.2017.09.001

Xing ZC, Chen XY, Wang XK, et al., 2022. Collision and deadlock avoidance in multi-robot systems based on glued nodes. *IEEE/CAA J Autom Sin*, 9(7):1327-1330. https://doi.org/10.1109/JAS.2022.105710

Yoo JW, Sim ES, Cao CX, et al., 2005. An algorithm for deadlock avoidance in an AGV system. *Int J Adv Manuf Technol*, 26(5):659-668. https://doi.org/10.1007/s00170-003-2020-4

Yu JJ, LaValle SM, 2016. Optimal multirobot path planning on graphs: complete algorithms and effective heuristics. *IEEE Trans Robot*, 32(5):1163-1177. https://doi.org/10.1109/TRO.2016.2593448

Zając J, Małopolski W, 2021. Structural on-line control policy for collision and deadlock resolution in multi-AGV systems. *J Manuf Syst*, 60:80-92. https://doi.org/10.1016/j.jmsy.2021.05.002

Zhang LJ, Kim YJ, Manocha D, 2007. A hybrid approach for complete motion planning. IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.7-14. https://doi.org/10.1109/IROS.2007.4399064

Zhao YL, Liu XP, Wu SB, et al., 2021. Spare zone based hierarchical motion coordination for multi-AGV systems. *Simul Model Pract Theory*, 109:102294. https://doi.org/10.1016/j.simpat.2021.102294

Zhong MS, Yang YS, Dessouky Y, et al., 2020. Multi-AGV scheduling for conflict-free path planning in automated container terminals. *Comput Ind Eng*, 142:106371. https://doi.org/10.1016/j.cie.2020.106371

Zhou Y, Hu HS, Liu Y, et al., 2020. A distributed method to avoid higher-order deadlocks in multi-robot systems. *Automatica*, 112:108706. https://doi.org/10.1016/j.automatica.2019.108706