Supplementary Material for "Collision and Deadlock Avoidance in Multi-Robot Systems Based on Glued Nodes"

Zichao Xing, Xinyu Chen, Xingkai Wang, Weimin Wu, *Senior Member, IEEE*, and Ruifen Hu

A. Roadmap

The roadmap mentioned in the letter is similar to the roadmap in the traffic road network. Nodes can be intersections of lanes, task points, or discrete points on the lanes. For example, in a multi-robot system 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 an 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. 1 is an example of a roadmap in a real warehousing scenario.

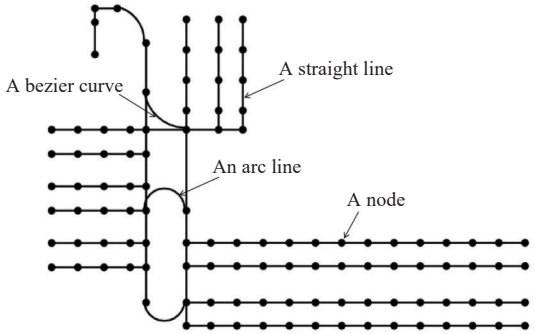

Fig. 1. An example of roadmap.

B. Control Architecture

There are many ways to generate the paths of robots [1], [2], which are not the focus of the letter. After a robot's path is generated, the robot will move along the path to the end node. However, all nodes are managed by the control center, and the robot can only move along the path to those nodes authorized by the control center. The architecture of the multi-robot system is shown in Fig. 2. 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 will determine 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, and then, the robot can move along the path to these authorized nodes.

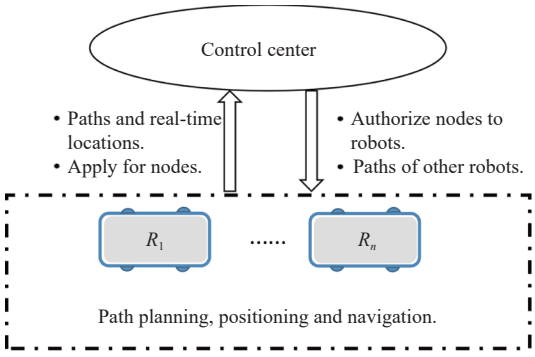

Fig. 2. The architecture of the multi-robot system.

Functions of the control center:

1) It responds to the robots' request of applying for nodes, and decides whether authorize the nodes to the robots according to the collision and deadlock avoidance algorithm.

2) It sends the path and location information of other robots to a robot so that the robot can plan the path.

Functions of the robots:

1) Each robot plans the path according to the task and move along the path, and can only reach the nodes authorized by the control center.

2) Each robot sends path and location information to the control center.

The definitions of occupied nodes and applying nodes are as follows.

Definition 1 (occupied nodes): The nodes that 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 that a robot R_i applies to the control center are called applying nodes, denoted as AV_i.

Let L_{ov}^i and L_{av}^i be the number of nodes in OV_i and AV_i respectively, L is a fixed value to control the number of applying nodes and occupied nodes for robots. The way about how a robot applies for nodes in this paper is: before the robot reaches the end node, once $L_{ov}^i < L$, the robot applies for new nodes along the path direction. The number of newly applied nodes is $L_{av}^i = L - L_{ov}^i$. Once the application is successful, these nodes will become occupied nodes. When the robot reaches a node, the previous node on the path is released and removed from OV_i.

We take Fig. 3 as an example to explain the interaction process between R_1 and the control center. Since V_4 and V_3 are already authorized to R_1, i.e., $OV_i = \{V_4, V_3\}$, R_1 can move the farthest to V_3. However, in order to prevent R_1 from slowing down, R_1 starts to apply for V_7, i.e., $AV_i = \{V_7\}$, before it reaches V_3 according to its own kinematic model. If V_7 is not authorized to R_1, then R_1 will stop after reaching V_3 and waits for the new nodes to be authorized. And when R_1 reaches V_3, V_4 will be released, it is denoted as $OV_i = \{V_3\}$ and $AV_i = \{V_7\}$. Once V_7 is authorized to R_1, denoted as $OV_i = \{V_4, V_3, V_7\}$(or $OV_i = \{V_3, V_7\}$) and $AV_i = \emptyset$, R_1 can move along the path to V_7, and before V_7 is released by R_1, V_7 cannot be authorized to other robots.

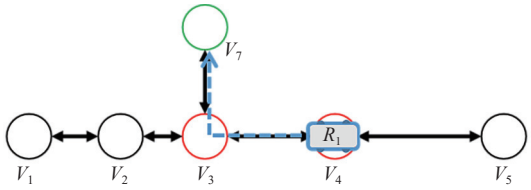

Fig. 3. An example of the path of robot R_1. V_4 and V_3 are authorized to R_1.

C. Glued Nodes

Since the method proposed in this paper is implemented based on the concept of glued nodes, how to judge whether a pair of nodes are glued nodes is very important. According to Definition 3, if the regions swept by two robots overlap when they perform two node actions, then the two nodes are a pair of glued nodes for the two robots. When it comes to specific calculations, a robot can be represented by a convex polygon, as shown in Fig. 4, an automated forklift can be represented by the smallest rectangle that can wrap it. Then, an action area can be discretized into a set of convex polygons. As shown in Fig. 5, an action area of the robot can be discretized into

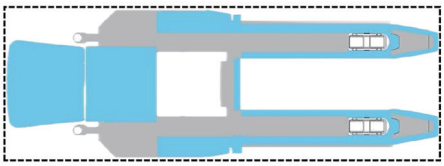

Fig. 4. An automated forklift is represented using the smallest rectangle that wraps it.

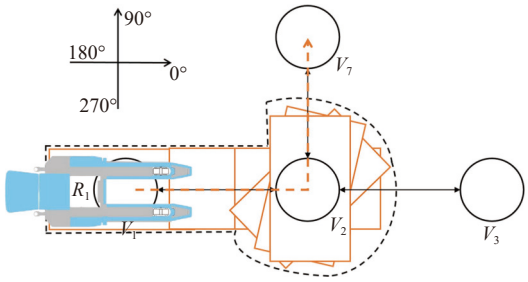

Fig. 5. An example of discretizing an action area into a set of rectangles.

a set of continuous intersecting rectangles.

The separating axis theorem is a classic and effective method for checking whether two convex polygons intersect [3]. Therefore, for R_i and R_j and their action areas Θ_i^m and Θ_j^n, the system first discretizes Θ_i^m and Θ_j^n into two sets of convex polygons $Rect_i^m$ and $Rect_j^n$. Then, it can be judged whether Θ_i^m and Θ_j^n overlap by judging whether there are intersecting rectangles in $Rect_i^m$ and $Rect_j^n$.

There is an example to help understand the concept of glued nodes, as shown in Fig. 6. At first, the concept of glued nodes are related to the paths of robots. As shown in Figs. 6(a) and 6(b), if V_2 is authorized to R_1, because $\Theta_1^2 \cap \Theta_2^5 \neq \emptyset$, $GN_{2,5}^{1,2} = 1$. However, as shown in Figs. 6(c) and 6(d), although R_1 still passes through node V_2, but $\Theta_1^2 \cap \Theta_2^5 = \emptyset$, $GN_{2,5}^{1,2} = 0$. Furthermore, the concept of glued nodes are related to the sizes of robots. As shown in Figs. 6(c) and 6(d), if the sizes of the two robots are larger, there may be $GN_{2,5}^{1,2} = 1$. At last, the concept of glued nodes are related to the structure of roadmap. As shown in Fig. 6(b), if V_2 and V_5 are far enough apart, they will not be a pair of glue nodes for R_1 and R_2. In a word, the concept of glued nodes is not only related to the structure of the roadmap but also to the real-time size and path of the robot, so it is dynamic.

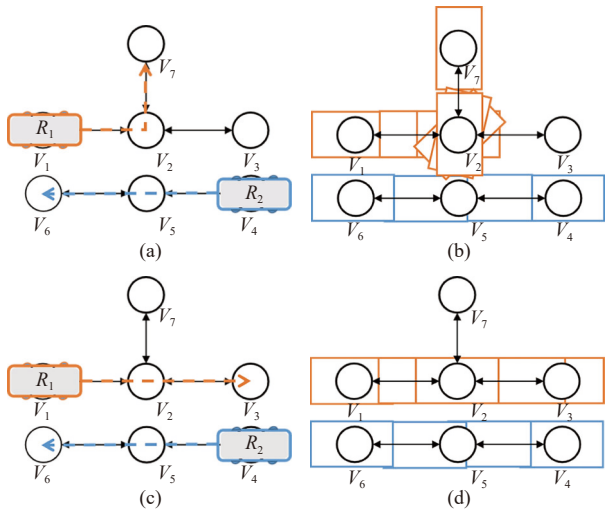

Fig. 6. An example showing the characteristics of glued nodes. (a) The paths of the two robots contain a pair of glued nodes: $GN_{2,5}^{1,2} = 1$. (b) The area swept by the two robots when executing Γ_1^2 and Γ_2^5. (c) The paths of the two robots do not contain glued nodes. (d) The area swept by the two robots when executing Γ_1^2 and Γ_2^5.

D. Assumption

The stable operation of multi-robot systems is supported by a variety of technologies, such as navigation algorithm, communication technology, sensor technology, motion control, etc. In this letter, we mainly study collision and deadlock avoidance in multi-robot systems. For convenience, we provide some assumptions. Nevertheless, these assumptions do not reduce the contribution of our work.

1) The communication between the robots and the control center is based on a wireless network, and there is no delay, error and packet loss in the communication.

2) Each robot can always move along the planned path within an acceptable derivation.

3) Taking into account the positioning accuracy, robots can only stop 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 to give way to the robot with a path.

5) The robots are all in a unified global coordinate system, and the expressions of positions and angles are consistent.

E. Example for Algorithm 1

We take Fig. 7 as an example to explain the collision avoidance algorithm. As shown in Fig. 7, $P_1 = \{V_1, E_{1,2}, V_2, E_{2,3}, V_3, E_{3,4}, V_4, E_{4,8}, V_8\}$, $P_2 = \{V_5, E_{5,4}, V_4, E_{4,3}, V_3, E_{3,7}, V_7\}$, $OV_1 = \{V_1, V_2\}$, $OV_2 = \{V_5\}$, $AV_1 = \{V_3, V_4, V_8\}$ and $AV_2 = \{V_4, V_3, V_7\}$. $AV_1 \cap AV_2 \neq \emptyset$ means R_1 and R_2 apply for the same nodes. However, which robot the nodes are authorized to can be judged by factors such as task priority and traffic congestion. This paper uses a simple strategy of applying first, occupying first. 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 can not be authorized for the reason of $GN_{2,3}^{1,2} = 1$. So the result is that only V_4 is authorized to R_2. It is worth noting that this result will cause a deadlock between R_1 and R_2. When R_2 applies the node V_4, V_4 will be authorized to R_2 for it will not cause any collisions. But V_3 can not be authorized to R_2 because $GN_{2,3}^{1,2} = 1$, i.e., R_2 is blocked by R_1. And, if V_3 is authorized to R_1, then R_1 will move to V_3, a deadlock formed. Actually, once V_4 is authorized to R_2, R_1 and R_2 will inevitably form a deadlock.

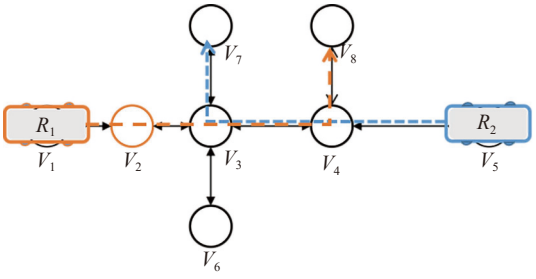

Fig. 7. An example of collision avoidance, with $GN_{2,3}^{1,2} = 1$.

F. Example for Algorithm 2

In Fig. 7, V_2, V_3 and V_4 are conflict nodes for R_1 and R_2, and the conflict nodes compose a conflict area $\Phi_{1,2} = \{V_2, V_3, V_4\}$. Since V_2 is authorized to R_1, there is $R_1 \rightarrow \Phi_{1,2}$. The conflict area is essentially the common area swept by two robots in the local space when they move along the paths. It is possible for a robot to stop in

such a conflict area to block another robot. The essence of deadlock is actually a cyclic block formed among robots, as shown in Fig. 8, there are two examples of deadlock. Once a robot occupies a node, it can move to the node autonomously. If a robot move to the furthest occupied node, no conflict circle will be formed, then it is impossible to form a cyclic block among the robot and other robots.

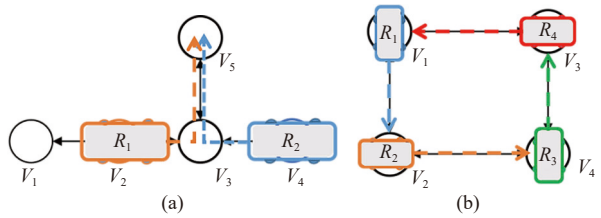

Fig. 8. Examples of deadlocks. (a) A deadlock of two robots. (b) A deadlock of four robots.

Fig. 9 shows the collision and deadlock avoidance process of four robots. Initially, as shown in Fig. 9(a), 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 R_3 first applies for node V_3 and get the 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 $\langle R_2\rightarrow\Phi_{1,2}, R_3\rightarrow\Phi_{2,3}, R_4\rightarrow\Phi_{3,4}, R_1\rightarrow\Phi_{4,1}\rangle$ will be generated. Therefore, the Algorithm 2 will prevent V_6 from being authorized to R_2, see Fig. 9(b). And then, as shown in Figs. 9(c) and 9(d), R_3 continues to move forward, and the four robots will eventually reach their respective destinations.

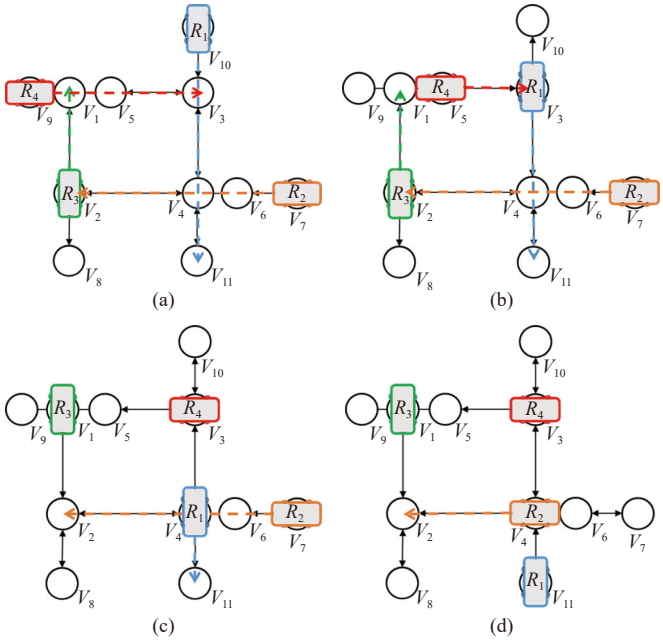

Fig. 9. An example of collision and deadlock avoidance for four robots. (a) Area occupations: ($R_4\rightarrow\Phi_{3,4}$), glued nodes: ($GN_{1,9}^{3,4}=1$, $GN_{1,5}^{3,4}=1$, $GN_{4,6}^{1,2}=1$). (b) Area occupations: ($R_4\rightarrow\Phi_{3,4}$, $R_1\rightarrow\Phi_{4,1}$, $R_3\rightarrow\Phi_{2,3}$), glued nodes: ($GN_{1,5}^{3,4}=1$, $GN_{4,6}^{1,2}=1$). (c) Area occupations: ($R_1\rightarrow\Phi_{1,2}$). (d) No area occupations and no glued nodes.

G. Further Explanation of the Experiments

Figs. 10–12 are graphical representations of the simulation results in the letter (Table 2 in the letter). Fig. 2 in the letter is a screenshot of the simulation software. The experiments are based on a real automated warehouse scenario. Warehouse management system

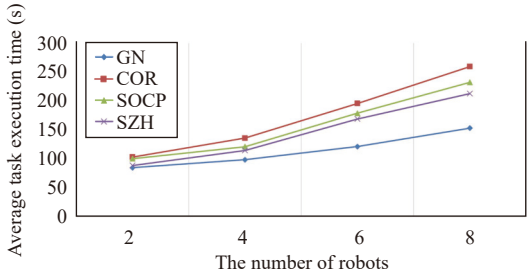

Fig. 10. Simulation result of the average task execution time.

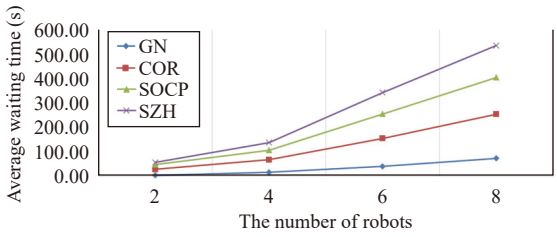

Fig. 11. Simulation result of the average waiting time.

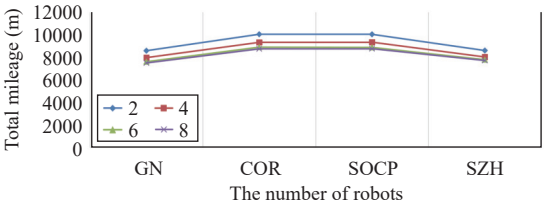

Fig. 12. Simulation result of the total mileage.

(WMS) generates two types of tasks: outbound tasks and inbound tasks. Each task includes a pickup point and a drop-off point, and will be assigned to the nearest idle robot. Each robot can only execute one task at a time. When a robot executes an inbound task, the robot moves to a work station to pick up a cargo and deliver it to a storage point. When a robot executes an outbound task, the robot moves to a storage point to pick up a cargo and deliver it to a work station. Those robots with no tasks will move to the robot stations and wait for new tasks. Whenever the battery level of robots drops below a set value (15% in the experiments), the robots will move to charging stations to charge.

This letter uses the average task execution time, the average waiting time of robots, and the total mileage of robots to evaluate the performance of different methods. Let $\mathbb{K}=\{1,2,...,K\}$ be the index of tasks, K is the number of tasks. TK_i, $i\in\mathbb{K}$ denotes a task in the system. The average task execution time is calculated as

$$\sum_{i=1}^{K}\frac{(T_e^i-T_s^i)}{K} \tag{1}$$

where T_s^i and T_e^i denote the start time and end time of TK_i, respectively. The average waiting time is calculated as

$$\sum_{i=1}^{K}\sum_{j=1}^{N}\frac{T_w^{i,j}}{K} \tag{2}$$

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

$$TM=\sum_{i=1}^{K}\sum_{j=1}^{N}M_{i,j} \tag{3}$$

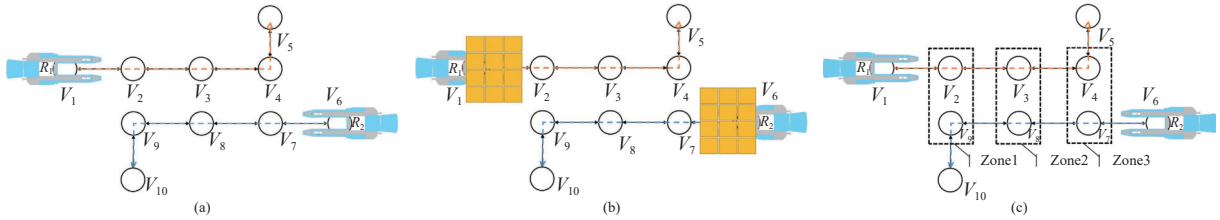

Fig. 13. An experimental scene with two automatic forklifts. (a) There are no glued nodes when the two robots are not loaded. (b) When the two robots are loaded, there are three pairs of glued nodes: $GN_{2,9}^{1,2} = 1$, $GN_{3,8}^{1,2} = 1$, $GN_{4,7}^{1,2} = 1$. (c) Based on the principle of zone control, V_2, V_3, V_4, V_7, V_8 and V_9 belong to three zones.

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, $M_{i,j} = 0$.

In some simple scenarios, if the layout of the roadmap is compact, the GN method also has obvious advantages. The methods based on zone control must first divide the environment, after the division, each zone does not change, making the method too imprecise. For example, in a compact roadmap shown in Fig. 13(a), using GN, when both robots are not loaded, there are no glued nodes, and both robots can pass normally. Only when the two robots are loaded, the glued nodes make it impossible for both robots to pass through at the same time (Fig. 13(b)). However, based on the principle of zone control, there are three zones shown in Fig. 13(c). This means that the two robots cannot pass through this channel at the same time under any circumstances.

In fact, based on the traffic network in [4] (SOCP), as shown in Fig. 14, the nodes of the traffic network are collision-free (Each node can represent a zone), and the dynamics of the glued nodes cannot be reflected. Even in this scenario our method is still more efficient. Supplementary experimental results are shown in Figs. 15 and 16. It can be seen that GN requires fewer steps than SOCP. This not only shows the adaptability of GN, but also proves its high efficiency in various scenarios.

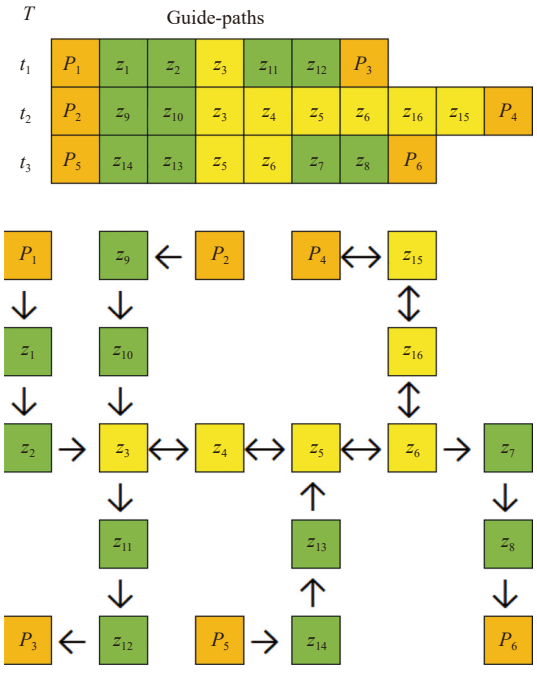

Fig. 14. An experimental map in [4].

ST	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}	Z_{11}	Z_{12}	Z_{13}	Z_{14}	Z_{15}	Z_{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	V_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	V_1	0	0	0	0	0	0	V_2	0	0	0	0	V_3	0	0
4	0	0	V_1	0	V_3	0	0	0	0	V_2	0	0	V_3	0	0	0
5	0	0	0	0	0	V_3	0	0	0	V_2	0	0	0	0	0	0
6	0	0	0	0	0	0	V_3	0	0	V_2	V_1	0	0	0	0	0
7	0	0	0	V_2	0	0	0	V_3	0	V_2	0	V_1	0	0	0	0
8	0	0	0	0	V_2	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	V_2	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	V_2	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	V_2
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	V_2	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 15. Zone states and process reservations for the system presented in Fig. 14 when using SOCP.

ST	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}	Z_{11}	Z_{12}	Z_{13}	Z_{14}	Z_{15}	Z_{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	V_1	0	0	0	0	0	0	0	V_2	0	0	0	0	V_3	0	0
2	0	V_1	0	0	0	0	0	0	V_2	0	0	0	V_3	0	0	0
3	0	0	V_1	0	V_3	0	0	0	V_2	0	0	0	0	0	0	0
4	0	0	0	0	0	V_3	0	0	V_2	V_1	0	0	0	0	0	0
5	0	0	V_2	0	0	0	V_3	0	0	0	V_1	0	0	0	0	0
6	0	0	0	V_2	0	0	0	V_3	0	0	0	0	0	0	0	0
7	0	0	0	0	V_2	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	V_2	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	V_2
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	V_2	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 16. Zone states and process reservations for the system presented in Fig. 14 when using GN.

References

[1] A. Willms and S. Yang, "An efficient dynamic system for real-time robot-path planning," *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 4, pp. 755–766, 2006.

[2] A. Guruji, H. Agarwal, and D. Parsediya, "Time-efficient a* algorithm for robot path planning," *Procedia Technology*, vol. 23, pp. 144–149, 12, 2016.

[3] S. Gottschalk, "Separating axis theorem," Technical Report TR96-024, Dept. of Computer Science, UNC Chapel Hill, USA, 1996.

[4] J. Zajac and W. Malopolski, "Structural on-line control policy for collision and deadlock resolution in multi-AGV systems," *Journal of Manufacturing Systems*, vol. 60, pp. 80–92, 2021.