Routing of a multi-robot system using a Time-Space Network model

Meng, Chuang; Xin, Jianbin; Peng, Jinzhu; Negenborn, Rudy R.

DOI
10.1109/CAC48633.2019.8996444

Publication date
2019

Document Version
Accepted author manuscript

Published in
Proceedings - 2019 Chinese Automation Congress (CAC 2019)

Routing of a Multi-Robot System Using a Time-Space Network Model

Chuang Meng, Jianbin Xin, Jinzhu Peng
School of Electrical Engineering
Zhengzhou University
Zhengzhou, China
zzumengchuang@163.com
j.xin@zzu.edu.cn
jzpeng@zzu.edu.cn

Rudy R. Negenborn
Department of Maritime and Transport Technology
Delft University of Technology
Delft, The Netherlands
r.r.negenborn@tudelft.nl

***Abstract*—In this paper, we study the collision-free routing of a multi-robot system to complete given tasks in the shortest time. In a robotic assembly unit, several stations work serially and in parallel. In a station, multiple robots share the same workspace and face the challenge of minimizing the cycle time and avoiding collisions at the same time. For this problem, we propose a new mathematical model that is the so-called time-space network (TSN) model. The TSN model can map the robot location constraints into the routing planning framework, leading a mixed integer programming problem. By solving this mixed integer programming problem, the collision-free path of multiple robots can be obtained. Finally, The simulation results illustrate the proposed TSN model can obtain the collision-free route of the multi-robot system.**

***Index Terms*—Multi-robot systems, routing, time-space network model, collision avoidance**

I. Introduction

Manufacturing systems used in modern industrial production are very complicated, consisting of hundreds of industrial machines and robots working in production lines and stations for processing tasks. For the producer, the expectations on the returns of the large investment impose that as many products are produced using available machinery in a given period as possible. To meet this requirement, it is necessary to increase productivity.

In manufacturing systems, robotic assembly cells which consist of several stations working serially and in parallel are crucial to reach productivity goals. In a robotic assembly cell, multiple robots work in the same workspace of the station to carry out complex tasks together within a predetermined period. High production rates require full use of limited equipment of the station to increase the production per unit time and shorten the production time. Usually, resource optimization plays an important role in saving energy and reducing workspace, both economically and ecologically, when considering sustainable production systems, see [1]. Therefore, optimizing multi-robot systems of the station is an effective way of increasing productivity.

In this work, we focus on optimizing the operation process of the multi-robot system in the station. We in particular focus on manipulators. Multiple robots often share the same workspace of the station and work cooperatively to operate studs, welding, and sealing. The number of products can be maximized by minimizing the time needed at each station to perform predefined tasks in a collision-free way. Therefore, collision-free routing for the robot of the station to complete given tasks in the shortest time can greatly improve productivity.

However, the path of a robot may intersect the path of another robot at the same time while performing tasks, which means that each robot becomes a dynamic obstacle. This requires us not only to assign tasks for each robot to improve productivity but also to coordinate the path to avoid collisions between robots. Currently, the collision-free routing for robots is only considered in static environments in which the shape and position of obstacles never change. In many practical application scenarios, only considering collision-free routing in a static environment is not enough. As described in [2], multiple robots generally work together in a dynamic environment in which the robots whose positions change need to be regarded as dynamic obstacles. In this case, collisions may occur between robots in the station, which cannot be solved using routing methods that only consider static environments.

Collision-free routing for multi-robot systems is complex. The multi-robot system is different from a single robot system which focuses on the collision-free path planning in a fixed static environment in which the shape and position of obstacles never change. Compared with a single-robot system, a multi-robot system has a more complex model, see [3], [4]. For the multi-robot system, the problem can usually be modeled as a vehicle routing problem (VRP) with conflicts, which is a general case of the multiple traveling salesman problem(MTSP); For a detailed introduction to VRP and its variants, see [5]. Furthermore, several articles have dealt with path planning for mobile and industrial robots, see [6] and [7]. Here, we combine the task assignment and path coordination problems of the multi-robot system.

Planning of multi-robot systems has two aspects: the first is task allocation, and the second is collision-free routing among multiple robots. There are several pieces of research on how to assign tasks in a multi-robot system to achieve the optimal

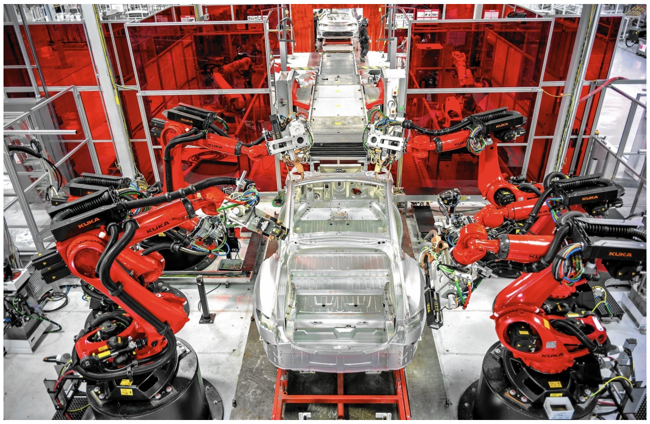

Fig. 1. A typical robotic assembly cell with stud welding robots.

utilization of resources [8]–[10]. However, the main problem of multi-robot systems is the conflict between multiple robots, e.g. regarding space usage. Therefore, how to coordinate the path to avoid collision between robots is the main problem to be considered.

In the last years, the problem of multi-robot collision-free routing has been treated. In [11], the collision-free path coordination of two robots is realized by the c-space method. In [12], mixed integer programming and standard solvers are used to solve the problem, and tasks are divided into different groups and subgroups. It is assumed that the motion time of robots between groups is constant, and conflicts can be avoided by prohibiting specific robot groups and task groups. In [13], the method of separating task allocation and path coordination is adopted, and decoupling is achieved by determining the priority of the delayed start of the manipulator and modifying the path of the manipulator. The iterative decoupling method is proposed in [14], which is to establish the generalized traveling salesman problem for the assigned task, and solve it repeatedly and iteratively through the objective function with no conflict and optimal time until a reliable solution is obtained. In [15], two methods were proposed for the collision-free path planning of multiple manipulators: the concentrating method and the decoupling method. In [16], a new probabilistic roadmap algorithm was proposed, which solves the practical problems in path planning by using the single-query bidirectional sampling strategy of delayed collision detection. But We observe that the collisions between these robots can take place at any time instant when planning their routes. Coordinating conflict paths could be cumbersome as the collision-free routing problem is not solved in single decision space. Therefore, it is desired to incorporate collision constraints into the routing problem and formulate the related problem as a single optimization problem. In this paper, we will address this collision-free routing problem.

This paper proposes a new methodology for collision-free routing of a multi-robot system to complete given tasks. In this paper, considering the relationship between task assignment and robot path planning, we construct a time-space network (TSN) model using mixed integer programming (MIP), which can describe the motion path of multiple robots. Under the premise of the synchronous starting of the multi-robot system, the space position of each robot for each unit time window can be obtained based on the TSN model. Based on the spatial position of each robot obtained by the TSN model, coordinate constraints are added to make each robot work within a safe working range to avoid collisions. Using the TSN model, different cases are studied. In these cases, robots can wait for multiple unit time windows or move to other safe positions to realize collision avoidance. The simulations demonstrate the flexibility and availability of the TSN model in collision-free routing for the multi-robot system.

The remainder of this paper is organized as follows: Section II provides a description of the problem and the proposed mathematical model. Section III provides simulation experiments. Section IV concludes the work and gives suggestions for future work.

II. Modeling

The model proposed in this paper can be applied to different multi-robot assembly units, differing in both type and number of robots and tasks. Examples of operations we have considered are stud-welding and spot-welding, typically on a car body. In this paper, We focus on the machining process of the car body-in-white. At each station, multiple robots are used to weld the car body, as visualized in Fig.1.

A. Problem Description

In this paper, we study the collision-free routing of a multi-robot system to complete given tasks in a dynamic environment. In this case, robots share the workspace of the station and may occur collision. Therefore, we need to coordinate the path of the robots to realize the collision-free between robots.

The problems now faced are as follows:

- Each task can only be assigned to one robot for executing;
- All robots start simultaneously to complete all tasks in the shortest time;
- Plan the collision-free path of robots;

In order to more vividly describe the dynamic obstacles between the robots mentioned above, consider Fig.2. It describes a basic situation; both robots R1 and R2 have a home position (robots start from the home position and finally return to the home position), and when each robot is assigned at least one task, there will be overlapping of task areas. Each robot poses a dynamic obstacle to other robots when performing tasks.

Here, several important assumptions are given as follows:

- The robot takes up a certain amount of space when performing tasks.
- The number of robots, the number of tasks, and the time of the tasks are all given in advance, and the task time is an integer multiple of the unit time.
- Each robot starts the next task after completing the current task.

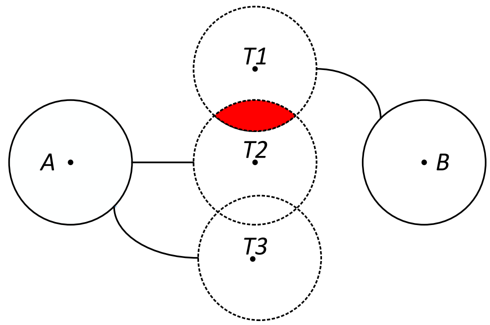

Fig. 2. A and B are the main positions of the two robots respectively. When the robot starts to perform the task, the shadow area is the conflicting area.

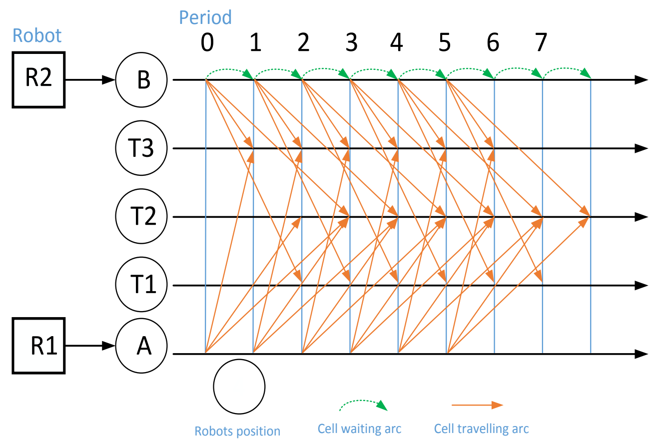

Fig. 3. Illustration of the time-space network model

B. Problem modeling

In this section, we propose a new methodology using the TSN model. Before we derive the model, we illustrate how the time-space network can be generated from the physical network. Fig.2 shows a basic physical network, A and B are the positions of the robots R1 and R2, respectively, and T1, T2, and T3 are the positions of the tasks. The time-space graph is illustrated in Fig.3, and has been constructed as follows: Define one node for each cycle (one-time unit), with different nodes representing different locations. Each node is linked by arcs to the corresponding adjacent nodes. Each arc corresponds to the movement of the robot between two endpoints of the segments of the path. If an arc joins the same node in an adjacent period (ie, a horizontal arc), it means that it waits for a time unit at that node. All arcs have at least one unit time length. Each robot has an initial node in period 0. The method can decompose the whole moving process of the robots into several unit time windows, and obtain the motion trajectory of each robot. As shown in Fig.4, the path of the robot is decomposed into three time slots (start at the 11th unit time window and end at the 14th time window). According to the trajectory of each robot on the unit time window, we can add the collision avoidance constraint on each robot, so that each robot can access specific nodes at specific times to achieve effective collision avoidance.

The subscripts involved in the model, input parameters and decision variables are shown in Table I, Table II, and Table III, respectively.

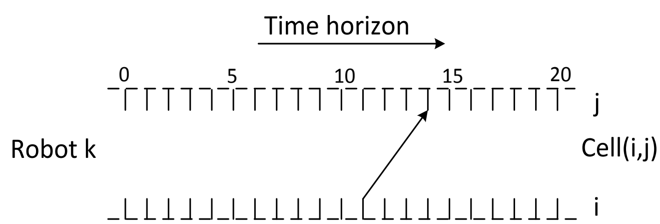

Fig. 4. Robot k arrives in the 11th unit time window and then leaves in the 14th time window.

TABLE I
INDEX LABEL.

Variable	Description
C	The number of tasks
K	The number of robots
N	The collection of all nodes
T	Total time period
i, j, n	Network node, $i, j, n \in N$
k	Robot label, $k \in K$
t	time index, $t \in T$

TABLE II
INPUT PARAMETERS.

Variable	Description
O_k	Robot k starting position
S_k	Robot k target position
$E^O(i)$	The section starting from node i
$E^S(i)$	The section to the end of node i
E_k	The set of arcs that robot k may use
V_k	The set of nodes assigned to robot k
$W_{i,j}$	Time spent on the road segment ij
$D_{i,j}$	Distance between road segments ij
Y_{ik}	Assign node i to robot k, $Y_{ik} = 1$ if $i \in V_k$
R	Robot's safe range of activities

TABLE III
DECISION VARIABLES.

Variable	Description
X_{ijk}	0-1 variable, $Xijk$ is 1 means that the arm k passes the link ij,otherwise it is 0
A_{ijkt}	0-1 variable, the robot k is 1 when it reaches the arc ij at time t, otherwise it is 0.
D_{ijkt}	0-1 variable, robot k is 1 when leaving arc ij at time t, otherwise 0
Y_{ijkt}	0-1 variable, robot k is 1 when occupying arc ij at time t, otherwise 0
Y_{ik}	0-1 variable, the robot k is 1 when passing the point i, otherwise it is 0
C_{ipqt}	0-1 variable, collision avoidance coefficient between robots p and q at time t, $p, q \in K$
X_{kt}	Real variable, the abscissa of robot k at time t
Y_{kt}	Real variable, the abscissa of robot k at time t

The objective function Z represents minimization of the maximum time for each robot to complete the task.

$$Z = min\{max\{\sum_t t \times \sum_{i;(i,S_k \in E^s(S_k) \cap E_k} [D_{ijkt} - D_{ijk(t-1)}],\ \forall k\}\} \quad (1)$$

Subject to:
Group 1: Space constraint:

$$\sum_{i.j:(i.j) \in E^O(O_k) \cap E_k} X_{ijk} \geq 1,\ \forall k \quad (2)$$

$$\sum_{i.j:(i.j) \in E^S(S_k) \cap E_k} X_{ijk} \geq 1,\ \forall k \quad (3)$$

$$\sum_{i:(i.j) \in E^S(j) \cap E_k} X_{ijk} = \sum_{n:(j.n) \in E^O(j) \cap E_k} X_{jnk} \quad \forall k,\ j \in N - O_k - S_k \quad (4)$$

$$\sum_{k=1} \sum_{j=1} X_{ijk} \geq 1,\ \forall i \quad (5)$$

$$X_{ijk} \leq Y_{ik},\ \forall i, j, k \quad (6)$$

$$X_{jik} \leq Y_{ik},\ \forall i, j, k \quad (7)$$

Group 2: Time constraint:

$$W_{ij} \times X_{ijk} = \sum_t (t * (D_{ijkt} - D_{ijk(t-1)})) - \sum_t (t * (A_{ijkt} - A_{ijk(t-1)})), \forall k, (i,j) \in E_k, i \neq j \quad (8)$$

$$W_{ij} \times X_{ijk} \leq \sum_t (t * (D_{ijkt} - D_{ijk(t-1)})) - \sum_t (t * (A_{ijkt} - A_{ijk(t-1)})), \forall k, (i,j) \in E_k, i = j \quad (9)$$

$$A_{ijkt} \geq A_{ijk(t-1)},\ \forall k, (i,j) \in E_k, t \quad (10)$$

$$D_{ijkt} \geq D_{ijk(t-1)},\ \forall k, (i,j) \in E_k, t \quad (11)$$

$$Y_{ijkt} = A_{ijkt} - D_{ijkt},\ \forall k, (i,j) \in E_k, t \quad (12)$$

$$\sum_{i.j:(i,j \in E_k)} Y_{ijkt} \leq 1,\ \forall k, t \quad (13)$$

Group 3: Time-space constraint

$$\sum_{i:(i,S_k) \in E_k} D_{iS_k kT} \geq 1,\ \forall k \quad (14)$$

$$\sum_{j:(O_k,j) \in E_k} A_{O_k jkt} = 1,\ \forall k, t = E_k \quad (15)$$

$$X_{ijk} = A_{ijkT},\ \forall k, (i,j) \in E_k \quad (16)$$

$$\sum_{i.j:(i,j) \in E_k} D_{ijkt} = \sum_{j.n:(j,n) \in E_k} A_{jnkt} \quad \forall k, j \in N - O_k - S_k \quad (17)$$

$$D_{ijkt} \leq A_{ijkt},\ \forall k, (i,j) \in E_k, t \quad (18)$$

Group 4: Collision avoidance constraint:

$$X_{qt} - X_{pt} \geq R - MC_{1pqt},\ \forall p, q \in K, t \quad (19)$$

$$X_{pt} - X_{qt} \geq R - MC_{2pqt},\ \forall p, q \in K, t \quad (20)$$

$$Y_{qt} - Y_{pt} \geq R - MC_{3pqt},\ \forall p, q \in K, t \quad (21)$$

$$Y_{pt} - Y_{qt} \geq R - MC_{4pqt},\ \forall p, q \in K, t \quad (22)$$

$$\sum_{i=1}^{4} C_{ipqt} \leq 3,\ \forall p, q \in K, t \quad (23)$$

Equation (1) is the objective function, representing minimization of the maximum time for each robot to complete the task; In Group 1: (2)-(4) are constraints on the path order of the starting position, intermediate position and ending position of each robot; Equation (5) is a constraint that guarantees that all tasks are executed; Equations (6) and (7) ensure that different tasks correspond to different robots.

In Group 2: Equations (8) and (9) are time constraints for each robot to perform the task, and each robot can wait more than one unit time in place; Equations (10) and (11) are the time connectivity constraint for each robot to perform tasks; Equations (12) and (13) guarantee that each robot can only perform one task per unit time.

In Group 3: Equation (14) and (15) are the time constraints of each robot at the start position and the end position, ensuring that each robot moves at the set start time; Equation (16) is the mapping constraint between the time-space network and the physical network; Equations (17) and (18) guarantee the continuity of each robot arrival time and departure time.

In Group 4: Equations (19)-(23) guarantee each robot collision-free movement, where M is a great positive number. The coordinates of robot k_1 $(X_{k_1 t}, Y_{k_1 t})$ and k_2 $(X_{k_2 t}, Y_{k_2 t})$ are the coordinates of robot k_1 and k_2 at time t.

III. Case studies

In this section, we give the detailed configuration and parameters of the multi-robot assembly unit. The performance of the proposed time-space network model is analyzed by two examples. One experiment is a general test case to verify the performance of the model in the normal scene, and the other is an industrial test case to verify the actual effect of the model application.

A. Setting of the multi-robot simulation system

The experiment was carried out using MatlabR2014a on Windows 8, 64-bit operating system, and the Yalmip toolbox (version: R20180612) and Cplex solver (version: 12.6), see [17] and [18]. The computer hardware configuration used in the experiment involved a CPU model Intel Core i5-4200U, clocked at 1.62HZ with 4GB of memory.

Here, the robot we used for the simulation was ABB's IRB2400 series robot. This robot moves horizontally and vertically within the working range centered on itself. For the sake of simplicity, the shape of the robot is assumed to be circular, the diameter R of which is the safe working range. The specific parameters are shown in Table IV. In addition, the length of a time window in TSN model can be a multiple

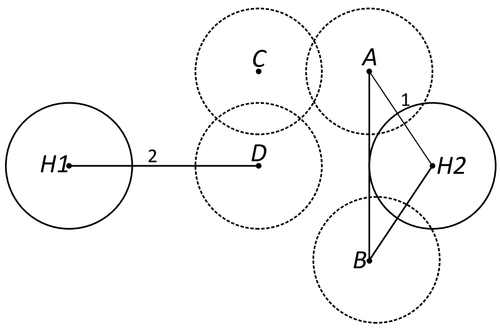

Fig. 5. Geometrical overview of two robots, the position distribution of robots and tasks.

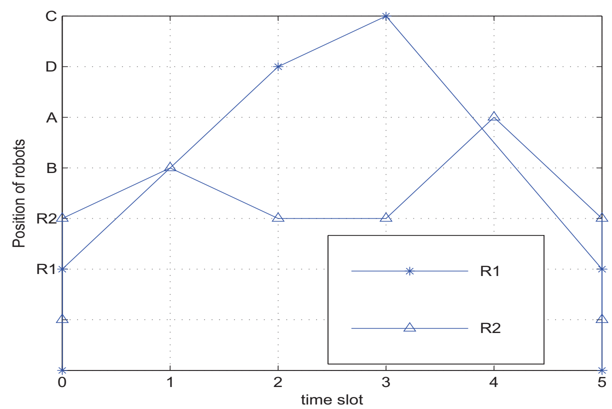

Fig. 6. The trajectories of two robots based on the time-space network model.

of the unit time (for instance, one unit time window means 2s or 4s or 8s, etc).

TABLE IV
KEY PARAMETERS OF THE MULTI-ROBOT WORKSTATION.

Parameter	Description	Value	Unit
R	Safe distance between each two robots	5	[cm]
WR	Working range of each robot	150	[cm]

B. Simulation results

As shown in Fig.5: there are two robots with circular shape at their home position H_1 and H_2, respectively. The robots have overlapping task areas in the four task points A, B, C, and D, and in the path in which the tasks are executed, conflicts can occur.

Fig.6 shows the detailed path of each robot obtained by optimizing with the time-space network model proposed in this paper. The determined visited node sequences are as follows: R_1:$H_1 \to D \to C \to H_1$; R_2:$H_2 \to B \to H_2 \to A \to H_2$. When robot R1 sequentially executes the tasks D and C, robot R2 selects to execute the task A first, then returns to the main position and then executes task B, avoiding the conflict with the robot R1. Therefore the collision-free paths for multiple robots can be obtained by solving this model.

The advantage of this model is that each point can be accessed not only by the robot many times, but also the robot can wait for multiple unit time at each point. Therefore, there can be a more flexible collision avoidance strategy in a complex industrial environment.

To deal with more practical examples of problems, we present the results of an industrial test case. As shown in Fig.7, it consists of 2 industrial robots performing stud welding operations and 10 predefined tasks. Fig.8 illustrates the visited node sequences of these two robots obtained using the TSN model. Table V records the computation time for different cases. When the number of robots is fixed, the computation time increases as the number of tasks increases.

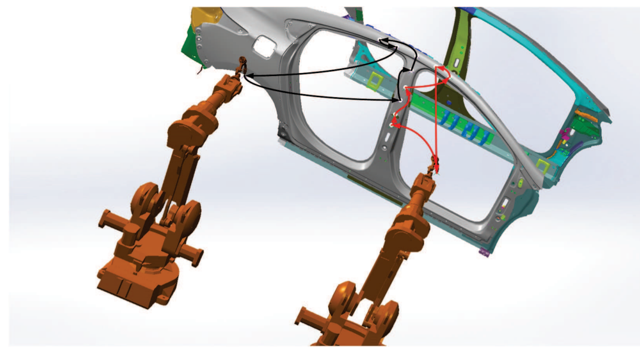

Fig. 7. Two robots with their home positions and 10 stud weld tasks.

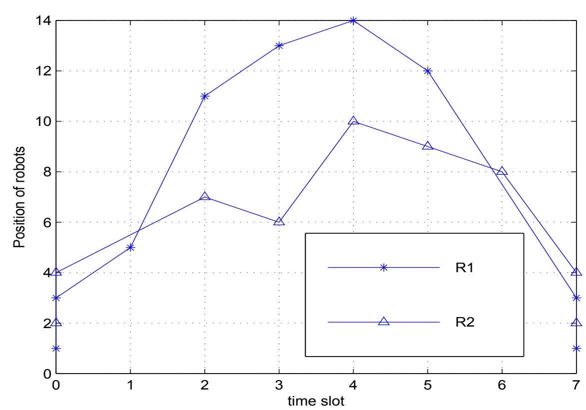

Fig. 8. The trajectories of two robots based on the time-space network model.

TABLE V
COMPUTER TIME FOR SEVERAL CASES.

Robots	Tasks	Case1	Case2	Case3	Case4
2	4	18.2397s	18.4105s	18.3109s	18.2942s
2	10	61.0793s	59.9914s	60.2512s	61.1480s

IV. Conclusions and Future Research

This paper studies and analyzes the collision-free routing problem of industrial robots at workstations. A routing method based on the time-space network model is proposed. After analyzing the actual situation at a work station, we make a reasonable assumption for the industrial robot system and establish the geometric model of the problem. Moreover, a time-space network model is established, which combines time and space constraints to propose collision avoidance constraints. Simulations of small case studies were carried out to illustrate the feasibility of the model.

Further research will consider solving a large-scale multi-robot system, and an efficient computational method is needed to be developed.

Acknowledgements

This research is supported by the National Natural Science Foundation of China under Grant 61703372, the Key Scientific Research Project of Henan Higher Education under Grant 18A413012 and the Science &Technology Innovation Team Project of Henan Province under Grant 17IRTSTHN013

References

[1] P. Almström, C. Andersson, A. Muhammad, and M. Winroth, "Achieving sustainable production through increased utilization of production resources," in *Proceedings of the 4th Swedish Production Symposium, Lund, May 3-5, 2011*, 2011.

[2] T. Arai, E. Pagello, and L. E. Parker, "Advances in multi-robot systems," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 655–661, 2002.

[3] M. D. Zivanovic and M. Vukobratovic, *Multi-arm cooperating robots: dynamics and control*. Springer Science & Business Media, 2006, vol. 30, pp. 189-236.

[4] Y. Gan and X. Dai, "Survey of coordinated multiple manipulators control," *Journal of Control and Decision*, vol. 28, no. 3, pp. 321–333, 2013.

[5] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, ser. MOS-SIAM Series on Optimization. SIAM, 2014, no. 18.

[6] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.

[7] J.-C. Latombe, *Robot Motion Planning*. Springer Science & Business Media, 2012, vol. 124.

[8] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," in *Proceedings of the Cooperative Robots and Sensor Networks 2015*. Springer, 2015, pp. 31–51.

[9] C. Liu and A. Kroll, "Memetic algorithms for optimal task allocation in multi-robot systems for inspection problems with cooperative tasks," *Soft Computing*, vol. 19, no. 3, pp. 567–584, 2015.

[10] S. Öztürk and A. E. Kuzucuoğlu, "Optimal bid valuation using path finding for multi-robot task allocation," *Journal of Intelligent Manufacturing*, vol. 26, no. 5, pp. 1049–1062, 2015.

[11] B. Tang, X. Wang, L. Xue, and X. Gu, "Dual-welding robots collision-free path planning," *Journal of East China University of Science and Technology: Natural Science Edition*, vol. 43, no. 3, pp. 417–424, 2017.

[12] T. C. Lopes, C. G. Sikora, R. G. Molina, D. Schibelbain, L. C. A. Rodrigues, and L. Magatão, "Balancing a robotic spot welding manufacturing line: An industrial case study," *European Journal of Operational Research*, vol. 263, no. 3, pp. 1033–1048, 2017.

[13] S. S. Chiddarwar and N. R. Babu, "Conflict free coordinated path planning for multiple robots using a dynamic path modification sequence," *Robotics & Autonomous Systems*, vol. 59, no. 7, pp. 508–518, 2011.

[14] D. Spensieri, J. S. Carlson, F. Ekstedt, and R. Bohlin, "An iterative approach for collision free routing and scheduling in multirobot stations," *IEEE Transactions on Automation science and Engineering*, vol. 13, no. 2, pp. 950–962, 2016.

[15] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 635–646, 2010.

[16] G. Sánchez and J.-C. Latombe, "On delaying collision checking in prm planning: Application to multi-robot coordination," *International Journal of Robotics Research*, vol. 21, no. 1, pp. 5–26, 2002.

[17] CPLEX, "CPELX 12.6 optimization studio," *IBM, New York, NY, USA*, 2014.

[18] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in matlab," in *Proceedings of the Computer Aided Control System Design Conference*, Taipei, Taiwan, 2004, pp. 1-6.