A Time-Space Network Model for Collision-Free Routing of Planar Motions in a Multirobot Station

Jianbin Xin, Chuang Meng, Frederik Schulte, Jinzhu Peng, Yanhong Liu, and Rudy R. Negenborn

***Abstract*—This article investigates a new collision-free routing problem of a multirobot system. The objective is to minimize the cycle time of operation tasks for each robot while avoiding collisions. The focus is set on the operation of the end-effector and its connected joint, and the operation is projected onto a circular area on the plane. We propose to employ a time-space network (TSN) model that maps the robot location constraints into the route planning framework, leading to a mixed integer programming (MIP) problem. A dedicated genetic algorithm is proposed for solving this MIP problem and a new encoding scheme is designed to fit the TSN formulation. Simulation experiments indicate that the proposed model can obtain the collision-free route of the considered multirobot system. Simulation results also show that the proposed genetic algorithm can provide fast and high-quality solutions, compared to two state-of-the-art commercial solvers and a practical approach.**

***Index Terms*—Collision avoidance, multirobot systems, routing, time-space network model.**

I. Introduction

ROBOTS significantly improve productivity in the manufacturing environment. It is not uncommon that hundreds of industrial machines and robots work in production lines and stations for processing different types of tasks. For the producer, expectations on returns of the large investment impose that as many products as possible are produced using available machinery in a given time period. In order to meet this requirement, it is necessary to maximize the productivity.

In manufacturing systems, a robotic assembly line consists of several stations that are placed serially or in parallel to reach production goals [1]. At each station, multiple robots, often sharing the same workspace, work together to carry out complex operations (e.g., stud, welding, or sealing) within a predetermined period [2]. High production rates require optimizing the operation and coordination within multirobot systems of the assemble station to increase productivity. Most works on collision avoidance in multirobot systems develop decentralized approaches [3], [4]. While well-suited for highly unstructured environments, they generally show a performance gap in comparison to centralized approaches [4].

In this article, the operational process of the multirobot system in an assembly station is optimized. In the station, a robot is referred to as a manipulator. The total number of tasks assigned at a station can be obtained by solving an assembly line balancing problem [1]. To complete the assigned tasks together, the station operation process needs to be optimized. Yet, optimizing this process is a complicated problem and the related decision problems are divided into several subproblems at different levels: task allocation, routing, and path planning [2]. These subproblems are typically regarded as operational scheduling and control problems. Here, we focus on integrated routing and path planning of the multirobot system in an assembly station. A time-space network model is proposed and a dedicated algorithm is developed for this integrated problem.

A. Related Work

Planning problems concerning multiple robots can be categorized into three types of problems in a hierarchical way: task allocation, routing, and path planning. At a higher level, task allocation focuses on optimally dispatching tasks to the available robots [5]–[7]. Routing problems typically determine task sequences and the robot route for performing several tasks to be given [2], [8]. Path planning problems compute more detailed paths in the working space subject to collision-aware constraints or physical constraints on the robot [9]. In path planning (also known as motion planning), a complete path is concerned with performing a single task of the robot from an initial point to an ending point. For routing problems and path planning problems, if the task involves a time-based objective, the associated problem also is regarded as a scheduling problem.

In manufacturing systems, task allocation of multiple robots has received extensive attention. In [10], an intersection-free geometrical partitioning method is proposed for allocating tasks of assembly cells to minimize the cycle time. Moreover, task allocation is also considered with layout design simultaneously

Manuscript received August 6, 2019; revised December 9, 2019 and January 7, 2020; accepted January 14, 2020. Date of publication January 22, 2020; date of current version June 22, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61703372, Grant 61773351, and Grant 61603345; in part by the Key Scientific Research Project of Henan Higher Education under Grant 18A413012; in part by the Program for Science & Technology Innovation Talents in Universities of Henan Province under Grant 20HASTIT031; and in part by the Outstanding Foreign Scientist Project in Henan Province under Grant GZS2019008. Paper no. TII-19-3629.
(Corresponding author: Jinzhu Peng.)

J. Xin, C. Meng, J. Peng, and Y. Liu are with the School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China (e-mail: j.xin@zzu.edu.cn; zzumengchuang@163.com; jzpeng@zzu.edu.cn; liuyh@zzu.edu.cn).

F. Schulte and R. R. Negenborn are with the Department of Marine and Transport Technology, Delft University of Technology 2628 CD Delft, The Netherlands (e-mail: f.schulte@tudelft.nl; r.r.negenborn@tudelft.nl).
Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2020.2968099

for improving the efficiency of the multirobot cellular system [11]. These studies assume a centralized approach; due to communication limits and other constraints, the task allocation also can be achieved in a distributed way [12], [13].

In routing problems, due to the type difference of the robot, a task could be a continuous action between two different locations and a typical example is transporting materials using automated guided vehicles (AGVs). The task also can be a discrete action performed at a particular location (e.g., a welding task completed by a manipulator). Earlier routing problems of multiple AGVs focus on optimally sequencing tasks of AGVs only under collision constraints [14] and time window constraints [15]. Later on, the routing problem of AGVs is extended into an integrated problem taking the equipment interaction into account [16], [17]. The routing of multiple manipulators in the assembly station has received little attention and it is initialized by Spensieri *et al.* [2].

When it comes to path planning, collision avoidance between robots is essential for the planner and it must be considered. For multiple AGVs, a collision-free path problem is investigated in [18] and a distributed optimization approach is proposed for reducing the computational burden. In [19], the collision-free trajectories of two collaborative robot manipulators are planned using a new fuzzy genetic algorithm (GA). Furthermore, path planning has been incorporated into a design problem for car body assembly [20]. Generalized planning methods can be found in [21].

For multirobot systems in the manufacturing environment, these decision problems have been increasingly integrated into one decision problem as they are strongly coupled [16], [22]. For AGVs, dispatching and routing are integrated, and resolving conflicts in the path space of the AGVs is incorporated in [22]. The decision problem is regarded as a particular pickup and delivery problem. For robotic welding stations, not much work has been done with automatically configuring tasks and routes [23]–[25]. Currently, these decision problems in the assembly station are planned sequentially [2]. The routing problem is regarded as a multiple traveling salesman problem (mTSP) that neglects collision constraints and collisions are checked separately after computing the solution to the mTSP. It is observed that the collisions between these robots can take place at any time instant when planning their routes. Therefore, collision avoidance must be taken into account when routing these robots in the assembly station and this new collision-free routing problem needs to be investigated.

B. Contributions

This article investigates a new collision-free routing problem of a multirobot system in an assembly station. The contributions of this article are twofold, which are as follows.

1) A new time-space network model is proposed for the collision-free routing problem of the multirobot system. Using the TSN model, task sequences and the collision avoidance of multiple robots can be considered simultaneously in a unified framework. The corresponding planning problem results in a single mixed integer programming (MIP) problem to be solved.
2) A dedicated metaheuristic algorithm is developed for solving the collision-free routing problem based on the TSN formulation. This routing problem is a variant of mTSP using the TSN formulation. Existing methods have not addressed such a TSN that considers task sequences and collision avoidance simultaneously. Here, a new encoding scheme is designed to fit the TSN framework and fix infeasibilities by applying a repair operation. The developed algorithm can provide fast and high-quality solutions to the considered MIP problem.

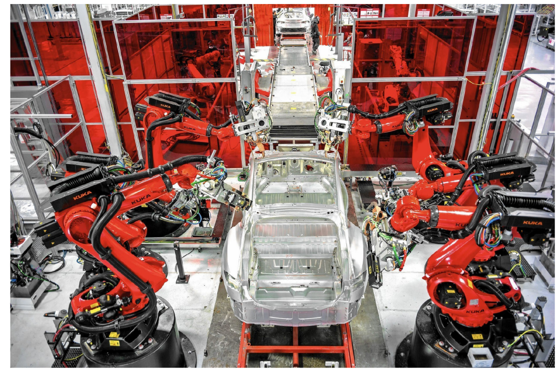


Fig. 1. Robotic assembly station in the Tesla factory. Tesla's special KUKA robots can perform different tasks [26].

The rest of this article is organized as follows. Section II formulates the mathematical model of the collision-free routing problem of the multirobot system using a TSN representation. Section III develops a GA for solving the corresponding MIP problem. In Section IV, simulation experiments are carried out and the results are analyzed. Finally, Section V concludes this article.

II. Time-Space Network Model

In this section, a mathematical model is proposed to formulate the collision-free routing problem of the multirobot system to complete a number of given tasks. The first part states the problem to be solved and associated assumptions in details. Then, the second part formulates this routing problem mathematically using a TSN model.

A. Problem Statement

In an assembly unit of the manufacturing system, multiple robots are required to cooperate for completing several tasks in a shared workspace. For instance, in the automobile industry, the tasks of stud welding or spot welding of a car body are performed by multiple cooperative robots together in the body-in-white process. Fig. 1 illustrates four cooperative robots for welding in an assembly station.

From the perspective of an operator, these robots need to complete these tasks in minimal cycle time on the one hand, and collision avoidances between robots during the operations must be considered on the other hand. Therefore, a collision-free routing problem for these multiple robots to complete several given tasks needs to be investigated. For this problem, important assumptions are given as follows.

1) All robots are identical.
2) Each task can be performed only by one robot.
3) Each robot starts the next task after completing the current task.
4) A particular group of tasks has been assigned to each robot in advance but the task sequence for each robot needs to be decided.
5) All robots start from their idle positions.
6) The task operation space is considered in the same plane, and the tool center point (TCP) is regarded as the reference point for the robot.
7) The focus is set on the operation of the end-effector and its connected joint, disregarding the state of other joints.

It is noted that, for the last assumption, each robot could use the kinematic redundancy of the manipulator to resolve the conflict of different operation points which are nearby. This redundancy enables the manipulator to obtain more degrees of freedom than needed to execute a given task. As a result, the focus can be set on the operation of the end-effector and its connected joint, and the three- dimensional (3-D) operation of the end-effector is projected onto an area in the plane. For the sake of simplicity, the area shape is considered to be circular. Furthermore, the last assumption applies to a small number of operational tasks and these tasks are located sparsely in open areas where every robot can operate.

In general, if collision constraints are not considered, the routing problem of multiple cooperative robots is identical to a mTSP. Therefore, the routing problem considered in this article can be regarded as a variant of the mTSP. For this routing problem, a new model taking into account the task sequences and collision avoidances simultaneously is proposed. In particular, the collisions at any operation point and between any operation points are both considered.

B. Modeling Using a TSN Model

Time-space network models have been widely used in the domain of transportation and logistics for describing the routing problem, including time constraints and location constraints [27], [28] enabling collision avoidance. Therefore, this article proposes a time-space network model for the particular mTSP considered.

It is assumed that, for m robots, robot k ($k \in \{1, 2, \ldots, m\}$) owns N_k tasks. In total, m directed graphs $G_k = (V_k, E_k)$ ($k \in \{1, 2, \ldots, m\}$) are considered. For robot k, V_k ($V_k = \{1, 2, \ldots, N_k\}$) is the collection of nodes and $E_k = \{(i, j)|i \in V_k, j \in V_k\}$ is the collection of arcs.

Here, a task is defined as performing a single operation (i.e., stud welding and spot welding) at a particular location. As a result, a particular node i corresponds to a location for the robot to perform a particular task and a particular arc (i, j) maps to the path from task i to task j. Different nodes represent different locations for the robot to visit.

TABLE I
NOMENCLATURE

Symbol	Description
V_k	Set of nodes assigned to robot k
E_k	Set of arcs that robot k may use
N_k	Number of tasks in V_k
m	Number of robots
k, k_1, k_2	Robot index, $k, k_1, k_2 \in \{1, 2, ..N\}$
i, j, n	Node index for a particular task, $i, j, n \in V_k$
T	Planning horizon
t	Time index, $t \in \{0, 1, .., T\}$
h_k	Idle position of robot k
$E_k^{\mathrm{o}}(i)$	Set of arcs starting from node i for robot k
$E_k^{\mathrm{s}}(i)$	Set of arcs ending at node i for robot k
w_{ij}	Time (integer) spent on arc (i, j)
z_{ik}	Given assignment of node i to robot k, $z_{ik} = 1$ if $i \in V_k$
(x_i, y_i)	Coordinate of node i
R	Diameter of the projected circular area occupied by the robot
M	A very large integer number

TABLE II
DECISION VARIABLES

Variable	Description
x_{ijk}	0-1 variable, $x_{ijk} = 1$ means that robot k moves over the arc (i, j), otherwise it is 0
a_{ijkt}	0-1 variable, $a_{ijkt} = 1$ when robot k reaches the arc (i, j) at time t, otherwise it is 0
d_{ijkt}	0-1 variable, robot k is 1 when leaving arc (i, j) at time t, otherwise 0
y_{ijkt}	0-1 variable, robot k is 1 when occupying arc (i, j) at time t, otherwise 0
$c_{k_1 k_2 t}^{q}$	0-1 variable, collision avoidance coefficient between robots k_1 and k_2 at time t
x_{kt}	Real variable, x-coordinate of robot k at time t
y_{kt}	Real variable, y-coordinate of robot k at time t

A planning horizon $T \times \Delta t$ equally discretized into a set of short time slots denoted by $\{\Delta t, 2\Delta t, \ldots, T \times \Delta t\}$ is considered. Δt is a time slot and T is the total number of time slots. As a result, the TSN model can decompose the overall routing process of multiple robots into several time slots. At each time instant $t \in \{0, 1, \ldots, T\}$, a particular robot can visit a particular node. During each time slot, the detailed path can be considered for avoiding collisions among these robots.

Before detailing the TSN model for representing this collision-free routing problem, the used subscripts, parameters, and decision variables are introduced in Tables I and II.

The objective is to minimize the cycle time for completing all the tasks performed by the cooperative robots. The details of this model are given as follows:

$$\min \left\{ \max_k \left\{ \sum_t t \times \sum_{i:(i,h_k) \in E_k^{\mathrm{s}}(h_k)} (d_{ijkt} - d_{ijk(t-1)}) \quad \forall k \right\} \right\}. \tag{1}$$

Subject to:

Group 1: Space constraints

$$\sum_{i,j:(i,j) \in E_k^{\mathrm{o}}(h_k)} x_{ijk} \geq 1 \quad \forall k \tag{2}$$

$$\sum_{i,j:(i,j) \in E_k^{\mathrm{s}}(h_k)} x_{ijk} \geq 1 \quad \forall k \tag{3}$$

$$\sum_{i:(i,j)\in E_k^{\mathrm{s}}(j)} x_{ijk} = \sum_{n:(j.n)\in E_k^{\mathrm{o}}(j)} x_{jnk} \quad \forall k,\ j \in V_k - h_k \tag{4}$$

$$\sum_{k=1}\sum_{j=1} x_{ijk} \geq 1 \quad \forall i \tag{5}$$

$$x_{ijk} \leq z_{ik} \quad \forall i,k \tag{6}$$

$$x_{jik} \leq z_{ik} \quad \forall i,k \tag{7}$$

$$x_{kt} = x_{k(t-1)} + \sum_{i,j:(i,j\in E_k)} \left(y_{ijk(t-1)} \times \frac{x_j - x_i}{w_{ij}} \right) \quad \forall k \tag{8}$$

$$y_{kt} = y_{k(t-1)} + \sum_{i,j:(i,j\in E_k)} \left(y_{ijkk(t-1)} \times \frac{y_j - y_i}{w_{ij}} \right) \quad \forall k. \tag{9}$$

Group 2: Time constraints

$$w_{ij} \times x_{ijk} = \sum_t (t \times (d_{ijkt} - d_{ijk(t-1)})) - \sum_t (t \times (a_{ijkt} - a_{ijk(t-1)})) \quad \forall k, (i,j) \in E_k, i \neq j \tag{10}$$

$$w_{ij} \times x_{ijk} \leq \sum_t (t \times (d_{ijkt} - d_{ijk(t-1)})) - \sum_t (t \times (a_{ijkt} - a_{ijk(t-1)})) \quad \forall k, (i,j) \in E_k, i = j \tag{11}$$

$$a_{ijkt} \geq a_{ijk(t-1)} \quad \forall k, (i,j) \in E_k, t \tag{12}$$

$$d_{ijkt} \geq d_{ijk(t-1)} \quad \forall k, (i,j) \in E_k, t \tag{13}$$

$$y_{ijkt} = a_{ijkt} - d_{ijkt} \quad \forall k, (i,j) \in E_k, t \tag{14}$$

$$\sum_{i.j:(i,j\in E_k)} y_{ijkt} \leq 1 \quad \forall k, t. \tag{15}$$

Group 3: Time-space constraints

$$\sum_{i:(i,h_k)\in E_k} d_{ih_k kT} \geq 1 \quad \forall k \tag{16}$$

$$\sum_{j:(h_k,j)\in E_k} a_{h_k jkt} = 1 \quad \forall k \tag{17}$$

$$x_{ijk} = a_{ijkT} \quad \forall k, (i,j) \in E_k \tag{18}$$

$$\sum_{i,j:(i,j)\in E_k} d_{ijkt} = \sum_{j.n:(j,n)\in E_k} a_{jnkt} \quad \forall k, j \in V_k - h_k \tag{19}$$

$$d_{ijkt} \leq a_{ijkt} \quad \forall k, (i,j) \in E_k, t. \tag{20}$$

Group 4: Collision avoidance constraints

$$x_{k_1 t} - x_{k_2 t} \geq R - M c^1_{k_1 k_2 t} \quad \forall k_1, k_2 \quad \forall t \tag{21}$$

$$x_{k_2 t} - x_{k_1 t} \geq R - M c^2_{k_1 k_2 t} \quad \forall k_1, k_2 \quad \forall t \tag{22}$$

$$y_{k_1 t} - y_{k_2 t} \geq R - M c^3_{k_1 k_2 t} \quad \forall k_1, k_2 \quad \forall t \tag{23}$$

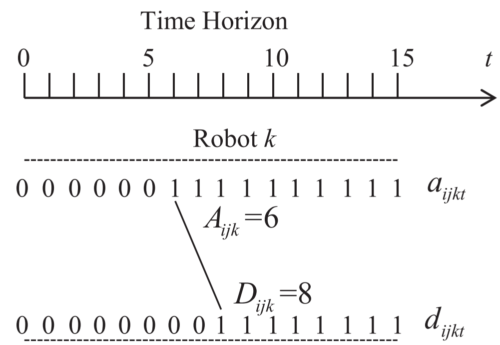


Fig. 2. Illustrative example of the defined variables A_{ijk} and D_{ijk}.

$$y_{k_2 t} - y_{k_1 t} \geq R - M c^4_{k_1 k_2 t} \quad \forall k_1, k_2 \quad \forall t \tag{24}$$

$$\sum_{q=1}^{4} C^q_{k_1 k_2 t} \leq 3 \quad \forall k_1, k_2 \quad \forall t. \tag{25}$$

Equation (1) is the objective function, representing the minimization of the cycle time of all robots to complete given tasks. The term $\sum_t t \times \sum_{i:(i,h_k)\in E_k^{\mathrm{s}}(h_k)} (d_{ijkt} - d_{ijk(t-1)})$ represents the completion time of robot k as there is only one arc to be selected from $E_k^{\mathrm{s}}(h_k)$. The term $\max$ denotes the cycle time of all robots.

In Group 1: Constraints (2)–(4) ensure the task sequence of the starting position, intermediate position, and ending position of each robot. Constraint (5) guarantees that all tasks are executed. Constraints (6) and (7) ensure that arcs (i,j) and (j,i) only can be visited by robot k when task i is assigned to it. Equations (8) and (9) update the position of robot k, ensuring that this robot can stay at node i or a fictitious point of arc (i,j) that is discretized into $w_{ij} - 1$ fictitious points.

In Group 2: Inequalities (10) and (11) are time constraints for each robot between two successive tasks ($i \neq j$), and each robot can wait more than one unit time in place (if $i = j$ and $w_{ij} = 1$). Inequalities (12) and (13) are the time connectivity constraint for each robot to perform two successive tasks. Inequalities (14) and (15) guarantee that each robot can only perform one task per unit time.

Here, variables A_{ijk} and D_{ijk} are defined for denoting the arrival time and the departure time of robot k at arc (i,j), respectively, which are illustrated in Fig. 2

$$A_{ijk} = \sum_t (t \times (a_{ijkt} - a_{ijk(t-1)})) \quad \forall k, (i,j) \in E_k \tag{26}$$

$$D_{ijk} = \sum_t (t \times (d_{ijkt} - d_{ijk(t-1)})) \quad \forall k, (i,j) \in E_k. \tag{27}$$

In Group 3: Inequalities (16) and (17) are time constraints of each robot at the start position and the end position, ensuring that each robot moves at the set start time. Equation (18) is the mapping constraint between the time-space network and the physical network. Constraints (19) and (20) guarantee the continuity of the arrival time and the departure time for robot k when visiting arc (i,j).

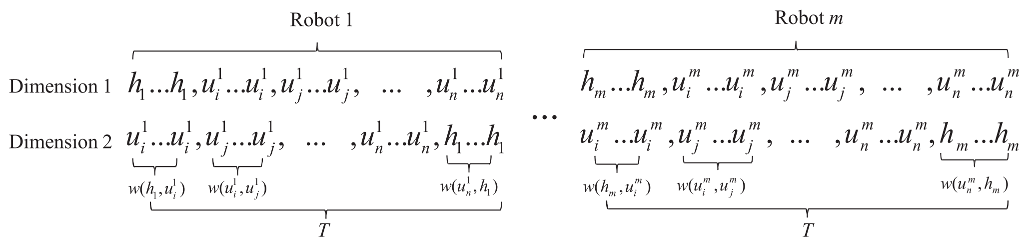


Fig. 3. Illustration of the 2-D encoding scheme.

In Group 4: Inequalities (21)–(25) guarantee that at any time t, each robot do not collide with each other [16]. The coordinates of robot k_1 (x_{k_1t}, y_{k_1t}) and (x_{k_2t}, y_{k_2t}) are the coordinates of robot k_1 and k_2 at time t. These coordinates are computed using (8) and (9). These constraints ensure that any two robots avoid collisions either at a node or at a fictitious point of the arc for each robot.

III. Solution Algorithm

This section develops a dedicated GA that efficiently solves the MIP problem based on the TSN model formulated in Section II. The MIP problem is a variant of mTSP known to be NP-hard. Commercial MIP solvers (e.g., CPLEX and Gurobi) struggle with providing high-quality solutions in a shortly reasonable time. Therefore, an efficient algorithm needs to be developed for solving this MIP problem.

GAs are regarded as an efficient metaheuristic method inspired by the process of natural selection and they are successfully used to solve real-world problems in operational research (e.g., job shop scheduling and mTSP) [29]. For the mTSP, typical problem formulations solved by GAs are limited to the assigned-based formulation and the flow-based formulation [30]–[32]. The GA for our problem needs to be tailored to work on a limited solution space—in comparison to common scheduling and routing problems—since a predefined task sequence needs to be considered and both arrival and departure times need to grant collision avoidance. These constraints are satisfied using a proposed two-dimensional (2-D) encoding scheme rather than the one dimensional (1-D) encoding scheme used for common scheduling and routing problems.

The proposed GA consists of the main procedures of selection, crossover, and mutation. The following parts introduce the encoding scheme and the algorithm details of the proposed new GA for the MIP problem based on the TSN formulation.

A. Encoding

For GAs, the first step is to generate an initial population that contains a set of possible solutions. The population quality closely relates to a proper encoding scheme that maps the problem into the algorithm [33]. Here, we propose a 2-D order based encoding used for the TSN model.

Given m robots and a task set V_k for robot k $(k \in \{1, 2, \ldots, m\})$. $V = V_1 \cup V_2 ... \cup V_m$ is the set of all tasks. For robot k, the set $V_k = \{u_1^k, u_2^k, \ldots, u_{N_k}^k\}$ and $u_i^k \neq u_j^k$ $(i \neq j)$.

Then, Fig. 3 illustrates the detailed encoding method based on defined symbols earlier. Dimension 1 denotes the starting node and dimension 2 represents the ending node within several time slots that corresponds to the time spent between these two nodes. The encoding length for each robot is the planning horizon T and the total length of the encoding scheme is mT. Each column of the encoding is a time slot indicating that the robot is occupying a particular path within this time window. The number of columns occupied by a particular arc in the encoding scheme corresponds to the number of the time slots occupied by a particular arc in the TSN model.

Algorithm 1: GA Based on the TSN Model.

iter $= 0$
initialize $P(\text{iter})$
if $P(\text{iter})$ have infeasible solutions **then**
    repair $P(\text{iter})$
end if
while iter $\leq$ iter$_{\max}$ **do**
    for $i = 1$ to N_{p} **do**
        evaluate the fitness $F(X)$ of $P(\text{iter})$
    end for
    select 1/4 of the $P(\text{iter})$ with the lower
    fitness as $P1(\text{iter})$
    for $i = 1$ to $N_{\text{p}}/4$ **do**
        selection operation to $P_1(\text{iter})$ as $P_2(\text{iter})$
        crossover operation to $P_1(\text{iter})$ as $P_3(\text{iter})$
        mutation operation to $P_1(\text{iter})$ as $P_4(\text{iter})$
    end for
    $P(\text{iter} + 1) =$
    $P1(\text{iter}) \cup P2(\text{iter}) \cup P3(\text{iter}) \cup P4(\text{iter})$
    iter $=$ iter $+ 1$
    if $P(\text{iter})$ have infeasible solutions **then**
        repair $P(\text{iter})$
    end if
end while

B. Algorithm

After introducing the encoding scheme, this part details the procedures of the proposed GA. Algorithm 1 presents the pseudocode of the GA proposed for the TSN model. iter denotes the iteration index, $F(X)$ is the fitness function of solution X. $P(\text{iter})$ is the entire population at iteration iter. $P_1(\text{iter})$, $P_2(\text{iter})$, $P_3(\text{iter})$, and $P_4(\text{iter})$ are the populations used for elite, selection, crossover, and mutation, respectively, and their sizes are all considered as one quarter of the entire population.

The core problem of algorithm design is to satisfy the constraints of the MIP problem. There are several ways to handle constraints in the GA and one simple method is the death penalty that abandons all infeasible solutions in the evolutionary process. However, the collision-free routing problem is a complicated constrained optimization problem. Many infeasible solutions in the population may be generated during the evolution process. If all infeasible solutions are abandoned, the diversity of the population may drop and the solution quality could be lowered. As a result, we choose the repair approach to make solutions that do not satisfy the constraints (2)–(20) feasible. Specifically, for any two successive arcs, the ending node of the first arc must be the starting node of the next arc. Also, for each arc, the number of columns must be consistent with its spending time.

In addition to the repair operation for avoiding infeasible solutions resulting from constraints (2)–(20), extra infeasible solutions could be detected due to constraints (21)–(25) for collision avoidance. For constraints (21)–(25), we apply a penalty function that avoids searching in the infeasible part of the solution space. The penalty function is incorporated into the fitness function $F(X)$ defined as follows:

$$F(X) = F_0(X) + p(\text{iter}, X) \tag{28}$$

where $F_0(X)$ represents the cycle time of solution X. $p(\text{iter}, X)$ is the penalty function at iteration iter defined as follows:

$$p(\text{iter}, X) = \rho_{\text{iter}}^{\alpha} d^{\beta}(X) \tag{29}$$

where $\rho_{\text{iter}}^{\alpha}$ is the variable multiplication factor and $d^{\beta}(X)$ is the penalty for breach of constraints. iter is the number of iterations of GAs. α and β are the parameters that adjust the size of the penalty value

$$\rho_{\text{iter}} = C \times \text{iter} \tag{30}$$

$$d(x) = \begin{cases} 0, & X \text{is feasible} \\ |R - l|, & \text{otherwise} \end{cases} \tag{31}$$

where C is a constant. l is the minimum distance between every two robots on all unit time windows. Following sections will carry out a number of simulation experiments using the proposed GA for providing fast and good solutions.

IV. Case Studies

In this section, simulation experiments are conducted to verify the planned collision-free routes of the multirobot system using the proposed TSN model. The first part gives the detailed configuration and parameters of the multirobot station. Afterwards, two typical types of case studies are considered and the results of the new GA are compared with two state-of-the-art commercial solvers and a practical approach.

A. Setting of the Multirobot Simulation System

The routing problem is mathematically modeled using MATLAB (version: R20180612) on Windows 8 (64 b) together with the Yalmip toolbox. The computer hardware is Intel Core i5-4200 U (1.62 Hz) with 4 GB of memory.

TABLE III
Key Parameters of the Multirobot Station (Adapted From [34])

Parameter	Description	Value	Unit
R	Diameter of the projected circular area	10	[cm]
WR	Working range of each robot	150	[cm]
Δt	time slot unit	1	[second]

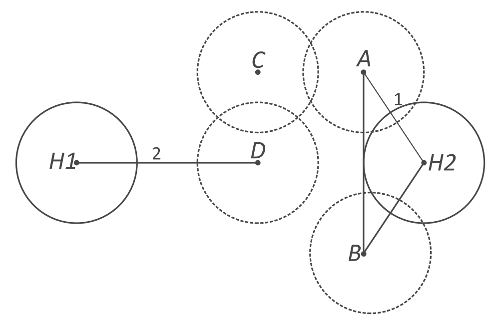


Fig. 4. Geometrical overview of two robots and four tasks.

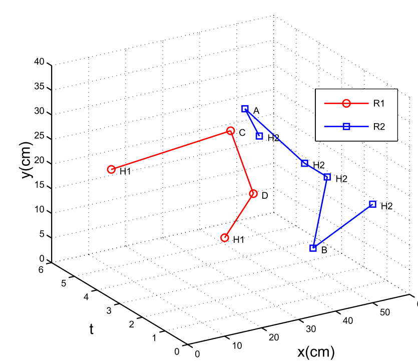


Fig. 5. Determined routes of the two robots (four tasks).

The software RobotStudio, which is ABB's offline robot programming tool, is used to simulate the operations of the multirobot station and verify the results using the proposed time-space network model. The ABB IRB2400 series robot is chosen for the experimental simulation. This robot moves horizontally and vertically within the working range centered on itself. The diameter R corresponds to its safe working range, taking into account the size of the joint connected to the end-effector. Key parameters for planning the multirobot system are given in Table III.

B. Welding on a Door

The first type of case study is to weld on a door, typically requiring two robots in a station. Two scenarios with four tasks and ten tasks are considered. For each scenario, the operation nodes are distributed equally and ten experiments have been carried out. The solver CPLEX is used for solving the MIP problem for determining the collision-free route.

For the scenario of four tasks, Fig. 4 gives geometrical overview of two robots and four tasks for the simulation. A, B, C, and D are task nodes, and H_1 and H_2 are their home nodes. Figs. 5 and 6 show the detailed collision-free routes

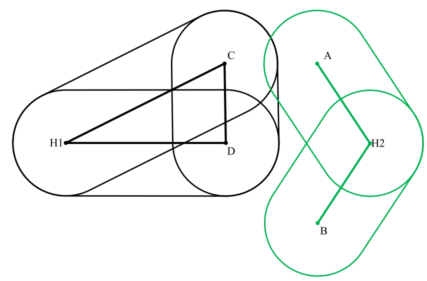


Fig. 6. Collision-free routes of two robots (four tasks).

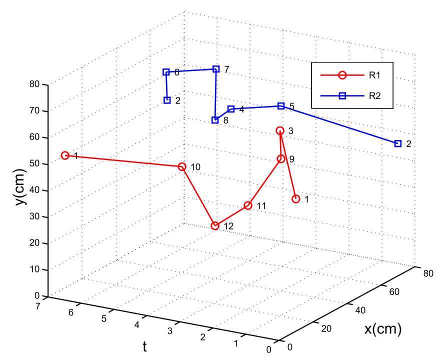


Fig. 7. Determined routes of the two robots (ten tasks).

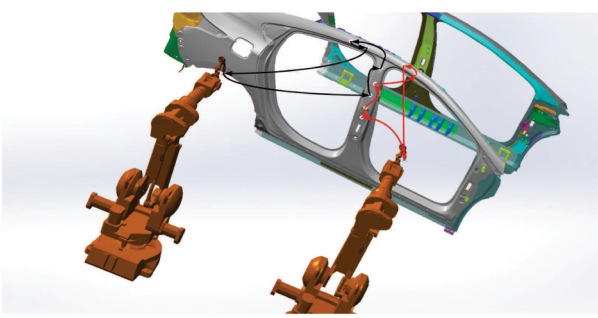


Fig. 8. Illustration of two robots for completing ten welding tasks on a door.

of each robot obtained using the proposed TSN model. The determined visited node sequences are as follows: R_1:$H_1 \rightarrow D \rightarrow C \rightarrow H_1$; R_2:$H_2 \rightarrow B \rightarrow H_2 \rightarrow H_2 \rightarrow A \rightarrow H_2$. When robot R_1 sequentially visits the nodes D and C, robot R_2 selects to visit node A and returns to its home position H_2, and then visit node B avoiding the conflict with robot $R1$. Therefore, the collision-free trajectories of these two robots can be obtained using the proposed time space model.

For another case of ten tasks, the relating results are shown in Figs. 7 and 8. Fig. 7 shows the visited node sequences of these two robots obtained using the proposed TSN model. Fig. 8 illustrates the collision-free routes of two robots for completing ten tasks, indicating that the routing of multiple robots based on the TSN model satisfies the collision-free and time constraints.

Table IV records the computation times of the case studies considered earlier. For these two robot, the computation time grows as the number of tasks increases.

TABLE IV
COMPUTATIONAL TIMES REGARDING THE CASE STUDIES OF WELDING ON A DOOR

Description	Robots	Tasks	Averaged computational times
Scenario 1	2	4	18.45±0.20 (seconds)
Scenario 2	2	10	63.70s±0.77 (seconds)

TABLE V
SETTINGS OF CASE STUDIES TO BE CONDUCTED

Description	Robots	Tasks
Scenario 3	2	10
Scenario 4	2	20
Scenario 5	4	20
Scenario 6	4	30
Scenario 7	4	40

TABLE VI
AVERAGED COMPUTATION TIMES USING DIFFERENT METHODS FOR CASE STUDIES OF WELDING ON AN UNDERBODY (UNIT: SECONDS)

	CPLEX	GUROBI	Sequential	Proposed GA
Scenario 3	732.33±0.94	832.02±1.23	21.54±0.08	22.00±0.95
Scenario 4	3600±0	3600±0	22.38±0.07	25.67±0.50
Scenario 5	3600±0	3600±0	24.45±0.22	70.01±0.34
Scenario 6	3600±0	3600±0	25.26±0.02	75.85±0.51
Scenario 7	3600±0	3600±0	26.67±0.89	83.17±1.49

TABLE VII
AVERAGED CYCLE TIMES USING DIFFERENT METHODS FOR CASE STUDIES OF WELDING ON AN UNDERBODY (UNIT: SECONDS)

	CPLEX	GUROBI	Sequential	Proposed GA
Scenario 3	18±0	18±0	26.6±1.2	18.8±0.4
Scenario 4	N.S	N.S	43.6±2.0	24.5±1.7
Scenario 5	N.S	N.S	46.3±0.7	17.7±0.7
Scenario 6	N.S	N.S	55.8±0.8	26.8±0.8
Scenario 7	N.S	N.S	67.7±0.7	32.4±1.1

Note: N.S—no solutions.

C. Welding on an Underbody

This part considers another type of case study, i.e., spot welding on an underbody, which constitutes the base structure for all assembly parts, such as the engine and chassis. Such circumstances normally require two or four robots.

The setting of case studies are given in Table V, as suggested by Spensieri *et al.* [2]. For each case, the task nodes are distributed randomly and ten experiments are carried out. The results of the proposed GA are compared with commercial solves CPLEX and Gurobi and a sequential approach. In the sequential approach, all the robots are planned in a prioritized order [35], [36] and the overall mTSP is decomposed into mTSPs. The collision-free routing of each TSP is solved by the general GA sequentially, regarding the previous planned robot routes as obstacles. The performances of these methods are compared in Tables VI and VII. The maximal computation time is set to be 1 h.

Table VI gives the averaged computation times of the proposed GA in comparison to two commercial solvers and the sequential method regarding Scenarios 3–7. It can be found that for the smaller case, CPLEX and GUROBI use significantly computational times, whereas for the bigger cases, these two commercial solvers cannot obtain feasible solutions within 1 h.

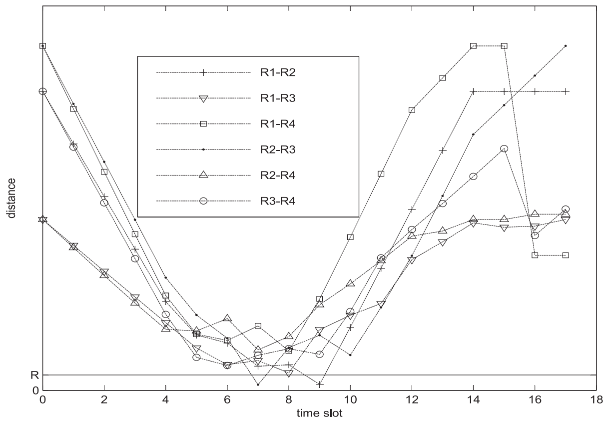


Fig. 9. Relative distances between each two robots using the mTSP neglecting collision constraints for Scenario 5.

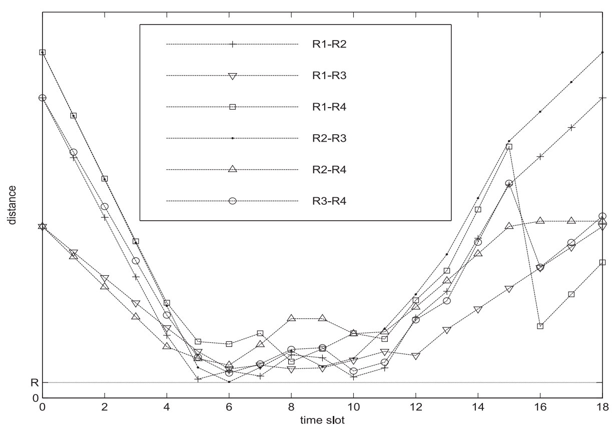


Fig. 10. Relative distances between each two robots using the proposed TSN model for Scenario 5.

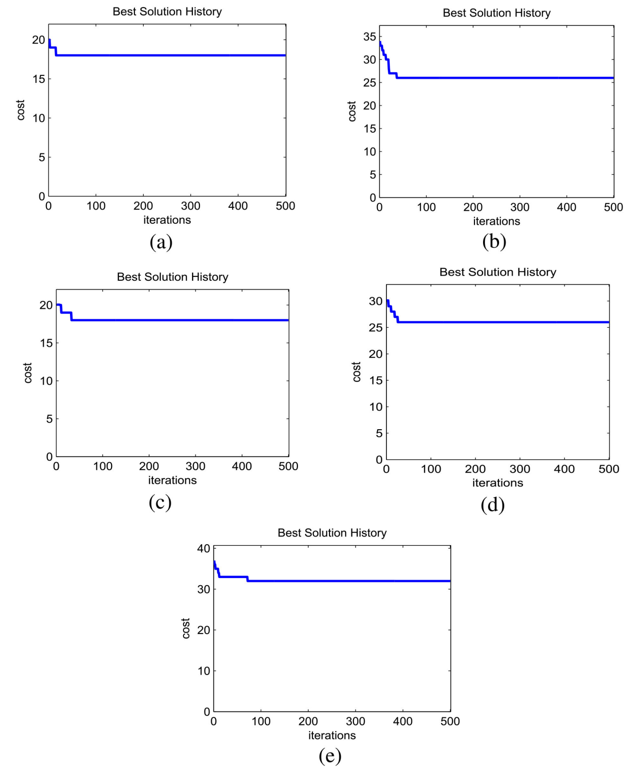


Fig. 11. Convergence curve of the proposed GA for Scenarios 3–7. (a) Scenario 3. (b) Scenario 4. (c) Scenario 5. (d) Scenario 6. (e) Scenario 7.

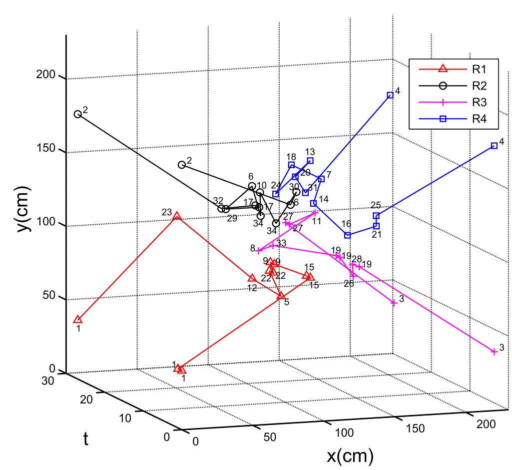


Fig. 12. Robot routes of Scenario 6 based on the TSN model.

The sequential approach decomposes the overall optimization problem into several smaller optimization problems, and therefore, the computational times become considerably shorter. For Scenarios 3–7, the proposed GA can solve the overall optimization problem using greatly shorter times than using these two state-of-the-art commercial solvers.

Table VII lists the averaged cycle times, which are the crucial performance indicator for the manufacturer, of the compared four methods concerning Scenarios 3–7. CPLEX and GUROBI obtain the minimal cycle times in Scenario 3, which is the smallest scale. However, as the system increases, these two solvers struggle with the computational complexity of the problem and cannot give feasible solutions within the maximal setting times. For all the cases considered, the proposed GA can provide fast and good quality solutions. For Scenarios 3, the gap between the GA and these two commercial solvers is acceptable but the computation times are much shorter. Although the sequential approach achieves the shortest computation times, the solution quality is not satisfactory as its cycle times are much larger than the proposed GA.

To demonstrate the potential of the proposed time-space network modeling framework and the proposed new GA, here, we detail a simulation result of Scenario 5 consisting of 4 robots and 20 predefined tasks. For the experiment of Scenario 5, Fig. 9 gives the relative distances of every two robots if collision constraints are not considered, and the planning problem becomes a traditional mTSP. Fig. 10 shows these relative distances based the proposed TSN model, including collision constraints. It can be seen clearly from Figs. 9 and 10 that the proposed TSN model can avoid collisions between robots while the traditional mTSP model cannot obtain the collision-free routes. Fig. 11 presents the convergence curve of the proposed GA for solving the TSN

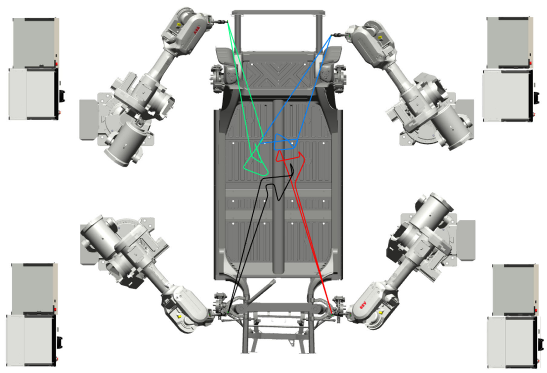


Fig. 13. Illustration of robot traces using the TSN model for Scenario 6.

model and this curve indicates the solution is obtained in less than 100 iterations.

In the last part, we present a more complicated Scenario 6 involving four robots and 30 predefined tasks. The determined routes are illustrated in Fig. 12 and their individual TCP traces in RobotStudio are illustrated in Fig. 13. The multiple robots move in a collision-free way among the predefined tasks avoiding collisions between each robot, aiming to minimize the cycle time. A more vivid demonstration can found in a video on[1].

V. Conclusion

This article proposed a new methodology for collision-free routing planning of the multirobot system of an assembly station. In the routing problem, the objective was to minimize the cycle time of given tasks for each robot. This routing problem included collision avoidance constraints, and a time-space network model was constructed to map the robot location constraints into the route planning framework, leading to an MIP problem. Since the related planning problem was a variant mTSP known as NP-hard, a dedicated GA was developed for solving this MIP problem. We designed a new encoding scheme for describing the transition between two adjacent task locations. The simulation experiments indicated that the proposed TSN model obtained the collision-free route of these robots. Compared to the state-of-the-art commercial solvers and a practical approach, the proposed new GA can provide fast and high-quality solutions.

Future research will consider including energy consumption as another objective for this routing problem and investigate the tradeoff between cycle time and energy consumption. Another interesting research direction is to incorporate attitude control of the robot [37], [38] into the considered collision-free routing problem.

References

[1] 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," *Eur. J. Oper. Res.*, vol. 263, no. 3, pp. 1033–1048, 2017.

[2] D. Spensieri, J. Carlson, F. Ekstedt, and R. Bohlin, "An iterative approach for collision free routing and scheduling in multirobot stations," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 950–962, Apr. 2016.

[1] [Online]. Available: https://youtu.be/WjyKmcr4CkM

[3] Y. Zhou, H. Hu, Y. Liu, and Z. Ding, "Collision and deadlock avoidance in multirobot systems: A distributed approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1712–1726, Jul. 2017.

[4] P. Long, T. Fanl, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 6252–6259.

[5] D. Herrero-Perez and H. Martinez-Barbera, "Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems," *IEEE Trans. Ind. Informat.*, vol. 6, no. 2, pp. 166–180, May 2010.

[6] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *Int. J. Robot. Res.*, vol. 32, no. 12, pp. 1495–1512, 2013.

[7] D. H. Lee, S. A. Zaheer, and J. H. Kim, "A resource-oriented, decentralized auction algorithm for multirobot task allocation," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1469–1481, Oct. 2015.

[8] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith, "Dynamic vehicle routing for robotic systems," *Proc. IEEE*, vol. 99, no. 9, pp. 1482–1504, Sep. 2011.

[9] G. Demesure, M. Defoort, A. Bekrar, D. Trentesaux, and M. Djemai, "Decentralized motion planning and scheduling of AGVs in an FMS," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1744–1752, Apr. 2018.

[10] E. Åblad, D. Spensieri, R. Bohlin, and J. S. Carlson, "Intersection-free geometrical partitioning of multirobot stations for cycle time optimization," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 842–851, Apr. 2018.

[11] I. Suemitsu, K. Izui, T. Yamada, S. Nishiwaki, A. Noda, and T. Nagatani, "Simultaneous optimization of layout and task schedule for robotic cellular manufacturing systems," *Comput. Ind. Eng.*, vol. 102, pp. 396–407, 2016.

[12] S. Giordani, M. Lujak, and F. Martinelli, "A distributed multi-agent production planning and scheduling framework for mobile robots," *Comput. Ind. Eng.*, vol. 64, no. 1, pp. 19–30, 2013.

[13] L. Jin and S. Li, "Distributed task allocation of multiple robots: A control perspective," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 5, pp. 693–701, May 2018.

[14] T. Nishi and Y. Tanaka, "Petri net decomposition approach for dispatching and conflict-free routing of bidirectional automated guided vehicle systems," *IEEE Trans. Syst., Man, Cybern., A, Syst. Humans*, vol. 42, no. 5, pp. 1230–1243, Sep. 2012.

[15] N. Smolic-Rocak, S. Bogdan, Z. Kovacic, and T. Petrovic, "Time windows based dynamic routing in multi-AGV systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 151–155, Jan. 2010.

[16] J. Xin, R. R. Negenborn, F. Corman, and G. Lodewijks, "Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 377–396, 2015.

[17] M. Saidi-Mehrabad, S. Dehnavi-Arani, F. Evazabadian, and V. Mahmoodian, "An ant colony algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs," *Comput. Ind. Eng.*, vol. 86, pp. 2–13, 2015.

[18] T. Nishi, M. Ando, and M. Konishi, "Distributed route planning for multiple mobile robots using an augmented Lagrangian decomposition and coordination technique," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1191–1200, Dec. 2005.

[19] E. A. Merchán-Cruz and A. S. Morris, "Fuzzy-GA-based trajectory planner for robot manipulators sharing a common workspace," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 613–624, Aug. 2006.

[20] S. Pellegrinelli, N. Pedrocchi, L. M. Tosatti, A. Fischer, and T. Tolio, "Multi-robot spot-welding cells for car-body assembly: Design and motion planning," *Robot. Comput.-Integr. Manuf.*, vol. 44, pp. 97–116, 2017.

[21] J. C. Latombe, *Robot Motion Planning*, vol. 124. Berlin, Germany: Springer, 2012.

[22] T. Miyamoto and K. Inoue, "Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems," *Comput. Ind. Eng.*, vol. 91, pp. 1–9, 2016.

[23] N. Papakostas, G. Michalos, S. Makris, D. Zouzias, and G. Chryssolouris, "Industrial applications with cooperating robots for the flexible assembly," *Int. J. Comput. Integr. Manuf.*, vol. 24, no. 7, pp. 650–660, 2011.

[24] S. Pellegrinelli, N. Pedrocchi, L. M. Tosatti, A. Fischer, and T. Tolio, "Multi-robot spot-welding cells: An integrated approach to cell design and motion planning," *CIRP Ann.*, vol. 63, no. 1, pp. 17–20, 2014.

[25] D. Hömberg, C. Landry, M. Skutella, and W. A. Welz, "Automatic reconfiguration of robotic welding cells," in *Proc. Math Digit. Factory*, 2017, pp. 183–203.

[26] Tesla, 2019. [Online]. Available: www.tesla.com

[27] A. I. Corréa, A. Langevin, and L.-M. Rousseau, "Scheduling and routing of automated guided vehicles: A hybrid approach," *Comput. Oper. Res.*, vol. 34, no. 6, pp. 1688–1707, 2007.

[28] L. Meng and X. Zhou, "Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables," *Transp. Res. B, Methodological*, vol. 67, pp. 208–234, 2014.

[29] B. Çaliş and S. Bulkan, "A research survey: Review of AI solution strategies of job shop scheduling problem," *J. Intell. Manuf.*, vol. 26, no. 5, pp. 961–973, 2015.

[30] T. Bektas, "The multiple traveling salesman problem: An overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.

[31] S. Yuan, B. Skinner, S. Huang, and D. Liu, "A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms," *Eur. J. Oper. Res.*, vol. 228, no. 1, pp. 72–82, 2013.

[32] J. Wang, O. K. Ersoy, M. He, and F. Wang, "Multi-offspring genetic algorithm and its application to the traveling salesman problem," *Appl. Soft Comput.*, vol. 43, pp. 415–423, 2016.

[33] V. Toğan and A. T. Daloğlu, "An improved genetic algorithm with initial population strategy and self-adaptive member grouping," *Comput. Struct.*, vol. 86, no. 11/12, pp. 1204–1218, 2008.

[34] ABB, "Technical data for the IRB 2400 industrial robot," 2019. [Online]. Available: https://new.abb.com/products/robotics/industrial-robots/irb-2400/irb-24 00-data

[35] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Algorithmica*, vol. 2, no. 1–4, pp. 477–521, 1987.

[36] J. P. Van Den Berg and M. H. Overmars, "Prioritized motion planning for multiple robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 430–435.

[37] L. Jin, S. Li, X. Luo, Y. Li, and B. Qin, "Neural dynamics for cooperative control of redundant robot manipulators," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 3812–3821, Sep. 2018.

[38] Z. Xie, L. Jin, X. Du, X. Xiao, H. Li, and S. Li, "On generalized RMP scheme for redundant robot manipulators aided with dynamic neural networks and nonconvex bound constraints," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5172–5181, Sep. 2019.

Jianbin Xin received the B.Sc. degree in electrical engineering from Xidian University, Xi'an, China, in 2007, the M.Sc. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2010, and the Ph.D. degree in operational control for logistics from the Delft University of Technology, Delft, The Netherlands, in 2015.

He is currently an Associate Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China. His research interests include modeling and control of smart logistics systems and hybrid systems control.

Chuang Meng received the bachelor's degree in electrical engineering from the Luoyang Institute of Science and Technology, Luoyang, China, in 2018. He is currently working toward the M.Sc. degree in control science and engineering with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China.

His research interest focuses in planning of multiple robots for manufacturing and logistics.

Frederik Schulte received the Diploma degree in industrial engineering (with majors) in information systems and logistics from the Technical University of Hamburg, Hamburg, Germany, in 2012, and the Ph.D. degree in logistics from the University of Hamburg, Hamburg, Germany, in 2017.

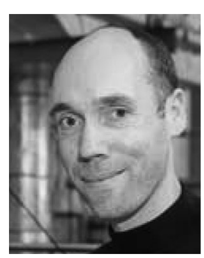
He is currently a Tenure Assistant Professor with the Department of Maritime Transport and Technology, Delft University of Technology, Delft, The Netherlands. His research focuses on collaborative transportation and implications of the fourth industrial revolution.

Jinzhu Peng received the B.E. degree in measurement and control technology, M.E. and Ph.D. degrees in control science and engineering from Hunan University, Changsha, China, in 2002, 2005, and 2008, respectively.

From 2009 to 2011, he was a Postdoctoral Fellow with the University of New Brunswick, Fredericton, NB, Canada. He is currently an Associate Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China. His research interests include robotic control, compliant control, and human–robot interactions and collaborations.

Yanhong Liu received the B.E. degree in electrical engineering from the Zhengzhou University of Light Industry, Zhengzhou, China, in 1992, and the M.E. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, China, in 2002 and 2006, respectively.

From 2012 to 2013, she was a Visiting Scholar with the University of California San Diego, San Diego, CA, USA. She is currently a Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China. Her research interests include nonlinear system modeling and control, robotic control, and human–robot interactions and collaborations.



Rudy R. Negenborn received the M.Sc. degree in computer science from Utrecht University, Utrecht, The Netherlands, in 2003, and the Ph.D. degree in distributed control from the Delft Center for Systems and Control, Delft University of Technology (TU Delft), Delft, The Netherlands, in 2007.

He is currently a Full Professor in Multi-Machine Operations and Logistics and the Head of the Section 'Transport Engineering and Logistics,' Department of Maritime and Transport Technology, Faculty of Mechanical, Maritime and Materials Engineering, TU Delft. He was the Director of Studies of the M.Sc. program on Transport, Infrastructure and Logistics, in 2017 and 2018. In addition, he edited the books *Intelligent Infrastructures* (Springer, 2009), *Distributed Model Predictive Control Made Easy* (Springer, 2014), and *Transport Of Water versus Transport Over Water* (Springer, 2015). He has coauthored more than 200 publications in peer-reviewed journals and conference proceedings. He develops theories to address control problems in large-scale transport and logistics systems in general, and for realizing the potential of autonomous vessels in particular. His more fundamental research interests include distributed control, multiagent systems, model predictive control, and optimization.

Dr. Negenborn is the recipient of an NWO/STW VENI grant and several other national and international research grants from 2010 onward.