Front. Mech. Eng. 2023, 18(2): 25
https://doi.org/10.1007/s11465-022-0741-z

RESEARCH ARTICLE

High-efficiency inspecting method for mobile robots based on task planning for heat transfer tubes in a steam generator

Biying XU, Xuehe ZHANG, Yue OU, Kuan ZHANG, Zhenming XING, Hegao CAI, Jie ZHAO, Jizhuang FAN (✉)

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China
✉ *Corresponding author. E-mail: fanjizhuang@hit.edu.cn (Jizhuang FAN)*

ABSTRACT Many heat transfer tubes are distributed on the tube plates of a steam generator that requires periodic inspection by robots. Existing inspection robots are usually involved in issues: Robots with manipulators need complicated installation due to their fixed base; tube mobile robots suffer from low running efficiency because of their structural restricts. Since there are thousands of tubes to be checked, task planning is essential to guarantee the precise, orderly, and efficient inspection process. Most in-service robots check the task tubes using row-by-row and column-by-column planning. This leads to unnecessary inspections, resulting in a long shutdown and affecting the regular operation of a nuclear power plant. Therefore, this paper introduces the structure and control system of a dexterous robot and proposes a task planning method. This method proceeds into three steps: task allocation, base position search, and sequence planning. To allocate the task regions, this method calculates the tool work matrix and proposes a criterion to evaluate a sub-region. And then all tasks contained in the sub-region are considered globally to search the base positions. Lastly, we apply an improved ant colony algorithm for base sequence planning and determine the inspection orders according to the planned path. We validated the optimized algorithm by conducting task planning experiments using our robot on a tube sheet. The results show that the proposed method can accomplish full task coverage with few repetitive or redundant inspections and it increases the efficiency by 33.31% compared to the traditional planning algorithms.

KEYWORDS steam generator transfer tubes, mobile robot, dexterous structure, task planning, efficient inspection

1 Introduction

Nuclear power is widely used due to its environmental friendliness, cleanliness, and safety. It currently accounts for more than 10% of the world's total annual electricity generation [1]. A steam generator (SG) is a heat exchanger that carries the water heated in the primary circuit to a turbine to power a turbine generator in a nuclear power plant (NPP) [2]. The thickness of the tubes is only 1–1.2 mm, which makes an SG fragile. Thus, regular maintenance is required [3]. Because of the high radioactivity, robots replace manual operations to deliver the inspection tools [4].

SG inspection robots can be categorized by their installation and operation mode: robots with fixed-base manipulators and tube mobile robots moving on the tube plate, as shown in Fig. 1. The former robot fixes its base and sends the eddy probes into the heat transfer tubes for inspection. As early as the 1990s, Westinghouse put ROSA-III, a 6-degree-of-freedom (6-DoF) manipulator, in service. The robot uses distance and vision sensors to locate and inspect the tube holes [5]. AREVA in France developed the ROGER [6]. This robot is folded to enter the water chamber through the manway and then reaches the required position along the slideway. Researchers at Harbin Engineering University developed a 7-DoF manipulator and designed the joint structure [7]. The experiments proved that the manipulator could accomplish the plugging operation on the tube plate. Although the fixed-base manipulator has relatively simple structures and plain control strategies, its workspace is strictly limited so that frequent disassembling and installing are required to cover large-scale tube plates.

Received May 6, 2022; accepted November 7, 2022

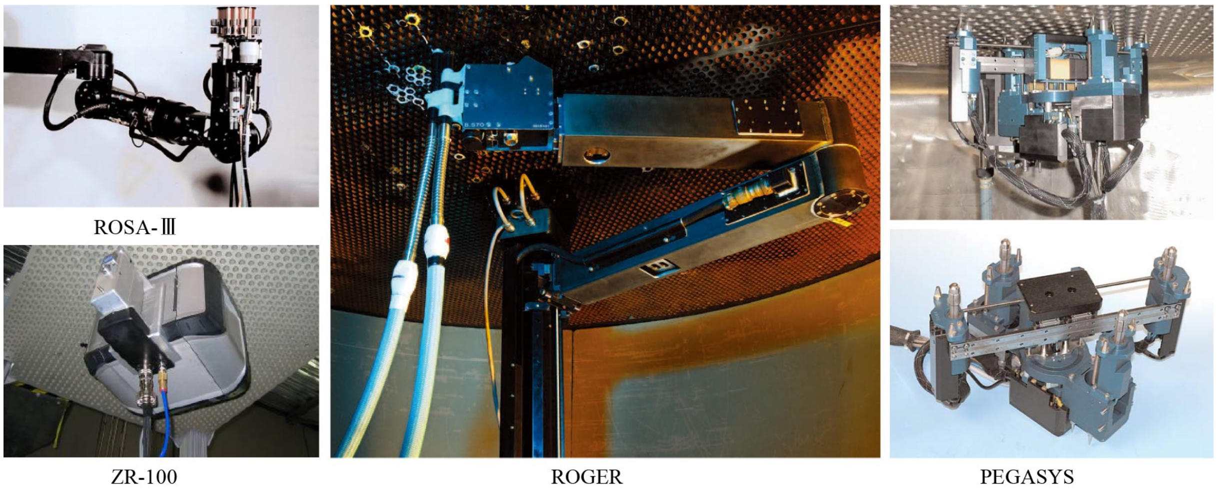


Fig. 1 An overview of steam generator heat transfer tube inspection robots.

In comparison, the tube mobile robot can hang on the tube plate by a special tensing mechanism. The robot can tighten the tube hole and travel on the tube plate, carrying the eddy probes to inspect the heat transfer tubes. As the pioneer, the Pegasys, designed by Westinghouse, can travel on the tube plate by its body parts moving alternately [8,9]. However, the robot has to rotate at fixed angles subject to the structural constraints, which significantly decreases its efficiency. The ZR-100 by ZETEC is small and flexible with a high moving speed and unlimited rotation angle [9]. Nonetheless, its tensing mechanisms must stay tight during the inspection so that the workspace is limited and the inspection time is prolonged.

Current researches have focused on inspection robot structures [10], but there is less research on inspection algorithms. Due to the large size of the tube plate, the inspection tasks should cover many tubes, each of which takes about two minutes to inspect. Therefore, the total inspection time can be quite long [11]. To improve efficiency, the mobile robot generally carries two tools to work simultaneously [12]. The current algorithms perform a row-by-row and column-by-column traversal search along the direction of tube distribution. However, these methods result in duplication and redundancy in inspections, prolonging the shutdown of NPPs. Therefore, a more effective robot search algorithm is necessary to heighten the reliability of NPPs.

Task allocation for a single robot involves forming one-to-one mapping from the robot to the task. To solve this problem, many optimal allocation algorithms for robotic arm operation have been proposed [13]. The Hungarian algorithm forms a cost matrix for performing tasks and solves the matrix exchange to obtain the optimal allocation [14], but the computation is complex when the number of tasks is large [15]. Gerkey and Mataric [16] converted the task allocation problem into a linear programming problem to reduce computational complexity, but their algorithm cannot handle multiple tasks. Khaluf et al. [17] proposed an optimal ant colony algorithm that rationally distributed multiple tasks within a specific environment and validated the algorithm in a static environment. However, the algorithm is slow to iterate and may miss optimal solutions. In addition, these algorithms do not fully utilize the distribution of tasks.

After task allocation, planning appropriate robot base positions for sub-regions also plays an essential role in task planning. Thakar et al. [18] proposed a hierarchical search algorithm that combines the time cost and the search space to plan the robot's stations for picking up and transporting parts during motion. A collision-free inverse kinematic solution was also proposed with a pre-calculated reachability database to determine the feasible base positions. Moreover, based on gripping efficiency and stability, it reduces the base positions [19]. Vafadar et al. [20] presented a novel motion planning approach to minimize the number of robot positions and end-effector motions. However, most base planning methods focus on the time optimality and the balance between the robot movement and end-effector gesture changes, but they do not delve into collision avoidance and accessibility. Furthermore, some researchers introduced efficient multi-agent planning algorithms [21,22]. But due to the economic benefit and the space limitation in SG environment, we focus on task planning using a single robot at this stage.

To take advantages of the tube mobile robots and improve their inspection flexibility, we designed a tube mobile robot with a highly-integrated structure and robust control system [23]. Based on the characteristic of the robot, our previous paper concentrates on path planning by considering every single task individually. However, we notice that this method is time-consuming. To make full use of the distribution of tasks, we propose a method to allocate the task regions and plan the robot base positions of the sub-regions. By applying the method, a robot can perform all the tasks with a short movement time and a low completed task ratio (the ratio of the completed tasks to required tasks, which evaluates the number of the repetitive or redundant tasks). An overview of the method of a robot inspection system of an SG is shown in Fig. 2.

This paper is organized as follows: Section 2 introduces the structure and control system of the tube mobile robot. Section 3 defines the task planning for inspected tubes. Section 4 introduces the methods for task allocation and base position search. Section 5 shows experiments comparing the proposed and traditional methods. Section 6 summarizes this paper.

2 Robot structure and control system

HIT-Crawler, the tube mobile robot we design, works on the tube plate (a semicircular sheet). It consists of four parts: the base, the inspection tool, the foot (a sliding mechanism), and the toes (tensing mechanisms). Both the base and foot are equipped with a group of toes to make them fixed or active by corresponding tensing mechanisms that grip or release from the plate. The base is designed with a prismatic joint and a revolute joint, as well as a linear motor that can drive the robot to move vertically. With different gripping states, the base and foot can translate and rotate. To be more specific, the foot can rotate and translate when the base grips and the foot releases, whereas the base can rotate and translate when the foot releases and the base grips. Besides, if both grip, the robot cannot move. At no time do the base and foot release simultaneously. The robot can move around the plate with the above states and then rotate the tool to deliver the eddy probes into the tube to be inspected. The robot with DoF and the tool workspace are shown in Fig. 3.

The operation of the robot is mainly separated into two phases: the walking phase and the inspection phase. In the walking phase, the robot can travel around the plate through the state conversion, the translation, and the rotation of the robot. As for the inspection phase, the robot can be modeled as a revolute–prismatic–revolute manipulator and the schematic diagram is shown in Fig. 4. The Denavit–Hertenbeg (D–H) parameter is shown in Table 1.

To boost the performance of our robot, we design a highly-integrated modular control system, as shown in Fig. 5. There is a control box outside the robot body, containing an industrial computer and an Elmo motion controller. The operator uses our software on a remote PC to exchange status and commands with the industrial computer via TCP/IP protocol. Then the industrial computer controls the Elmo controller using Modbus protocol. The control unit on the robot body, including a driver module and an I/O module, is under the control of the Elmo controller. The motor driver module receives instructions from EtherCAT and drives the motors, while the I/O module controls the proximity switches and the electromagnetic valves.

Our software is composed of four modules: the display interface, the operation interface, the calculation module, and the communication module. The display interface shows the real-time status of the robot, including both 2D

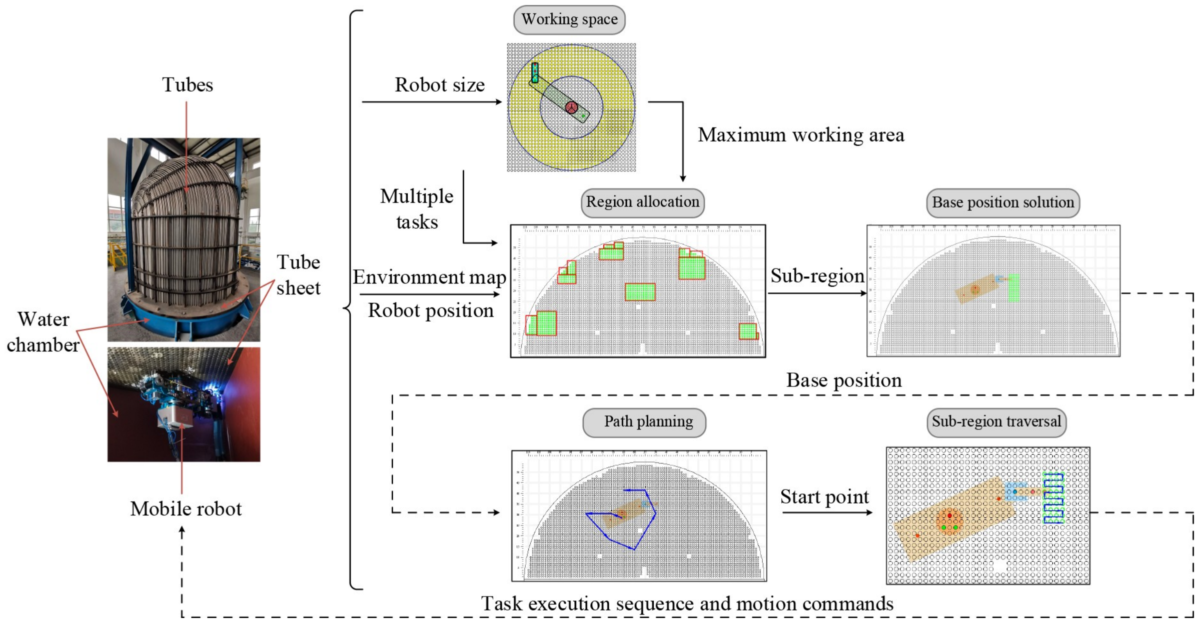
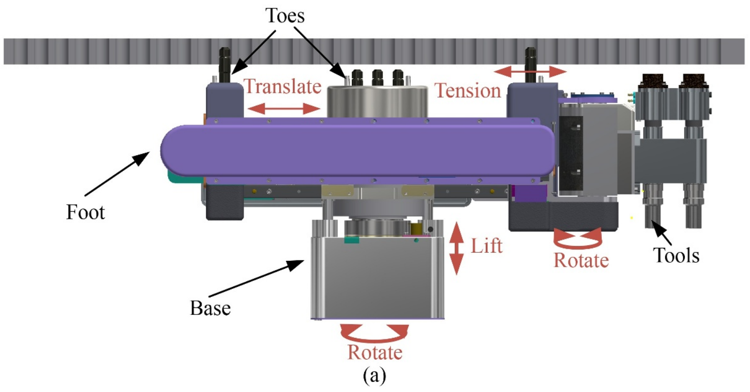


Fig. 2 An overview of the high-efficiency inspecting method for mobile robots based on task planning for heat transfer tubes in a steam generator.



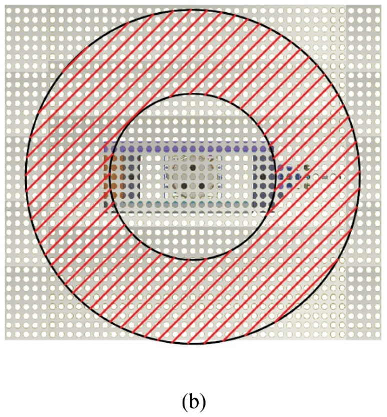


Fig. 3 Robot and tool workspace. (a) Front view of the robot with degrees of freedom. Joint 1 is a translation joint and Joints 2 and 3 are rotation joints and their limitations are [5d, 11d], [0, 2π), and [0, π], respectively, where d is the distance between two tube holes. (b) Tool workspace (red ring area) with the tool length of 2.

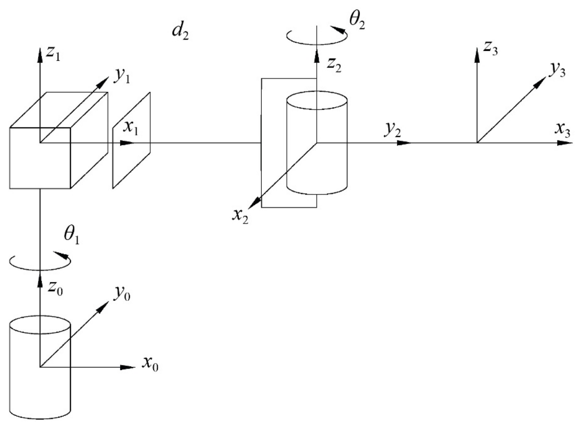


Fig. 4 Schematic diagram of HIT-Crawler.

Table 1 D–H parameter of the robot

Link	θ_i	α_i	L_i	d_i
1	θ_1	π/2	0	0
2	0	π/2	0	d_2
3	θ_3	0	d	0

and 3D demonstrations of the current robot location and motion, based on data obtained from the control system. The operation interface allows operators to control the robot manually. The calculation module is mainly responsible to conduct planning procedures and generalize corresponding control commands, and then the communication module delivers the commands to the control system. Moreover, our software has the capacity to handle exceptions and errors to make sure that our robot can work safely and reliably.

Furthermore, we experimented with the motion accuracy of the designed robot. In our previous paper, results showed that the motion accuracy of the robot is up to 0.37 mm and the robot speed can reach approximately 9.54 mm/s. Therefore, the structure and control system ensure the perfect motion performance of the robot.

3 Problem formulation

In a robot inspection system of an SG shown in Fig. 2, multitasking is used to plan the robot positions and joint configurations corresponding to tasks so that the robot can fully inspect the tube holes efficiently.

To formulate the problem, we have these conditions:

1) Only one omnidirectional mobile robot works on the tube plate of the SG and it inspects along any direction.

2) The robot translation and rotation need to satisfy the joint limits.

3) The robot must avoid collision with the water chamber during motion and inspection.

The task planning problem is defined as follows: The

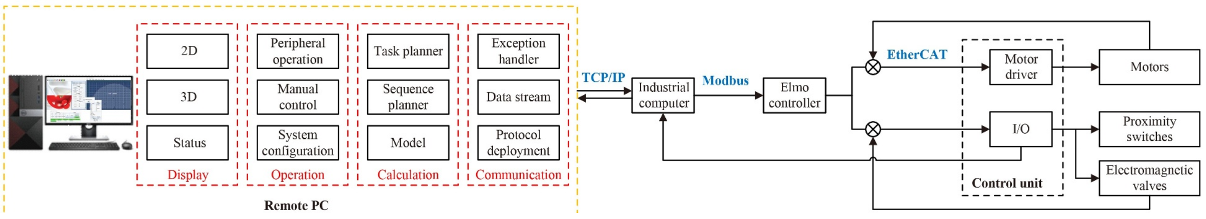


Fig. 5 Diagram of the robot control system.

task set $\mathcal{T}=\{T_1,T_2,...,T_n\}$ consists of n tasks, where T_i denotes the task tube hole (x_i,y_i), and (x_i,y_i) denotes the coordinates of the tube hole i. Task allocation is to divide the task set $\mathcal{T}$ into sub-regions $\{\mathcal{V}_1,\mathcal{V}_2,...,\mathcal{V}_{n_s}\}$. For each sub-region $\mathcal{V}_j$ $(j=1,2,...,n_s)$, a unique group of parameter (w_j,l_j) can be determined to describe $\mathcal{V}_j$, where integers w_j and l_j denote the width and length of $\mathcal{V}_j$ respectively. Then, the task allocation result can be obtained as

$$\{\mathcal{V}_1,\mathcal{V}_2,...,\mathcal{V}_{n_s}\}=\underset{\mathcal{V}_1,\mathcal{V}_2,...,\mathcal{V}_{n_s}\subset\mathcal{T}}{\arg\min}\sum_j f(w_j,l_j,k), \quad (1)$$

where $f(w_j,l_j,k)$ denotes the minimum number of tasks to be completed of $\mathcal{V}_j$. The distance between the robot two tools is k times the distance between two holes. For a specific tube plate and robot, k is a fixed integer greater than 1.

A base search is to obtain the configuration matrix R_p with n positions and configurations:

$$R_\mathrm{p}=\begin{bmatrix} x_1 & y_1 & q_1^1 & q_1^2 & q_1^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_h & y_h & q_h^1 & q_h^2 & q_h^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & q_n^1 & q_n^2 & q_n^3 \end{bmatrix}, \quad (2)$$

where (x_h,y_h) is the robot base position solution, and $[q_h^1 \ \ q_h^2 \ \ q_h^3]$ is the robot joint configuration solution corresponding to task h, $h=1,2,...,n$.

The base search aims to find a matrix R_p^* that minimizes the path cost and the number of base positions, i.e.,

$$\begin{aligned} R'_\mathrm{p} &= \underset{R_\mathrm{p}}{\arg\min}\, F_\mathrm{D}(\mathcal{B}), \\ R_\mathrm{p}^* &= \underset{R_\mathrm{p}\in R'_\mathrm{p}}{\arg\min}\, L_\mathrm{Ant}(\mathcal{B}), \end{aligned} \quad (3)$$

where $\mathcal{B}$ is a point set containing n base positions $(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$, $F_\mathrm{D}(\mathcal{B})$ is the number of different elements in $\mathcal{B}$, $L_\mathrm{Ant}(\mathcal{B})$ is the total distance of the points in the set $\mathcal{B}$ traversed sequentially after path planning and R'_p is an intermediate variable to obtain R_p^* which denotes all matrices that minimize the number of base positions.

4 Method

4.1 Input parameters

As shown in Table 2, the input parameters of the proposed method are mainly categorized into two types: explicit parameters, which are influenced by the working environment, and implicit parameters, which describe the specification of the robot.

1) Explicit parameters

a) Distribution of the tube holes

The distribution of the tube holes determines the robot moving direction. In this paper, the robot can move quad-directionally on the square-distributed tube plate, as shown in Fig. 6.

b) Distribution of the plugging holes

The plugging holes are the holes in which the robot toes cannot grasp or release. Therefore, the distribution of the plugging holes directly influences the robot motion. The robot must always plan a path to prevent from fixing in a plugging hole. Too many plugging holes result in an unfavorable long path with many turns.

c) Distribution of the task holes

The distribution of the task holes is the most essential factor affecting the task planning results. A series of task holes scattered over the plate may cause exponentially increasing number of base positions for the robot inspection.

2) Implicit parameters

a) Robot maximum translation distance

The robot should be designed with a proper maximum translation distance to ensure its better performance: A greater distance means the robot can move longer in a single translation, but the robot needs to be with a larger size and weight. Nevertheless, the limited workspace and plate weight capacity restrict the design of the robot: The

Table 2 Task planning input parameters

No.	Parameters	Symbol	Influence
1	Distribution of the tube holes	$\{\mathrm{Hole}(x,y)\}$	Robot moving direction
2	Distribution of the plugging holes	$\{\mathrm{Plug}(x,y)\}$	Planned path
3	Distribution of the task holes	$\{\mathrm{Task}(x,y)\}$	Task planning results
4	Robot maximum translation distance	$S_\mathrm{max}=6$	Motion speed and accuracy
5	Distribution of the base toes	L_Outer	Robot reachable position
6	Distance between the foot toes	$\{\mathrm{Base}(x,y)\}$	Tool working space
7	Distance between the tools	$L_\mathrm{OutertoTool}$	Task planning results
8	Distance from the foot toe to the tool	k	Tool working space
9	Robot rotation speed	C	Action cost
10	Robot translation speed	V_Rot	Action cost
11	Robot grasping time	V_Trans	Action cost
12	Robot releasing time	T_Grasp	Action cost

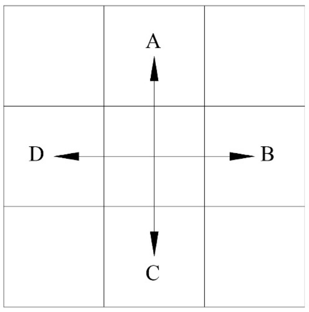


Fig. 6 Robot move direction.

robot should be compact with its dimension strictly within 400 mm $\times$ 300 mm $\times$ 300 mm and the body weight less than 12.5 kg.

b) Distribution of the base toes

The distribution of the base toes is considered when we check the reachability of a base position in a path searching process.

c) Distance between the foot toes, distance between the tools and distance from the foot toe to the tool

These distances are used to calculate the size of the tool workspace and judge the feasibility of a base position during path planning. In addition, the distance between the tools determines k, as we mentioned in the problem formulation section. A smaller distance k is preferred, because a short tool distance benefits the inspection efficiency. However, from the mechanical design limitation, we cannot assign $k = 1$ considering the width of attached tools.

d) The robot rotation and translation speed and time of the grasp and release from the tube

To evaluate the cost of an action during path planning, the motion speed and the time of grasp and release are two significant parameters, which determine the “preference” of the path planner. Let C denotes the cost of rotation relative to translation. As we discuss in our previous work [23], if C is too small, the path planner may mistakenly prefer adding too many turns. If C is large, the path planner may prefer a straighter path, but there may exist a necessary turn too close to the target holes. That results that the last step is short, which is unexpected.

Before we move to our task planning implementation, we parameterize all factors mentioned above, except k and C, which can be optimized then.

4.2 Workspace analysis

Since k is greater than 1, the total minimum inspection length for a line of holes is $2k$, as shown in Fig. 7, and the size of the optimal region is $(m\times 2k)$ ($m = 1,2,...,R_m$, R_m denotes the maximum size of the optimal region and is dependent on the tool workspace). $\{2k, 2\times 2k, 3\times 2k, ...\}$ is the optimal working width set, $\{k+1, k+2, ..., 2k-1\}$ is the suboptimal working width set, and 1 is the working width set along the length direction.

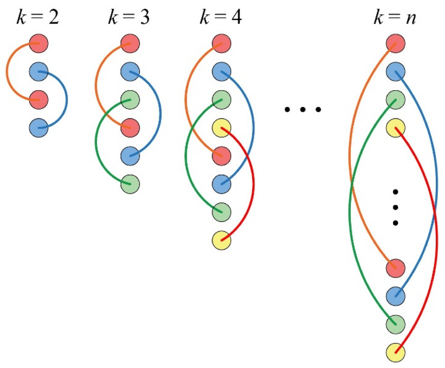


Fig. 7 An illustration of the total minimum inspection length for a line of holes.

To make full use of the tool characteristics, this paper establishes the optimal work matrix T_{pb}, the suboptimal work matrix T_{ps}, and the length work matrix T_{pl} as

$$T_{\text{pb}} = \begin{bmatrix} 2k & l_{2k} \\ 4k & l_{4k} \\ \vdots & \vdots \\ R_m\cdot 2k & l_{R_m\cdot 2k} \end{bmatrix},$$

$$T_{\text{ps}} = \begin{bmatrix} k+1 & l_{k+1} \\ k+2 & l_{k+2} \\ \vdots & \vdots \\ 2k-1 & l_{2k-1} \end{bmatrix}, \ T_{\text{pl}} = \begin{bmatrix} 1 & l_1 \end{bmatrix}, \tag{4}$$

where the first column of the matrix indicates the width of the respective work region, and the second column indicates the corresponding maximum length within the tool workspace.

For example, for $k = 2$, the regions are shown in Fig. 8. Here, the optimal width is shown only up to 16, and the actual $R_m = 6$ with the maximum length of 24.

4.3 Task allocation

Sub-regions are not always optimal in task allocation, which leads to unexpected duplicate inspections. Considering that the main direction affects the inspection performance, this paper proposes a region evaluation function with width as the optimization objective. We denote the width by w and the length by l.

To reduce the completed task ratio, three types of inspections are specified: inspecting along the optimal or suboptimal width direction (Fig. 9(a)), length direction (Fig. 9(b)), and compound direction (Fig. 9(c)). According to this approach, the number of completed tasks $N(w)$ is that in Eq. (5), where $\lfloor x \rfloor$ is the downward rounding function and $a\%b$ is the modular function. The robot works along the optimal width in the region corresponding to w_{2k}. For the compound region, the robot inspects along the length corresponding to w and works along the sub-optimal width in the remaining region.

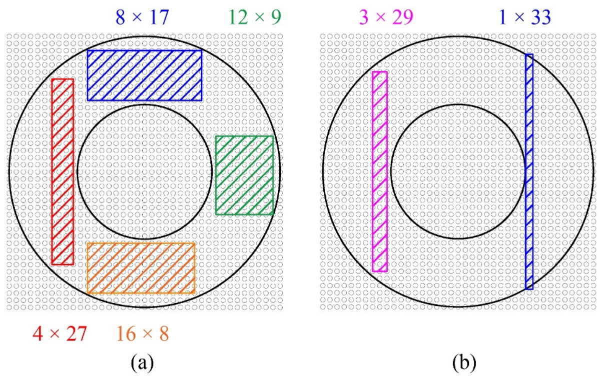


Fig. 8 Working region for $k = 2$: (a) optimal work regions, (b) suboptimal work region and length work region. The colored regions in (a) are the optimal work regions, the pink region in (b) is the sub-optimal work region, and the blue region in (b) is the length work region.

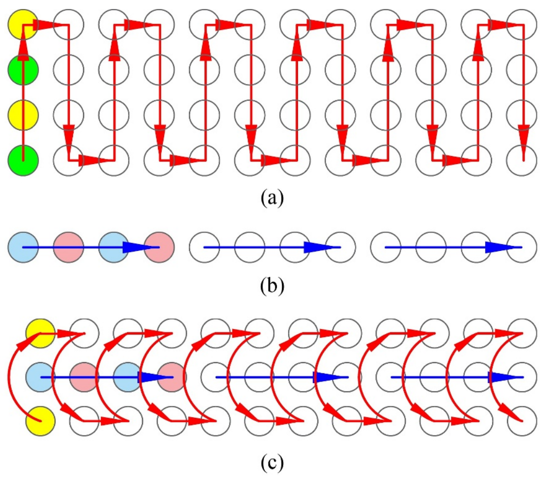


Fig. 9 Three types of inspection: (a) width direction, (b) length direction, and (c) compound direction.

$$N(w) = \sum_{m=k+1}^{2k} \lfloor w_m \% m \rfloor \cdot 2(m-k) + 2\bar{w}. \tag{5}$$

For $m = 2k, 2k-1, ..., k+1$, w_m and w satisfy

$$\begin{cases} w_{2k} = w, \\ w_{2k-1} = w_{2k} - (w_{2k} \% 2k) 2k, \\ w_{2k-2} = w_{2k-1} - (w_{2k-1} \% (2k-1))(2k-1), \\ \vdots \\ w_{k+1} = w_{k+2} - (w_{k+2} \% (k+2))(k+2), \\ \bar{w} = w_{k+1} - (w_{k+1} \% (k+1))(k+1). \end{cases} \tag{6}$$

The evaluation function of the main working direction is defined as $V(w)$ in Eq. (7), where $\alpha_{2k} > 0$ is the factor along the width direction, $\bar{\alpha} < 0$ is the factor along the length direction, and $\alpha_{2k-1}, \alpha_{2k-2}, ..., \alpha_{k+1}$ is the factor of the compound direction, and all of them take values between [0, 1]. A larger evaluation function value means a higher superiority of the direction as a main direction. Given a rectangular task region (with length l and width w), the optimal main direction of inspection can be determined from the evaluation function value $\max\{V(l), V(w)\}$.

$$V(w) = \sum_{m=k+1}^{2k} \alpha_m \frac{\lfloor w_m \% m \rfloor m}{N(w)} + \bar{\alpha} \frac{2\bar{w}}{N(w)}. \tag{7}$$

The given multiple tasks are usually distributed randomly. For better integration of task allocation, two principles for handling task regions are proposed in this paper: the merging principle and complementing principle.

The merging principle applies to close regions with the same dimension in the same direction. We use a rectangular region of width/length a to cover m rectangular regions of width/length: $a_1, a_2, ..., a_m$, where $a = a_1 = a_2 = \cdots = a_m$. The merged region is only adopted when the following condition is satisfied:

$$V(w_{R_M}) > \sum_{k=1}^{m} V(w_{R_k}). \tag{8}$$

An example is shown in Fig. 10(a) for better demonstration of the merging principle. Here $k = 2$ and there are three task regions. We take $\alpha_4 = 0.3$, $\alpha_3 = 0.2$, and $\bar{\alpha} = -0.5$. The task regions have a common width dimension of 6, and the evaluation function value of the merged region can be calculated that $V(16) = 0.3$, while the evaluation function values of the three sub-regions before merging are $V(5) = 0.03$, $V(3) = -0.15$, and $V(4) = 0.3$. Because $V(16) > V(5) + V(3) + V(4)$, the regions can be merged.

The complementing principle means that for some irregularly shaped regions, the rectangular region with the smallest size can be computed to cover all tasks, as shown in Fig. 10(b).

Merging and complementing the task regions ensure integration during the task allocation instead of having to plan regions one by one. These two principles are

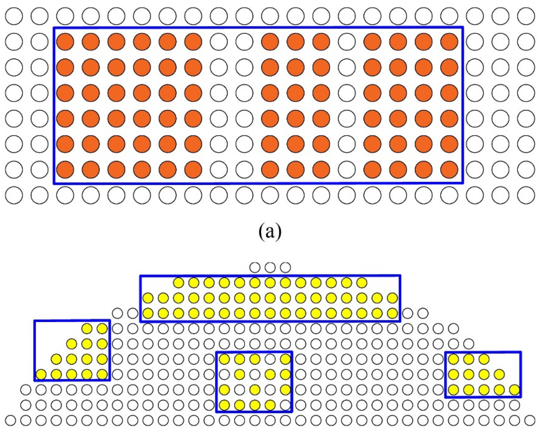


Fig. 10 Examples of (a) merging and (b) complementing principles.

essential for optimizing the robot base search and enhancing the speed and reliability of a robot.

4.4 Robot base position search

The robot base positions are where a robot can deliver tools to all tubes in the corresponding sub-regions. A base position is feasible if a set of poses relative to the base exists so that the tools can inspect all tubes. The base area consists of all practicable base positions of the sub-regions.

The robot tool workspace of a fixed base and the base area with a fixed tool are equally-sized ring areas. By considering collision and structure, the base area of the corresponding task can be obtained through the inverse kinematics (IK) traverse method, as shown in Fig. 11.

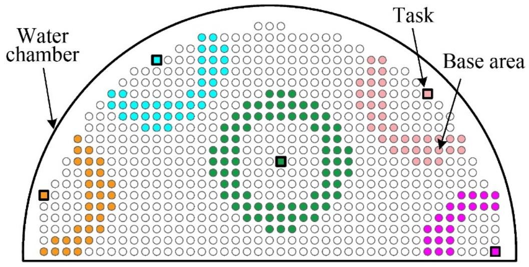


Fig. 11 An illustration of task base area. The colored squares represent task tubes, with corresponding task areas filled with the same color.

An illustration of task base area. The colored squares represent task tubes, with corresponding task areas filled with the same color.

Base area intersection of multiple tasks. The black point is current robot base. The ring areas are the base areas of corresponding tasks. Their intersections are red-colored. The red point surrounded by a pentagon is the optimal base position.

The closest position to the current robot in the intersection of base areas is the optimal base position of multiple tasks, as shown in Fig. 12. Because the task number is usually large, the computation can be massive if IK is solved to find the optimal position for each task. To speed up the calculation, an overall base search method is performed by using optimal, suboptimal, and length work regions.

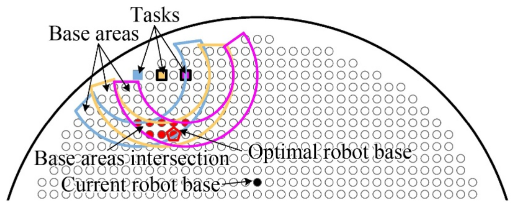


Fig. 12 Base area intersection of multiple tasks. The black point is current robot base. The ring areas are the base areas of corresponding tasks. Their intersections are red-colored. The red point surrounded by a pentagon is the optimal base position.

First, the search direction needs to be determined. Considering the distance of the two boundaries along the width direction of the rectangular region, three main directions exist in the base search: from far to near, from near to far, and from the middle to both sides. The approach with the second direction only considers close tasks without advanced planning for the far tasks, and the positions planned will cause the robot to move frequently in small steps. In contrast, the latter approach will destroy the search integrity. Therefore, an overall base search with a far-to-near direction is ideal, and it is divided into three steps.

4.4.1 Determine the length of the overall search

Find the width of the unsearched sub-region in the pre-calculated work matrices to obtain the maximum working length. Compare the sub-region length with the maximum working length and select the smaller one as the search length.

4.4.2 Select key tasks

Judge the key tasks instead of all tasks in the sub-region. Selecting key tasks on both sides of the region may result in incorrect optimal work position solutions since the working rectangle area under the key tasks may be out of the working area ring, as shown in Fig. 13. Therefore, for the region of length l, the key tasks correspond to tasks in lines [0], [$l/2$], and [l].

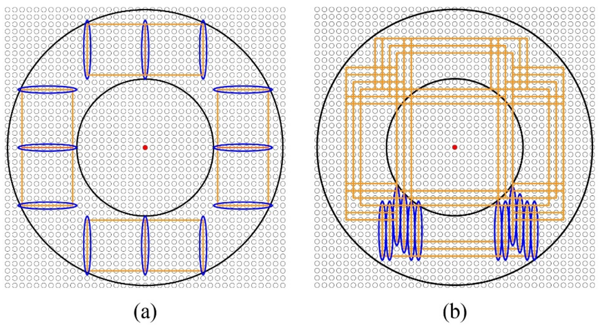


Fig. 13 Key task selection: (a) both-sides key tasks and (b) first-middle-last key tasks. The red point denotes the robot base, the orange rectangle denotes the sub-region, and the blue oval denotes the key tasks.

4.4.3 Re-judge the tasks

After the optimal base position is obtained, the tasks need to be re-judged. If tasks in each line meet the requirements of IK verification and collision conditions, the tasks in that line correspond to the optimal position. Otherwise, for the particular line where tasks fail to meet the requirements, IK calculation is separately performed to find the corresponding feasible base position, and then we revert to the first step.

To ensure that the robot moves a shorter distance when inspecting holes, the robot's current position refers to the previous optimal base position after the first solution is obtained. The robot base search algorithm is shown in Algorithm 1, where $\bar{l}$ denotes the length of the unassigned work row, l_s denotes the length of the search row, $l_{\max}$ denotes the maximum length of the region according to the work matrix, B_{Current} is the current base position, and B_{Best} is the best base position for the current searching region. Here the base position set $\mathcal{B}$ has N base points $B_1, B_2, ..., B_N$.

Algorithm 1: Base search

Input: Rectangular region of length l_R and width w_R

Output: Base point set $\mathcal{B}$

```
1  while $\bar{l} > 0$ do
2      if $N > 0$ then
3          if $B_N \in \mathcal{B}$ can search the current row then
4              $B_{\text{Current}} \leftarrow B_N$;
5              $\bar{l} \leftarrow \bar{l} - 1$;
6              Continue;
7          end
8      end
9      $l_s = \min(l_{\max}, \bar{l})$;
10     if $B_{\text{Best}}$ exists for row $[0]$, $[l_s/2]$, $[l_s]$ then
11         $B_{N+1} \leftarrow B_{\text{Best}}$;
12     else
13         Search row $[0]$ alone;
14     end
15     $\bar{l} \leftarrow \bar{l} - 1$;
16 end
```

When the path connecting base points is planned and the task order is determined, different actions are assigned to the robot so that the robot base position is updated. The process continues until all actions are executed.

4.5 Robot path, base sequence, and task sequence planning robot base position search

After the base positions are obtained, an order of the positions and a path between them needs to be planned. We have already proposed a path planning method in Ref. [23], and the base sequence planning is similar to the traveling salesman problem (TSP) [24]. Stutzle and Hoos [25] were inspired by the foraging behavior of ants and proposed an ant colony optimization (ACO) algorithm that can be used to solve the TSP problem. A modified particle swarm optimization (PSO) algorithm was presented by Cui et al. [26] for a typical combinatorial optimization problem. Khan and Maiti [27] explained the performance for artificial bee colony optimization (ABC) in solving TSP. Due to the excellent iteration and computation speed, the improved ACO is used [28,29]. The heuristic cost function $d(\text{cur}, \text{next}, t)$ from the current position $(x_{\text{cur}}, y_{\text{cur}})$ to the next position $(x_{\text{next}}, y_{\text{next}})$ is defined in Eq. (9) as the distance criterion:

$$d(\text{cur}, \text{next}, t) = \frac{|x_{\text{cur}} - x_{\text{next}}| + |y_{\text{cur}} - y_{\text{next}}|}{S_{\max}} \cdot C_w + t \cdot C_r, \quad (9)$$

where $S_{\max}$ denotes the maximum steps the robot can move, t is the number of turns, C_w is the translation cost, and C_r is the turning cost.

Because one base position often corresponds to multiple tasks, it is necessary to plan the task sequence at a fixed base position, i.e., the robot's work strategy needs to be determined within the task sub-region [30,31]. Acar covered the environmental cell using a simple round-trip method based on their proposed Morse decomposition [32]. According to the robot motion mode, we use the round-trip traversal method for task completion. We first select the task from the sub-region that is nearest to the current robot tool as the starting task and then traversed the sub-region sequentially along the planned direction. Whenever the traversal reaches the boundary, we skip to the near line and search in the opposite direction. This way, we can conduct iterative inspection until all tasks are covered.

5 Experiments

5.1 Task planning experiment

Taking both analytical and simulative results into consideration, combining the practical expertise, we use $C = 2.1$ and $k = 2$ hereinafter as a group of practically optimal parameters for our robot.

We performed task planning experiments to prove the advantage of the proposed algorithm. A mobile robot was used to move on a 110×54 semicircular tube plate (Fig. 14). Five different groups of experiment settings were designed, as shown in Fig. 15. We implemented algorithm M1 for searching by rows, algorithm M2 for searching by columns, and algorithm M3 proposed in this paper. Then, we applied them to conduct the tasks on a PC with 8 GB RAM, Intel Core i5 CPU, and Windows 10 OS with Java 1.8.0.

The experiments were reflected in the number of actual inspected tasks and the planned base positions, distance cost of the planned path, computation time, and robot's running time as the leading indicators. When calculating the cost function, C_w was taken as 1, and C_r was taken as 2.1 [25]. The number of ants was 100, and the number of iterations was 150 in the ant colony algorithm. The running time consisted of the time for motion and delivering the end tools to the designated holes, excluding the inspecting time.

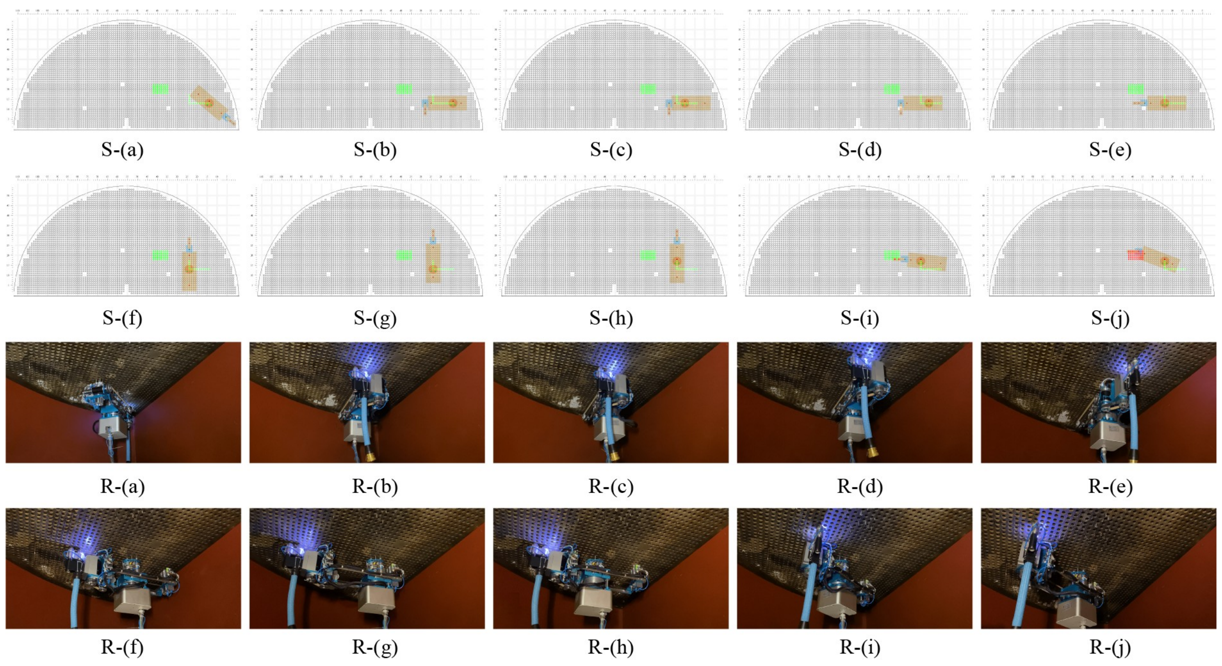


Fig. 14 Setup of the experiments and a part of the robot moving process of Group 1.

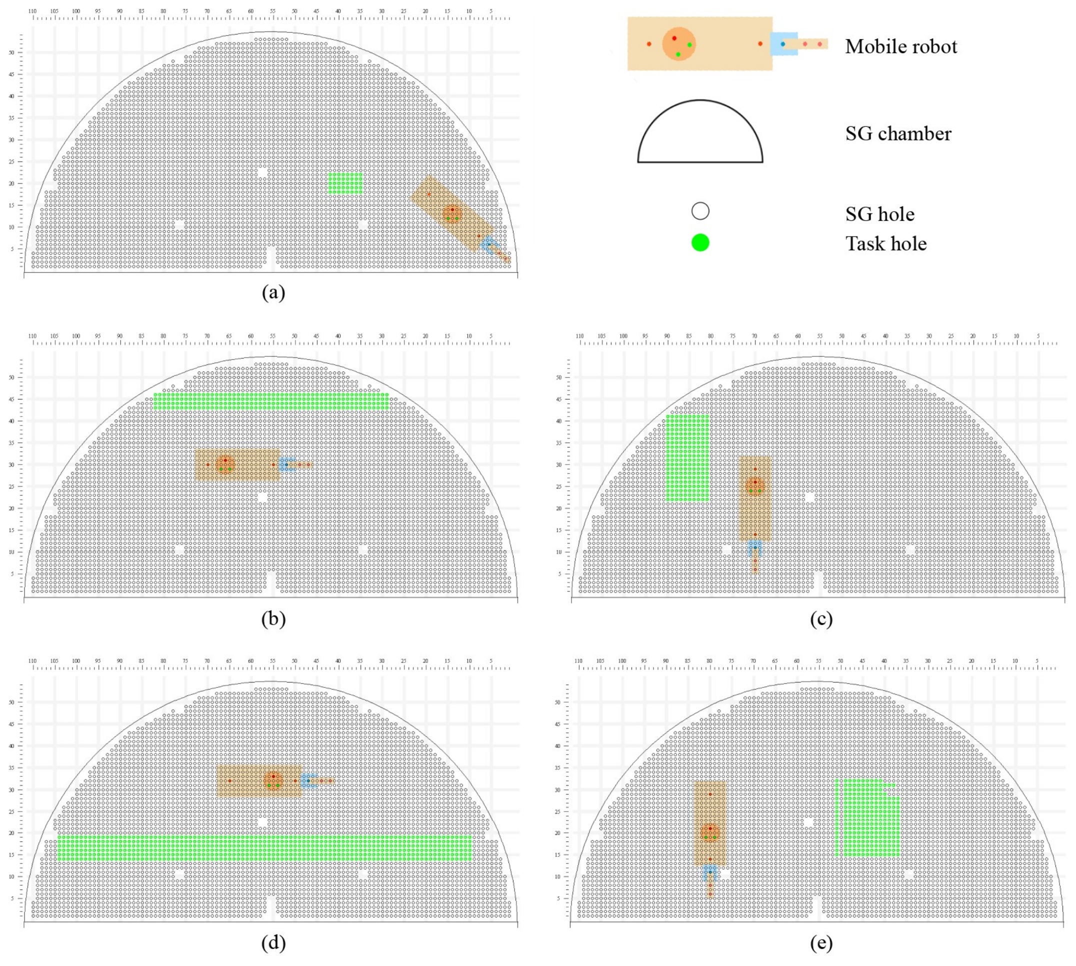


Fig. 15 Five groups of robot current position and task distribution for experiments. (a)–(e) correspond to the setup for experiments 1–5, respectively. SG: steam generator.

Experimental results are shown in Table 3, where the best results from our proposed algorithm M3 are marked with *. Since the performances of M1 and M2 depended on the distribution of the given tasks, we used the better between M1 and M2 results (marked with † in Table 3) to compare with M3. As seen, all three algorithms could cover the given tasks. We demonstrate a part of the robot moving process of Group 1 in Fig. 14, where the real robot pictures and the 2D maps are shown correspondingly. The planned base position number and completed task ratio are shown in Fig. 16. The robot configurations of the start and end tasks of the four sub-regions of Group 5 are demonstrated in Fig. 17.

Compared with the better solution between M1 and M2, the computation time, the number of base positions, and the distance cost of M3 were reduced by 93.37%, 82.92%, and 54.07%, respectively. Unlike M1 and M2, which searched all tasks, M3 used the key tasks to search to achieve the smallest judged task number. Furthermore, M3 had the fewest base positions, shortest path planning time, and lowest distance cost with the overall search based on the three types of inspection. Therefore, M3 had the highest computation efficiency. Comparing the second and fifth experiment results of M3, which had close completed task numbers, we found that the task planning time of the fifth was almost twice that of the second. This was because the fifth group experiment complemented regions and had redundant judgments, which increased the computation time.

Moreover, the number of actually inspected holes of M3 was reduced by 3.77%, compared to the better algorithm between M1 and M2. M3 had the smallest completed task ratio because it prioritized the optimal working region. In contrast, M1 and M2 searched for lines of tasks and could not take advantage of the width direction, resulting in many robot base positions and completed tasks.

In conclusion, in all experiments M3 had the best efficiency. Based on the superiority of M3 in the above indicators, M3 increased the inspection efficiency by 33.31% compared to the better algorithm between M1 and M2. The running time was affected by the base positions and distance cost. When the robot moved along the same distance, the more positions it needed to arrive at, the more times the tightened state changed, and each change took at least 8.5 s. M3 planed the most minor base positions and distance cost to achieve the best motion performance.

5.2 Base sequence planning

We conducted base sequence planning experiments to

Table 3 Experimental results for task planning on a semicircular tube plate

No.	Task number	Planning time/ms						Running time/ms			Base number			Base cost			Completed task number		
		Task planning			Path planning														
		M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3
1	40	25	30	18	236	537	47	349	415	306	5	8	1	8.1	9.0	5.1	40†	48	**40***
2	216	52	59	28	3023	13563	124	1508	1833	1072	17	38	3	41.3	40.0	10.1	224	216†	**216***
3	200	49	55	40	1544	5814	252	1349	1678	1206	14	29	5	19.1	29.0	18.2	240	200†	**200***
4	570	136	150	61	13041	30701	1595	4617	4935	2934	48	75	14	68.3	75.0	30.2	672†	760	**572***
5	240	92	63	53	7491	5261	167	2453	2178	1272	36	28	4	44.1	36.1	15.2	252†	260	**242***

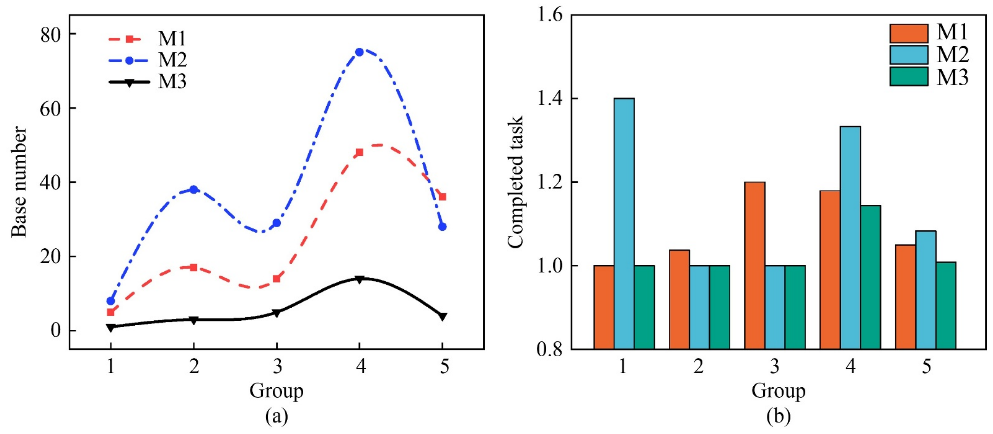


Fig. 16 Planned base positions number and the completed task ratio: (a) comparison of the planned base positions for different algorithms and (b) comparison of the completed task ratio for different algorithms.

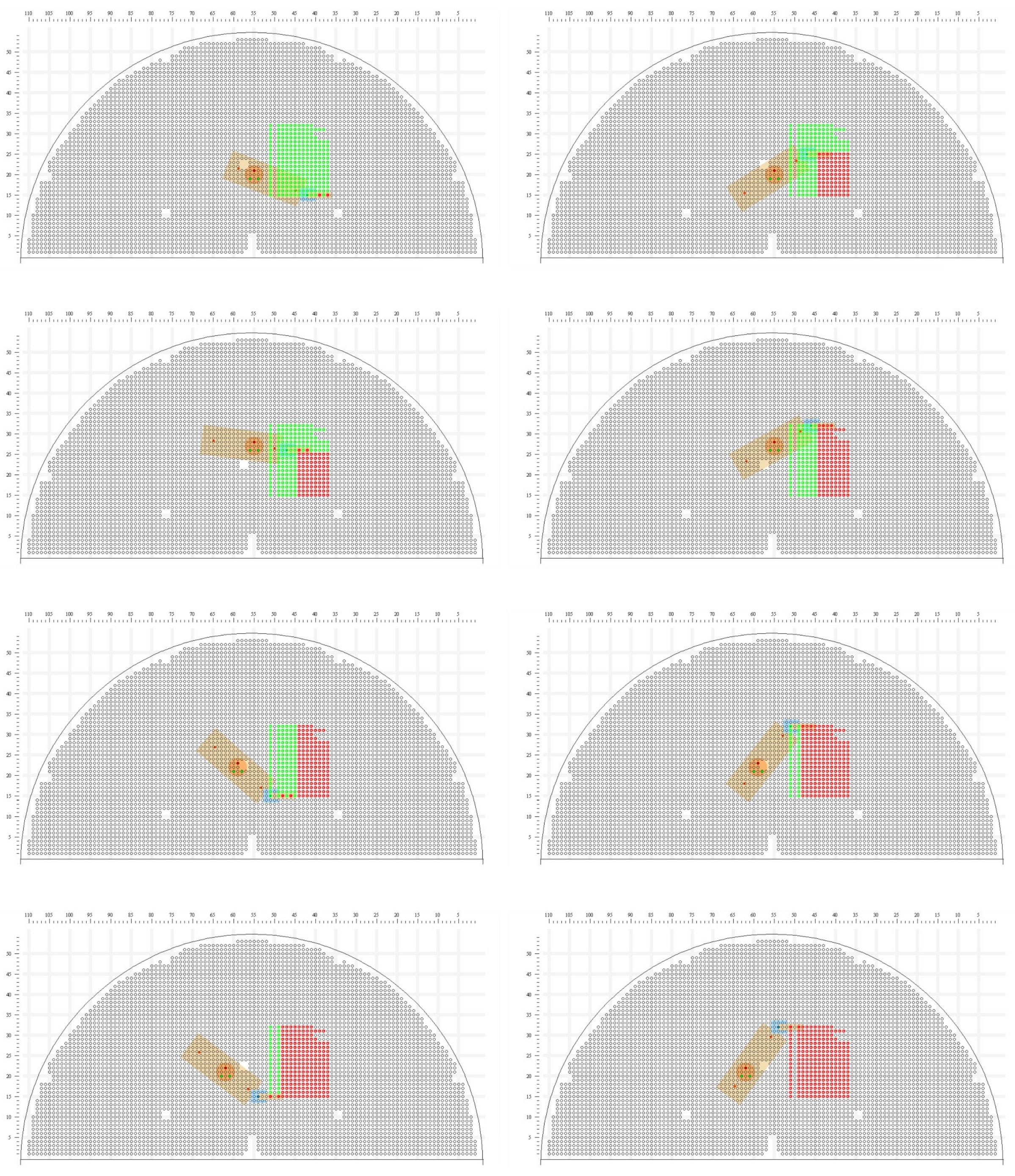


Fig. 17 Demonstrations of the inspection process of four sub-regions of Group 5. Red points are tasks that are completed.

compare the ACO, PSO, and ABC algorithms with 6, 10, and 15 base positions, respectively. The maximum number of generations and populations are 100 and 30.

The results are shown in Table 4. The obtained path costs versus iterations are plotted in Fig. 18. The results showed that all three algorithms could reach optima. Among three algorithms, the ABC cost the longest computation time, while the PSO took the shortest. However, PSO needed the largest iterations to converge, whereas ACO always converged within the least generations. When the maximum number of generations and populations were increased to 1000 and 500, the ACO cost the shortest computation time, followed by the PSO, while the computation time of the ABC was ten times as the ACO or PSO.

For our robots on the SG tube plate, the number of base positions planned from our task planning algorithm is not large. Hence, the ACO is our most profitable choice for base sequence planning considering its computation time and rapid convergence.

Table 4   Experiment results for base sequence planning

No.	Base position	Optimal path cost	Computation time/ms			Iteration		
			ACO	PSO	ABC	ACO	PSO	ABC
1	6	19.2	84	23	609	1	3	5
2	10	16.1	217	60	1390	1	14	8
3	15	21.1	108	30	1814	1	19	13

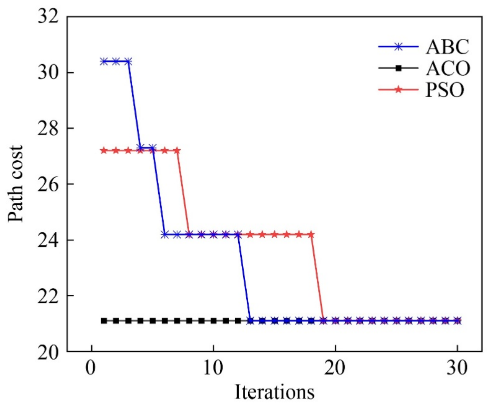


Fig. 18   Path cost versus iterations for artificial bee colony optimization (ABC), ant colony optimization (ACO), and particle swarm optimization (PSO).

6 Conclusions

This paper introduced a novel design of an inspection robot and proposed a high-efficiency inspecting method for HIT-Crawler for heat transfer tubes in an SG. This method takes advantage of the tube tasks distribution and form of tool inspection with its task allocation and overall planning. The planned base positions and configurations ensure that the robot completes all inspections so that the manual operations can be replaced completely. As a result, the error rates are reduced, and robot efficiency increases.

The experiment results show that all of the tasks planned by the proposed method can be accomplished. Moreover, the method can reduce the computation time by 93.37%, the number of actually inspected holes by 3.77%, and the running time by 33.31%. This method has been applied to a mobile robot in an NPP. Practical results show that the method enhances the efficiency and reliability of robot inspection and shortens the shutdown of the NPP.

To better demonstrate the performance of our robot, we compare it with two popular inspection robots in service: ZR-100 and Pegasys as follows.

1) Operation mode

ZR-100 requires all toes to grasp the tube plate while the tool is inspecting so that the inspection time is increased by toe releasing and grasping when it switches to a different tube hole. Whereas, the body of Pegasys can only rotate at a fixed angle limited by its mechanism, so the motion range of its end-effector tools is only 12 holes. As for our proposed robots, the workspace of the fixed tool is a ring area, allowing about 700 holes. Therefore, with the robot base at a fixed position, the number of holes that our robot inspects is far more than the other two robots in service.

2) Operation speed

To evaluate the operation speed, we time the translation of the robots from one hole to its neighbor. Our proposed robot takes 245 ms traveling across holes, which is less than 250 ms by ZR-100 or 300 ms by Pegasys.

3) Planning algorithm

Both ZR-100 and Pegasys are running algorithms using row-by-row or column-by-column planning. As we mentioned above, our proposed planning algorithm outperforms either method with fewer base positions. This benefit is not necessarily coherent with specific robot structure, which means our algorithm can achieve the best performance no matter which type of mobile robot is used.

In conclusion, our robot has the best planning algorithm, the highest inspection efficiency, and the fastest translation speed. Suppose we consider the motion mode, calculate the workspace, and properly implement our planning algorithm on ZR-100 and Pegasys, their performance can be sufficiently improved so that they can inspect the NPPs efficiently.

Nevertheless, the complementing principle can introduce several extra tasks in the task allocation phase, and a sufficient number of extra tasks cause unnecessary computation. In the future, we will research different task-region integration for fewer solving times and more optimal base positions.

Nomenclature

Abbreviations

ABC	Artificial bee colony optimization
ACO	Ant colony optimization
D–H	Denavit–Hertenbeg
DoF	Degree-of-freedom
IK	Inverse kinematics
NPP	Nuclear power plant
PSO	Particle swarm optimization
SG	Steam generator
TSP	Traveling salesman problem

Variables

a	Rectangular region of width/length
B_{Best}	Best base position for the current searching region

B_{Current}	Current base position
$\text{Base}(x, y)$	Distance between the foot toes
C	Robot rotation speed
C_{r}	Turning cost
C_{w}	Translation cost
d	Distance between two tube holes
$d(\text{cur}, \text{next}, t)$	Heuristic cost function
$f(w_j, l_j, k)$	Minimum number of tasks to be completed of $\mathcal{V}_j$
$F_{\text{D}}(\mathcal{B})$	Number of different elements in $\mathcal{B}$
$\text{Hole}(x, y)$	Distribution of the tube holes
k	Distance from the foot toe to the tool
$\bar{l}$	Length of the unassigned work row
l_j	Length of $\mathcal{V}_j$
$l_{\max}$	Maximum length of the region according to the work matrix
l_{s}	Length of the search row
$L_{\text{Ant}}(\mathcal{B})$	Total distance of the points in the set $\mathcal{B}$
L_{Outer}	Distribution of the base toes
$L_{\text{OutertoTool}}$	Distance between the tools
$N(w)$	Number of completed tasks
$\text{Plug}(x, y)$	Distribution of the plugging holes
$[q_{\text{h}}^1 \quad q_{\text{h}}^2 \quad q_{\text{h}}^3]$	Robot joint configuration solution
R_m	Maximum size of the optimal region
R_{p}	Configuration matrix
R'_{p}	Intermediate variable to obtain R_{p}^*
R_{p}^*	All matrices that minimize the number of base positions
$S_{\max}$	Robot maximum translation distance
t	Number of turns
T_{Grasp}	Robot releasing time
T_i	Task tube hole
T_{pb}	Optimal work matrix
T_{pl}	Length work matrix
T_{ps}	Suboptimal work matrix
$\text{Task}(x, y)$	Distribution of the task holes
V_{Rot}	Robot translation speed
V_{Trans}	Robot grasping time
$V(w)$	Evaluation function of the main working direction
w_j	Width of $\mathcal{V}_j$
$(x_{\text{h}}, y_{\text{h}})$	Robot base position solution
$\lfloor x \rfloor$	Downward rounding function
$\bar{\alpha}$	Factor along the length direction
α_{2k}	Factor along the width direction
$\alpha_{2k-1}, \alpha_{2k-2}, ..., \alpha_{k+1}$	Factor of the compound direction
$\mathcal{B}$	Point set containing n base positions
$\mathcal{T}$	Task set
$\mathcal{V}_j$	Sub-region

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. U2013214) and the Self-Planned Task of the State Key Laboratory of Robotics and System (HIT), China (Grant No. SKLRS202001A03).

References

1. Tao M, Ke Z W, Li X L, Qu R T, Zhao Z X, Wu J, Li Y. The research of the model-free adaptive control method of once-through steam generator in nuclear power. In: Proceedings of the 2017 25th International Conference on Nuclear Engineering. Shanghai: ASME, 2017, V001T04A014
2. Jeon S H, Hong S, Kwon H C, Hur D H. Characteristics of steam generator tube deposits in an operating pressurized water reactor. Journal of Nuclear Materials, 2018, 507: 371–380
3. Huang Z J, Zou Y K, Ding J, Lu J F. Experimental investigation of heat transfer in coiled tube type molten salt steam generator. Applied Thermal Engineering, 2019, 148: 1131–1138
4. Drexler A. Steam-water cycle chemistry relevant to nuclear steam generators. In: Riznic J, ed. Steam Generators for Nuclear Power Plants. Elsevier, 2017, 127–153
5. Anoda Y, Tasaka K, Kumamaru H, Shiba M. Rosa-III System Description for Fuel Assembly, 4. Technical Report, JAERI-M 9363, 1981
6. Xu B Y, Li G, Xing Z M, Cai H G, Zhao J, Fan J Z. Design and motion performance of new inspection robot for steam generator heat transfer tubes. In: Proceedings of 2021 IEEE International Conference on Real-time Computing and Robotics (RCAR). Xining: IEEE, 2021, 662–667
7. Wu J R. Research on key technology and prototype development of steam generator overhaul robot. Dissertation for the Doctoral Degree. Harbin: Harbin Engineering University, 2009 (in Chinese)
8. Green S J, Hetsroni G. PWR steam generators. International Journal of Multiphase Flow, 1995, 21: 1–97
9. Obrutsky L. Nuclear steam generator tube inspection tools. In: Riznic J, ed. Steam Generators for Nuclear Power Plants, Elsevier, 2017, 495–510
10. Li J K, Wu X Y, Xu T T, Guo H W, Sun J Q, Gao Q S. A novel inspection robot for nuclear station steam generator secondary side with self-localization. Robotics and Biomimetics, 2017, 4(1): 26
11. Sun B Z, Yang Y L. Numerically investigating the influence of

tube support plates on thermal-hydraulic characteristics in a steam generator. Applied Thermal Engineering, 2013, 51(1–2): 611–622

12. Ye C F, Huang Y, Udpa L, Udpa S S. Novel rotating current probe with GMR array sensors for steam generate tube inspection. IEEE Sensors Journal, 2016, 16(12): 4995–5002
13. Gerkey B P, Matarić M J. A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research, 2004, 23(9): 939–954
14. Elloumi W, El Abed H, Abraham A, Alimi A M. A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP. Applied Soft Computing, 2014, 25: 234–241
15. Le Ny J, Dahleh M, Feron E. Multi-agent task assignment in the bandit framework. In: Proceedings of the 45th IEEE Conference on Decision and Control. San Diego: IEEE, 2006, 5281–5286
16. Gerkey B P, Mataric M J. A framework for studying multi-robot task allocation. In: Proceedings of Multi-Robot Systems: From Swarms to Intelligent Automata. Washington DC, 2003, 15–26
17. Khaluf Y, Vanhee S, Simoens P. Local ant system for allocating robot swarms to time-constrained tasks. Journal of Computational Science, 2019, 31: 33–44
18. Thakar S, Fang L W, Shah B, Gupta S. Towards time-optimal trajectory planning for pick-and-transport operation with a mobile manipulator. In: Proceedings of 2018 IEEE the 14th International Conference on Automation Science and Engineering (CASE). Munich: IEEE, 2018, 981–987
19. Xu J R, Harada K, Wan W W, Ueshiba T, Domae Y. Planning an efficient and robust base sequence for a mobile manipulator performing multiple pick-and-place tasks. In: Proceedings of 2020 IEEE International Conference on Robotics and Automation (ICRA). Paris: IEEE, 2020, 11018–11024
20. Vafadar S, Olabi A, Panahi M S. Optimal motion planning of mobile manipulators with minimum number of platform movements. In: Proceedings of 2018 IEEE International Conference on Industrial Technology (ICIT). Lyon: IEEE, 2018, 262–267
21. Stern R, Sturtevant N, Felner A, Koenig S, Ma H, Walker T, Li J Y, Atzmon D, Cohen L, Kumar T K, Boyarski E, Bartak R. Multi-agent pathfinding: definitions, variants, and benchmarks. In: Proceedings of the 12th Annual Symposium on Combinatorial Search. 2019
22. Yu J J, LaValle S M. Optimal multirobot path planning on graphs: complete algorithms and effective heuristics. IEEE Transactions on Robotics, 2016, 32(5): 1163–1177
23. Xu B Y, Li G, Zhang K, Cai H G, Zhao J, Fan J Z. Motion planning of a steam generator mobile tube-inspection robot. Nuclear Engineering and Technology, 2022, 54(4): 1374–1381
24. Cheikhrouhou O, Khoufi I. A comprehensive survey on the multiple traveling salesman problem: applications, approaches and taxonomy. Computer Science Review, 2021, 40: 100369
25. Rokbani N, Kumar R, Abraham A, Alimi A M, Long H V, Priyadarshini I, Son L H. Bi-heuristic ant colony optimization-based approaches for traveling salesman problem. Soft Computing, 2021, 25(5): 3775–3794
26. Cui Y, Zhong J B, Yang F R, Li S X, Li P H. Multi-subdomain grouping-based particle swarm optimization for the traveling salesman problem. IEEE Access, 2020, 8: 227497–227510
27. Khan I, Maiti M K. A swap sequence based artificial bee colony algorithm for traveling salesman problem. Swarm and Evolutionary Computation, 2019, 44: 428–438
28. Ebadinezhad S. DEACO: adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem. Engineering Applications of Artificial Intelligence, 2020, 92: 103649
29. Yang K, You X M, Liu S, Pan H. A novel ant colony optimization based on game for traveling salesman problem. Applied Intelligence, 2020, 50(12): 4529–4542
30. Kapanoglu M, Alikalfa M, Ozkan M, Yazıcı A, Parlaktuna O. A pattern-based genetic algorithm for multi-robot coverage path planning minimizing completion time. Journal of Intelligent Manufacturing, 2012, 23(4): 1035–1045
31. Gabriely Y, Rimon E. Spanning-tree based coverage of continuous areas by a mobile robot. Annals of Mathematics and Artificial Intelligence, 2001, 31(1–4): 77–98
32. Acar E U, Choset H, Rizzi A A, Atkar P N, Hull D. Morse decompositions for coverage tasks. The International Journal of Robotics Research, 2002, 21(4): 331–344